# Evaluating Predictions of Transfer and Analyzing Student Motivation

by

Ethan Croteau

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

By

_____

April 2004

APPROVED:

_____
Professor Neil Heffernan, Thesis Advisor

_____
Professor David Brown, Thesis Reader

_____
Professor Michael Gennert, Head of Department

## Abstract

Cognitive Science is interested in being able to develop methodologies for analyzing human learning and performance data. Intelligent tutoring systems need good cognitive models that can predict student performance. Cognitive models of human processing are also useful in tutoring because well-designed curriculums need to understand the common components of knowledge that students need to be able to employ. A common concern is being able to predict when transfer should happen. We describe a methodology first used by Koedinger that uses empirical data and cognitively principled task analysis to evaluate the fit of cognitive models. This methodology seems particularly useful when you are trying to find evidence for "hidden" knowledge components, which are hard to assess because they are confounded with accessing other knowledge components. We present this methodology as well as an illustration showing how we are trying to use this method to answer an important cognitive science issue.

## Acknowledgements

I would like to thank my extraordinary thesis advisor, Professor Neil Heffernan. He gave me constant advice and motivation to progress forward. My thesis reader, Professor David Brown, who I have known as a fencing coach, MQP advisor and incredible teacher, has been wonderful inspiration in the area of artificial intelligence and expert systems. I would like to mention my friends Larissa and Jon who shared the experience of researching a similar and relevant topic. I benefited from the Tutor Research Group (TRG), which provided much needed snacks as well as wisdom. From the math department, thanks goes to Jason Wilbur and Carlos Morales who provided insight as well as a theoretical perspective. It would have been difficult to finish without the motivation from my friends. Without the support of my loving family and girlfriend, my thesis would have been impossible.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

Intelligent Tutoring Systems are becoming more common due to the increased presence of personal computing. They are a relevant topic of research as their study involves a foundation of both computer and cognitive science. An Intelligent Tutoring System (ITS) is a form of an expert system, capable of presenting course material and intelligently guiding a student to the solution through the use of domain specific strategies. These strategies are often referred to as "Tutoring Strategies" and their study is of great importance as seen by the large investment of time and resources to enhance the educational curriculums and materials used by educators and those who are self-taught. It is obvious that some strategies are better than others, but as to what extent one strategy is better than another is of great interest. Key factors such as time-on-task when using a tutorial strategy must be considered and also motivation (a student's desire to continue learning).

This thesis presents two separate but related themes in the form of papers that were submitted to ITS 2004 (http://www.itsconference.org/2004/). They are similar in that they both discuss the tutoring strategies employed by the Ms. Lindquist ITS developed by Neil Heffernan (see http://www.algebratutor.org). The first paper presented is an approach to constructing and evaluating models of student learning. This approach is applied to the data collected by the Ms. Lindquist tutor for one of her intelligent tutoring strategies. The approach is unique in that it attempts to go from a conventional linear mapping of skill usage to a non-linear mapping of skills based on the concept of a "Transfer Model". We suggest that transfer models can be useful for proposing and evaluating domain theories. A particularly interesting domain theory is investigated, which relates to an ongoing debate between several theorists, which is, "Why are algebra word problems difficult?" The second paper presents a web-based evaluation for the Ms. Lindquist tutor, which serves as a continuation of a previous web-based evaluation conducted by Heffernan (2003). In this evaluation, student learning and motivation are compared between those students receiving different advanced tutoring strategies. From the web-based evaluation, it can be seen that not only are students able to benefit from using the tutoring system, but also the researchers by learning more about the students being tutored.

Koedinger and Junker's[1] insight invented the basic methodology that is described, extended and applied in this thesis. Suppose you were a tutor that tried to get students ready to take the SAT (or some similar mathematics test.) Suppose your normal method was to present students a somewhat random SAT problem to see if they got it right. If they failed, you would provide some tutoring to make sure they eventually got the right answer. Let us suppose you wanted to know what other problems a student would do well on if they got practice on problems of a given type. Presumably, giving students

[1] The idea of evaluating a transfer model by looking for a parsimonious fit (using the Bayesian Information Criterion) to the data is due to Koedinger & Junker who shared this idea with my advisor during his postdoc. They conceived the idea of using two parameters in the logistic regression for each knowledge component. One of the parameters was used to indicate if the knowledge components were required. (We generalized the idea to not just the Boolean- present or absent- but to the number of times that knowledge component was used- zero, one, two, etc. times.) I am not familiar enough with the statistics literature to say these ideas are totally novel.

practice on algebra problems will tend to transfer to other algebra problems (meaning that practice on algebra problems will make it more likely that that student will get other algebra problems correct), and might transfer, by a smaller amount, to geometry problems, but is unlikely to transfer to Verbal SAT Vocabulary problems due to the non-existent (presumably) overlap in the bits of knowledge between algebra problems and vocabulary problems. We desire a method that will allow us to build a model that predicts when transfer will happen between problems of a given type. We call this model a *Transfer model*. The better your transfer model, the more accurately you will be able to predict students' performance on different types of problems due to practice on certain other types of problems. If two problem types share no underlying knowledge then in theory, practice on either will make no difference in the average performance of the other problem type.

## 1.1 Motivation

There are several reasons for improving a tutor's predictive accuracy of student's performance. The most notable benefit is allowing the ITS to keep better track of students' performance, which would in turn be useful for determining when a student had mastered a particular cognitive skill. Tracking students' performance is useful for visually displaying a student's progress by means of a *skillometer* (a gauge indicating the level of proficiency for a set of cognitive skills). Improved tracking of students' performance would also lend to allowing students to resume at approximately the same level of skill mastery as they had left during a previous tutoring session.

From a cognitive science standpoint, the methodology used to improve a tutor's predictive accuracy would also be useful for determining the difficulty of the individual cognitive skills and the rate that those skills are acquired. Also of great scientific interest would be the potential discovery of *hidden skills* (cognitive skills that only exist in the presence of another skill). Awareness of hidden skills increases the predictive accuracy of a transfer model and provides researchers with a greater understanding of how cognitive skills are learned.

Motivation for this work is also driven by the demand to have a convenient set of tools to allow domain experts to develop their own ITS. Although quality tools are currently being developed for ITS construction, the methodology developed here could supplement tutor development by providing the capability of analyzing students' usage of an ITS and refining the cognitive model in use. In this case, not only would students be learning from the tutor, but also the tutor would be learning from its students. Every so often the tutor may discover a new cognitive skill that would better its predictive accuracy and such a discovery would be praiseworthy. The ability to produce tutors with a self-improving cognitive model is fascinating and would result in tutors having a greater ability to predict student performance.

# 2 Why Are Algebra Word Problems Difficult? Using Tutorial Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model

Some researchers have argued that algebra word problems are difficult for students because they have difficulty in comprehending English. Others have argued that because algebra is a generalization of arithmetic, and generalization is hard, it's the use of variables, per se, that cause difficulty for students. Heffernan and Koedinger (1997, 1998) presented evidence against both of these hypotheses. In this paper we present how to use tutorial log files from an intelligent tutoring system to try to contribute to answering such questions. We take advantage of the Power Law of Learning, which predicts that error rates should fit an inverse power function, to try to find the best fitting mathematical model that predicts whether a student will get a question correct. We decompose the question of "Why are Algebra Word Problems Difficult?" into two pieces. First, is there evidence for the existence of this articulation skill that Heffernan and Koedinger argued for? Secondly, is there evidence for the existence of the skill of "composed articulation" as the best way to model the "composition effect" that Heffernan and Koedinger discovered?

## 2.1 Introduction

Many researchers had argued that students have difficulty with algebra word-problem *symbolization* (writing algebra expressions) because they have trouble comprehending the words in an algebra word problem. For instance, Nathan, Kintsch, & Young (1992) "claim that [the] symbolization [process] is a highly *reading-oriented* one in which poor *comprehension* and an inability to access relevant long term knowledge leads to serious errors." [emphasis added]. However, Heffernan & Koedinger (1997, 1998) showed that many students can do *computation* tasks well, whereas they have great difficulty with the *symbolization* tasks [See Table 1 for examples of *computation* and *symbolization* types of questions]. They showed that many students could comprehend the word problems, yet still could not do the symbolization. An alternative explanation for "Why Are Algebra Word Problems Difficult?" is that the key is the use of variables. Because algebra is a generalization of arithmetic, and it's the variables that allow for this generalization, it seems to make sense that it's the variables that make algebra symbolization hard.

However, Heffernan & Koedinger presented evidence that cast doubt on this as an important explanation. They showed there is hardly any difference between students' performance on *articulation* (see Table 1 for an example) versus *symbolization* tasks, arguing against the idea that the hard part is the presence of the variable per se.

Instead, Heffernan & Koedinger hypothesized that a key difficulty for students was in articulating arithmetic in the "foreign" language of algebra. They hypothesized the existence of a skill for articulating one step in an algebra word problem. This articulation step requires that a student be able to say (or "articulate") how it is they would do a computation, without having to actually do the arithmetic. Surprising, the found that is was easier for a student to actually do the arithmetic then to articulate what they did in an expression. To successfully articulate a student has to be able to write in

the language of algebra. Question 1 for this paper is "Is there evidence that supports the conjecture that the *articulate* skill really exists?"

In addition to conjecturing the existence of the skill for articulating a single step, Heffernan and Koedinger also reported what they called the "composition effect" which we will also try to model. Heffernan and Koedinger took problems requiring two mathematical steps and made two new questions, where each question assessed each of the steps independently. Their results (1997) showed that "a two operator problem is harder than both of the parts that make it up put together"[2]. They termed this the composition effect. This led them to speculate as to what the "hidden" difficulty was for students that explained this difference in performance. They argued that the hidden difficulty included knowledge of *composition of articulation*. Heffernan and Koedinger attempted to argue that the composition effect was due to difficulties in articulating rather than on the task of comprehending, or at the symbolization step when a variable is called for. In this paper we will compare these hypotheses to try to determine the source of the composition effect. We refer to this as Question 2.

Heffernan and Koedinger's arguments were based upon two different samplings of about 70 students. Students' performances on different types of items were analyzed. Since students were not provided feedback during this twenty-minute assessment, it is reasonable to assume that no learning was taking place (Laurillard, 1993) and likewise no need to model learning. Heffernan and Koedinger went on to create an intelligent tutoring system, "Ms Lindquist", to teach student how to do similar problems. In this paper we attempt to use tutorial log file data collected from this tutor to shed light on this controversy. The technique we present is useful for intelligent tutoring system designers as it shows a way to use log file data to refine the mathematical models we use in predicting whether a student will get an item correct. For instance, Corbett and Anderson (1992) describe how to use "knowledge tracing" to track students performance on items related to a particular skill, but all such work is based upon the idea that you know what skills are involved already. But in this case there is controversy (see also Nathan and Koedinger, 2000) over what are the important skills (or more generally, *knowledge components*[3]). Because Ms Lindquist selects problems in a curriculum section randomly, we can learn what the knowledge components are that are being learned. With out problem randomization we would have no hope of separating out the effect of problem ordering with the difficulty of individual questions.

In the following sections of this paper we present the investigations we did to look into the existence of both the skills of *articulation* as well as *composition of articulation*. In particular, we present mathematically predictive models of a student's chance of getting a question correct. It should be noted, such predicative models have many other uses for intelligent tutoring systems, so this methodology has many uses.

### 2.1.1 Knowledge Components and Transfer Models

We use the concept of a *Transfer Model* (Heffernan & Croteau, 2003) to build a model that predicts when transfer will happen between problems of a given type. If two

[2] The experiment design consisted of a difficulties factors assessment whereby Heffernan and Koedinger sampled student performance on a set of 128 problems created by systematically modifying 8 core problem situations along 4 binary factor dimensions.

[3] Rather than referring to these students' abilities as skills, which makes the assumption of being procedural, these abilities are referred to as knowledge components, which also includes declarative knowledge.

question types share no underlying knowledge then in theory, practice on either question type will make no difference in average performance on either of the two. To summarize the concept of a transfer model, it can be seen as a two dimensional array, in which question types are represented by the rows and knowledge components are represented by the columns. Looking at a single question type (horizontal row), each cell contains a number indicating the number of times that column's knowledge component is found in that row's question type (the number that KC will be applied for a correct answer).

As we said in the introduction, some (i.e., Nathan, Kintsch and Young) believed that comprehension was the main difficulty in solving algebra word problems. We summarize this viewpoint in Table 1 which has three skills labeled the "Base" model.

| | | Knowledge Components Demonstrated | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *Base + Model* | | | | Composition KCs | | | |
| | | *Base Model* | | | | | | | |
| | **Example dialog – correct response within brackets** | Arithmetic | Comprehending One-Step | Using a Variable | Articulating One-Step | Composition of Articulation | Composition of Comprehension | Composition of Arithmetic | Composition of Symbolization |
| 1 Step Computation | Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for *3 minutes* and stops to rest. How far **did she row**? = [120] | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 Step Computation | Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for *3 minutes* and stops to rest. How far **is she from the dock now**? = [680] | 2 | 2 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 Step Articulation | Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for *3 minutes* and stops to rest. Write an expression for how far **did she row**? = [40*3] | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 Step Articulation | Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for *3 minutes* and stops to rest. Write an expression for how far **is she from the dock now**? = [800-40*3] | 0 | 2 | 0 | 2 | 1 | 1 | 0 | 0 |
| 1 Step Symbolization | Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for *"m" minutes* and stops to rest. How far **did she row**? = [40m] | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 Step Symbolization | Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for *"m" minutes* and stops to rest. How far **is she from the dock now**? = [800-40m] | 0 | 2 | 1 | 2 | 1 | 1 | 0 | 1 |

**Table 1 A Transfer Model Showing the mapping between questions and knowledge components.**

The *Base Model* consists of arithmetic knowledge components (KC), comprehension KCs, and KCs related to using a variable. Each number in Table 1 indicates the number of times a particular KC has been applied. For illustration purposes, we see that the second row shows that for a two-step "compute" problem the student will have to

*comprehend* two different parts of the word problem (including but not limited to, figuring out what operators to use with which literals mentioned in the problem) as well as using the *arithmetic* KC twice. Each knowledge component has an associated difficulty, possibly different for each knowledge component, which is imposed by its existence. When comparing two question types having all but one of the knowledge components with the same value, then the question type with the higher knowledge component value is considered as being more difficult. We see that this model proposes symbolization problems as being harder than articulation problems due the presence of a variable in the symbolization problems, which is indicated by the "Using a Variable" knowledge component. If students are proficient at doing arithmetic (the arithmetic knowledge component can be ignored due to this knowledge component already being learned), then the *Base Model* suggests that computation problems should be easier than articulation problems.

The 4th column of Table 1, "*articulating one-step*" is the KC that Heffernan and Koedinger (1997, 1998) conjectured was important to understanding what make algebra problems so difficult for students. We want to build a mathematical model with the *Base Model* KCs and compare it what we call the "*Base+Model"*, that also includes the *articulating one-step* KC.

So Question 1 in this paper compares the *Base Model* with a model that adds in the *articulating one-step* KC. Question 2 goes on to find the best way of adding knowledge components that allow the model to predict the compensation effect. The four knowledge components in the last four columns represent different hypotheses about what explains the composition effect. You will notice that all four columns have KCs that apply to only a two-step problem, thus representing competing hypotheses for where the difficulty in composition comes into play. Is the composition during the articulation, comprehension, articulation, or the symbolization? Heffernan and Koedinger speculated that there was a composition effect during articulation, suggesting that knowing how to treat an expression the same way you treat a number would be a skills that students would have to learn if they were to be good at problems that involved two-step articulation problems. If Heffernan and Koedinger's conjecture was correct we would expect to find that the *composition of articulation* KC is better (in combination with one of the two Base Models variants) at predicting students' difficulties than any of the other composition KCs.

### 2.1.2 Understanding how we use this Model to Predict Transfer

Qualitatively, we can see that the transfer model in Table 1 predicts that practice on one-step computation questions should transfer to one-step articulation problems only to the degree that a student learns (i.e., receives practice at employing) the *comprehending one-step* KC. We can turn this qualitative observation into a quantified prediction method by treating each knowledge component as having a *difficulty* parameter and a *learning* parameter. This is where we take advantage of the Power Law of Learning, which is one of the most robust findings in cognitive psychology. The power law says that the performance of cognitive skills improve approximately as a power function of practice (Newell & Rosenbloom, 1981, Anderson & Lebiere, 1998). This has been applied to both error rates as well as time to complete a task, but our use here will be with error rates. This can be stated mathematical as follows:

$$\text{Error Rate}(x) = b * x^{-d}$$

Where *x* represents the number of times the student has received feedback on the task, *b* represents a *difficulty* parameter related to the error rate on the first trail of the task, and *d* represents a *learning* parameter related to the learning rate for the task. Tasks that have large *b* values represent tasks that are difficult for students the first time they try it (could be due to the newness of the task, or the inherit complexity of the task). Tasks that have a large *d* coefficient represent tasks where student learning is fast. Conversely, small values of *d* are related to tasks that students are slow to improve[4]. In this work, we follow Junker, Koedinger, & Trottini (2000) in using logistic regression to try to predict whether a student will get a question correct, based upon both item factors (like what knowledge components are used for a given question, which is what we are calling *difficulty* parameters), student factors (like a students pretest score) and factors that depend on both students and items (like how many times this particular students has practiced their particular knowledge component, which is what we are calling *learning* parameters.) Corbett & Anderson (1992), Corbett, Anderson & O'Brien (1995) and Draney, Pirolli, & Wilson (1995) report results using the same and/or similar methods as described above. There is also a great deal of related work in the psychometric literature related to item response theory (see Embretson & Reise, 2000 for an introduction) but most of it is focused on analyzing test (e.g., SAT or GRE) rather than student learning.

### 2.1.3 Using the Transfer Model to Predict Transfer

Heffernan (2001) created Ms. Lindquist, an intelligent tutoring system, and put it online (***www.algebratutor.org***) and collected tutorial log files for all the students learning to symbolize. For this research we selected a data set for which Heffernan (see Heffernan's (2003) reporting of "Experiment 3") had previously reported evidence that students were learning during the tutoring sessions. Some 73 students were brought to a computer lab to work with Ms. Lindquist for two class periods totaling an average of about 1 hour of time for each student. We present data from students working only on the second curriculum section, since the first curriculum was too easy for students and showed no learning. We chose to analyze the half of the students that were randomly selected to get the experimental dialog strategy, which involved asking computation and articulation questions when students were having trouble symbolizing. (An example of this dialog is shown in Table 2 and will be discussed shortly). This resulted in a set of log files from 43 students, comprising 777 rows where each row represents a student's first attempt to answer a given question.

[4] All *learning* parameters are restricted to be positive otherwise the parameters would be modeling some sort of forgetting effect.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 8 | | | | 9 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Question Type | | | | | Outcomes | | Knowledge Components: *Difficulty* Parameters | | | | Knowledge Components: *Learning* Parameters | | | |
| | Scenario Identifier | Factor: *Task directions* | Factor: *Steps* | Attempt at question type | First Attempt in sequence | | Response Time (Seconds) | Response (Correct or not) | Arithmetic | Comprehending One-Step | Using a Variable | Articulating One-Step | Arithmetic | Comprehending One-Step | Using a Variable | Articulating One-Step |
| 1 | 1 | S | 2 | 1 | T | **Scenario 1:** Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now? **Tutor**: Please write an expression for the distance Anne has left to row. | | | **0** | **2** | **1** | **2** | **0** | **0** | **0** | **0** |
| 2 | | | | | | **Student**:8-40/m | **130** | **N** | | | | | | | | |
| 3 | 1 | C | 2 | 1 | T | **Tutor**: Hmm. No. We will come back to the distance Anne has left to row. Let me try to break this down for you. Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes? | | | **2** | **2** | **0** | **0** | **0** | **0** | **0** | **0** |
| 4 | | | | | | **Student**:580 | **38** | **N** | | | | | | | | |
| 5 | 1 | C | 1 | 1 | T | **Tutor**: Hmm. No. Let me ask you a different question. Please compute the distance rowed so far if the number of minutes rowing was 3 minutes? | | | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| 6 | | | | | | **Student**:12 | **19** | **N** | | | | | | | | |
| 7 | 1 | C | 1 | 2 | F | **Tutor**: Hmm. No. Try again please. Please compute the distance rowed so far if the number of minutes rowing was 3 minutes? Hint: Use the fact that her speed rowing is 40 yards per minute and the number of minutes rowing is 3 minutes. | | | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| 8 | | | | | | **Student**: 120 | **9** | **Y** | | | | | | | | |
| 9 | 1 | C | 2 | 2 | T | **Tutor**: Correct. Now, lets try this question again: Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes? | | | **~~2~~ 1** | **~~2~~ 1** | **0** | **0** | **1** | **1** | **0** | **0** |
| 10 | | | | | | **Student**: 680 | **24** | **Y** | | | | | | | | |
| 11 | 1 | A | 2 | 1 | T | **Tutor**: Correct. Using only 800, 40, and 3 write an expression for the distance Anne has left to row | | | **0** | **~~2~~ 0** | **0** | **2** | **~~3~~ 2** | **~~3~~ 2** | **0** | **0** |
| 12 | | | | | | **Student:** 120-800 | **18** | **N** | | | | | | | | |
| 13 | 1 | A | 1 | 1 | T | **Tutor:** No. Your answer of "120-800" equals "-680", but you wanted to write an expression that equaled "680". But that is only because you didn't say where the 120 came from. What math did you do to get the 120? | | | **0** | **~~1~~ 0** | **0** | **1** | **~~3~~ 2** | **~~3~~ 2** | **0** | **0** |
| 14 | | | | | | **Student:** 3*40 | **22** | **Y** | | | | | | | | |
| 15 | 1 | A | 2 | 2 | T | **Tutor:** Correct. Now, lets try this question again: Using only 800, 40, and 3 write an expression for the distance Anne has left to row. | | | **0** | **~~2~~ 0** | **0** | **~~2~~ 1** | **~~3~~ 2** | **~~4~~ 2** | **0** | **1** |
| 16 | | | | | | **Student**: 800-40*3 | **18** | **Y** | | | | | | | | |
| 17 | 1 | S | 2 | 2 | T | **Tutor**: Correct. Now use "m" instead of 3 to again express the distance Anne has left to row. | | | **0** | **~~2~~ 0** | **1** | **~~2~~ 0** | **~~3~~ 2** | **~~6~~ 2** | **0** | **~~3~~ 2** |
| 18 | | | | | | **Student**:800-40*m | **9** | **Y** | | | | | | | | |
| 19 | | | | | | **Tutor**: Correct. Lets do a new problem. | | | | | | | | | | |
| 20 | 2 | S | 2 | 1 | T | **Scenario 2:** Michael works as a waiter. Saturday he worked "h" hours. The restaurant pays him an hourly wage of $3 an hour. He also made $40 dollars in tips. Write an expression for the total amount he earned Saturday Night. **Tutor:** Write an expression for the total amount he earned Saturday Night. | | | **0** | **2** | **1** | **2** | **~~3~~ 2** | **~~8~~ 2** | **1** | **~~5~~ 2** |

**Table 2 Showing a made-up tutor log file and how it uses the Base+Model Transfer Model**

Table 2 shows an example of the sort of dialog Ms. Lindquist carries on with students (this is with "made-up" student responses). Table 2 starts by showing a student working on scenario identifier #1 and only in the last row (Row 20) does the scenario identifier switch. Each word-problem has a single *top-level* question which is always a *symbolize* question. If the student fails to get the top level question correct, Ms. Lindquist steps in to have a dialog (as shown in the 6th column) with the student, asking questions to help break the problem down into simpler questions. The second column indicates the task provided to the student as part of the tutor's dialog strategy, which we refer to as the *Task Direction* where S=Symbolize, C=Compute and A=Articulate. The third column indicates the number of mathematical operations required to solve the problem, which is labeled as *Steps*. By crossing *task direction* and *steps,* there are six different question types, which were previously listed in Table 1. The 4th column defines what we call the *attempt* at a question type. The number appearing in the attempt column is the number of times the problem type has been presented during the scenario. For example, the first time one of the six question types is asked, the attempt for that question will be "1". Notice how on row 7, the attempt is "2" because it's the second time a *one-step computation* question has been asked for that scenario identifier. For another example see rows 3 and 7. Also notice that on line 20 the attempt column indicates a first attempt at a two-step symbolize problem for the new scenario identifier.

Notice that on row 5 and 7, the same question is asked twice. If the student did not get the problem correct at line 7, Ms Lindquist would have given a further hint of presenting six possible choices for the answer. For our modeling purposes, we will ignore the exact number of attempts the student had to make at any given question. Only the first attempt *in a sequence* will be included in the data set. For example, this is indicated in Table 2, in the 7th row of the 5th column, where the "F" for false indicates that row will be excluded from the data set.

The 6th column has the exact dialog that the student and tutor had. The 7th and 8th columns are grouped together because they are both outcomes that we will try to predict.[5] Columns 9-16 show what statisticians call the *design matrix*, which maps the possible observations onto the fixed effect (independent) coefficients. Each of these columns will get a coefficient in the logistic regression. Columns 9-12 show the *difficulty* parameters, while columns 13-16 show the *learning* parameters. We only list the four knowledge components of the *Base+ Model*, and leave out the four different ways to deal with composition. The *difficulty* parameters are simply the knowledge components identified in the transfer model; this means they are the row from Table 1 that corresponds the question type as indicated in the 2nd and 3rd column of Table 2. The *learning* parameter is calculated by counting the number of previous attempts a particular knowledge component has been learned (we assume learning occurs each time the system gives feedback on a correct answer). Notice that these learning parameters are strictly increasing as we move down the table, indicating that students' performance should be monotonically increasing.

Notice that the question asked of the student on row 3 is the same as the one on row 9, yet the problem is easier to answer after the system has given feedback on "the distance rowed is 120". Therefore the difficulty parameters are reduced to reflect the knowledge components yet to be demonstrated correct by subtracting any previous

[5] Currently, we are only predicting whether the response was correct or not, but later we will do a Multivariate logistic regression to take into account the time required for the student to respond.

knowledge components correctly demonstrated for the current scenario. For this example, the KCs on row 7 were correctly demonstrated, indicated by the student's correct response. This explains the reduction in knowledge components appearing on row 9, columns 9 and 10, to reflect that the student had received positive feedback for those knowledge components. By using this technique we make the credit-blame assignment problem easier for the logistic regression because the number of knowledge components that could be blamed for a wrong answer had been reduced. Notice that because of this method with the difficulty parameters, we also had to adjust the learning parameters, as shown by the crossed out learning parameters. Notice that the learning parameters are **not** reset on line 20 when a new scenario was started because the learning parameters extend across all the problems a student does.

### 2.1.4 How the Logistic Regression was applied

With some minor changes, Table 2 shows a snippet of what the data set looked like that we sent to the statistical package to perform the logistic regression. We performed a logistic regressions predicting the dependent variable *response* (column 8) based on the independent variables on the knowledge components (i.e., columns 9-16). For some of the results we present, we also add a student specific column (we used a student's pretest score) to help control for the variability due to students differing incoming knowledge.

## 2.2 Stepwise Removal of Model Parameters

This section discusses how a fit model is made parsimonious by a stepwise elimination of extraneous coefficients. We only wanted to include in our models those variables that were reasonable and statistically significant. The first criterion or reasonableness was used to exclude a model that had "negative" learning curves that predict students would do worse over time. The second criterion of being statistically significant was used to remove, in a stepwise manner, coefficients that were not statistically significant (those coefficients with t-values between 2 and –2 is a rule of thumb used for this). We choose, somewhat arbitrarily, to first remove the learning parameters before looking at the difficulty parameters. We made this choice because the learning parameters seemed to be, possibly, more contentious. At each step, we chose to remove the parameter that had the least significance (i.e., the smallest absolute t-value)[6].

### 2.2.1 Model Evaluation

A systematic approach to evaluating a model's performance (in terms of error rate) is essential to comparing how well several models built from a training set would perform on an independent test set.

We used two different was of evaluating the resulting models: BIC and a k-holdout strategy. The Bayesian Information Criterion is one method that is used for model selection (Raftery, 1995) that tries to balance goodness of fit with the number of parameters used in the model. Intuitively, BIC, penalizes models that have more parameters. Differences in BIC greater than 6 between models are said to be strong evidence while differences of greater than 10 is said to be very strong (See Baker, Corbett

[6] As with any modeling problem, the order in which parameters are added or removed can influence the final resulting model. Future work could investigate the sensitivity of these models to see if they are stable.

& Koedinger (2003) for another example of cognitive model selection using BIC for model selection in this way.)

We also used a k-holdout strategy that worked as follows. The standard way of predicting the error rate of a model given a single, fixed sample is to use a stratified k-fold cross-validation (we choose k=10). Stratification is simply the process of randomly selecting the instances used for training and testing. Because the model we are trying to build makes use of a student's successive attempts, it seemed sensible to randomly select whole students rather than individual instances. Ten fold implies the training and testing procedure occurs ten times. The stratification process created a testing set by randomly selecting one-tenth of the students not having appeared in a prior testing set. This procedure was repeated ten times in order to have included each student in a testing set exactly once.

A model was then constructed for each of the training sets using a logistic regression with the student response as the dependent variable. Each fitted model was used to predict the student response on the corresponding testing set. The prediction for each instance can be interpreted as the model's fit probability that a student's response was correct (indicated by a "1"). To associate the classification with the bivariate class attribute, the prediction was rounded up or down depending if it was greater or less than 0.5. The predictions were then compared to the actual responses by dividing the number of correctly classified instances by the total number of instances to determine the overall classification accuracy on the testing set.

## 2.3 Results

We summarize the results of our model construction, with Table 3 showing the results of models we attempted to construct. To answer Question 1, we compared the *Base Model* to the *Base+ Model* that added the *articulate one-step* KC. After applying our criterion for eliminating non-statistically significant parameters we were left with just two difficulty parameters for the *Base Model* (all models in Table 3 also had the very statistically significant pretest parameter).

| | **Models** | | | | | |
|---|---|---|---|---|---|---|
| | | | Composition Knowledge Components | | | |
| **Name** | Base | Base + | Arithmetic | Comprehension | Articulation | Symbolization |
| **Model #** | 0 | 1 | 2 | 3 | 4 | 5 |
| **BIC** | 2508.9 | 2493.7 | 2504.9 | 2496.9 | Same as Base+ | Same as Base+ |
| **Overall Evaluation** | 59.6% | 64.3% | 65.0% | 64.6% | | |
| **KCs** | Comprehending One-Step | Articulating One-Step | Articulating-One-Step | Articulating One-Step | | |
| | Articulating Variable | Articulating Variable | Articulating Variable | Articulating Variable | | |
| | | Arithmetic | Composition of Comprehension | Composition of Arithmetic | | |
| | | | Composition of Comprehension | Composition of Arithmetic | | |

**Table 3 Models Computed: BIC and K-holdout evaluation, and the KCs for each unique model**

It turned out that the *Base+Model* did a better statistically significant better job (smaller BIC are better) than the *Base Model* in terms of BIC (the difference was great than 10 BIC points suggesting a statistically significant difference). The *Base+ Model* also did better when using the K-holdout strategy (59.6% vs 64.3%). We see from Table 3 that the *Base+ Model* eliminated the *comprehending one-step* KC and added instead the *articulating one-step* and *arithmetic* KCs suggesting that "articulating" does a better job than comprehension as the way to model the difficulty in symbolizing algebra word problems.

So after concluding that there was good evidence for *articulating one-step*, we then computed Models 2-4. We found that two of the four ways of trying to model composition resulted in models that were inferior in terms of BIC and not much different in terms of the K-holdout strategies. We found that models 4 and 5 were reduced to the *Base+ Model* by the step-wise elimination procedure. We also tried to calculate the effect of combining any two of the four composition KCs but all such attempts were reduced by the step-wise elimination procedure to already found models. This suggests that for the set of tutorial log files we used, there was not sufficient evidence to argue for the *composition of articulation* over other ways of modeling the composition effect.

It should be noted that while none of the learning parameters of any of the knowledge components were in any of the final models (thus creating models that predict no learning over time[7]) that on models 4 and 5, the last parameter that was eliminated was a *learning* parameters that both had t-test values that were within a very small margin of being statistically significant (t=1.97 and t=1.84). It should also be noted that in Heffernan (2003) the learning within Experiment 3 was only close to being statistically significant. That might explain why we do not find any statistically significant learning parameters.

We feel that Question 1 ("Is there evidence from tutorial log files that support the conjecture that the *articulating one-step* KC really exists?") is answered in the affirmative, but Question 2 ("What is the best way to model the composition effect") has not been answered definitely either way. All of the models that tried to explicitly model a composition KC did not lead to significantly better models. So it is still an open question of how to best model the composition effect.[8]

## 2.4 Conclusion

This paper presented a methodology for evaluating models of transfer. Using this methodology we have been able to compare different plausible models. We think that this method of constructing transfer models and checking for parsimonious models against student data is a powerful tool for building cognitive models.

A limitation of this techniques is that the results depend on what curriculum (i.e., the problems presented to students, and the order in which that happened) the students were presented with during their course of study. If students were presented with a

[7] The procedure used for model evaluation required a great deal of parsimony, eliminating those modeling parameters that were not statistically significant. Most of the learning parameters were found to be insignificant, which we suspect is partially due to the small sampling period with an average problem solving time of only 14 minutes.

[8] If we had tried to predict response time as opposed to error rate, the results may have been different. Clearly this is an area for future work, as predicting response time was not examined.

different sequence of problems, then there is no guarantee of being able to draw the same conclusions.

We think that transfer models are important tool to use in building and designing cognitive models, particularly where learning and transfer are of interest. We think that this methodology makes a few reasonable assumptions (the most important being the Power Law of Learning). We think the results in this paper show that this methodology could be used to answer interesting cognitive science questions.

# 3 Web-Based Evaluations Showing Differential Learning for Tutorial Strategies Employed by the Ms. Lindquist Tutor

In a previous study, Heffernan & Koedinger (2002) reported upon the Ms. Lindquist tutoring system that uses dialog and Heffernan (2003) conducted a web-based evaluation. The previous evaluation considered students coming from three separate teachers and analyzed the individual learning gains based on the number of problems completed depending on the tutoring strategy provided. This paper examines a set of new web-based experiments. One set of experiments is targeted at determining if a differential learning gain exists between two of the tutoring strategies provided. Another set of experiments is used to determine if student motivation is dependent on the tutoring strategy. We replicate some findings from Heffernan (2003) with regard to the learning and motivation benefits of Ms Lindquist's intelligent tutorial dialog. These experiments related to learning report on over 1,000 participants contributing at most 20 minutes each, for a grand total of over 200+ combined student hours.

## 3.1 Introduction

Several groups of researchers are working on incorporating dialog into tutoring systems: for instance, CIRCSIM-tutor (2002), AutoTutor (Graesser et al., 2000), the PACT Geometry Tutor (Aleven et al., 2001), and Atlas-Andes (Rosé et al., 2001). The value of dialog in learning is still controversial because dialog takes up precious time that might be better spent telling students the answer and moving on to another problem.

In previous work, Heffernan & Koedinger (2002) reported upon the Ms. Lindquist tutoring system that uses dialog and Heffernan (2003) conducted a web-based evaluation using the students from one classroom teacher. This paper reports upon some additional web-based evaluations using the students from multiple teachers. Ms. Lindquist was the first model-tracing tutor that had both a model of student thinking and a model on tutorial planning (Heffernan, 2001). The Ms. Lindquist tutoring system helps students become proficient in writing expressions for algebra word problems. This system is of the "coached practice" variety that does not offer explicit instruction (i.e., long web pages or lectures), but instead is meant to scaffold "learning by doing" while students practice their problem solving skills. An assumption in the development of this system was that students would learn more if they could have an intelligent dialog rather then simply being told the answer. In earlier work (Heffernan, 2001) it was shown that students learned more when they engaged in a dialog with the Ms. Lindquist tutoring system, but did not investigate if it was worth the extra time spent. Heffernan (2003) reported some web-based results that suggested a motivation benefit as some learning benefits might come from using these more intelligent dialogs that Ms. Lindquist provided, but the evidence was not unequivocal.

This paper reports the latest results of experiments that were conducted with students using the tutor via the internet (*www.AlgebraTutor.org*). Each student who visits the web site is informed that they are taking part in an experiment. Then they are randomly assigned to a condition in the experiment. There are several experimental conditions in these experiments, each of them representing different types of "intelligent" feedback that involve a tutorial dialog. These tutorial strategies are based upon the

observation of an experienced tutor (Heffernan, 2001). For instance, the "Concrete Articulation" strategy conducts a dialog based upon the idea of using a concrete instance. The following dialog is a necessarily-short dialog example of the Concrete Articulation Strategy:

> **Tutor**: In the marching band there are 7 fewer flute players than drummers. If the number of drummers is "x", write an expression for the total number of flute players and drummers.
>
> **Student: 7-x+x** {Correct answer is x+(x-7)}
>
> **Tutor**: No. We will come back to the total number of flute players and drummers. Let me try to break this down for you. Please compute the number of flute players if the number of drummers was 10 drummers? {Note: Ms Lindquist notices that the student computed the number of drummers wrong, so she focuses in on that subgoal.}
>
> **Student: 10-7**
>
> **Tutor**: Correct. Now use "x" instead of 10 to again express the number of flute players.
>
> **Student: x-7**
>
> **Tutor**: Correct. Now, lets try this question again: Please write an expression for the total number of flute players and drummers.
>
> **Student: x-7+x**
>
> **Tutor**: Correct. Let us go onto the next problem.

The next example will illustrate the "Verbal" strategy that Ms. Lindquist uses based on the idea that students are more accustomed to expressing ideas in English than in algebraic expressions. It makes sense to first ask the student to explain in English how to achieve a goal and then to ask them how to do it using algebraic symbols afterwards. The following dialog illustrates the Verbal Strategy:

> **Tutor**: Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now?
>
> **Student: 40*m** {Correct answer is 800-40*m}
>
> **Tutor**: Hmm. No. We will come back to the distance Anne has left to row. Let me try to break this down for you. Can you explain how you would find the distance rowed so far?
>
> **Student**: The distance rowed so far is equal to the number of minutes rowing times the distance Anne has left to row (see Table 4).
>
> **Tutor**: Hmm. No. Figure out some way of using both her speed rowing and the number of minutes rowing.
>
> **Student**: The distance rowed so far is equal to her speed rowing times the number of minutes rowing (see Table 4).

**Tutor**: Correct. Now, follow that up by writing an expression for the distance rowed so far.

**Student**: **40*m**

**Tutor**: Correct. Now, lets try this question again: Please write an expression for the distance Anne has left to row.

**Student: 800-40*m**

**Tutor**: Correct. A new problem is needed.

Table 4 provides an example of the menu system employed by the Ms. Lindquist verbal strategy, which prompts a student to provide an English expression. The menu used by the above dialog consists of three drop-down boxes containing the following items:

| [Select a phrase] | [Select an operator] | [Select a phrase] |
|---|---|---|
| her speed rowing | Plus | her speed rowing |
| the distance Anne has left to row | Minus | the distance Anne has left to row |
| the distance rowed so far | Times | the distance rowed so far |
| the distance she started from the dock | Divided by | the distance she started from the dock |
| the distance she started from the dock | | the distance she started from the dock |

**Table 4 Example of Ms. Lindquist menu for verbal strategy**

The experiments reported upon in this paper mainly pertain to the Concrete Articulation Strategy, but the Ms. Lindquist tutoring system is quite complicated and has several different pedagogical strategies. Please refer to Heffernan & Koedinger (2002) for more information on Ms. Lindquist including other more interesting dialog examples.

The control condition in all of these experiments is to simply tell the student the correct answer if they make a mistake (i.e., "No. A correct answer is 5m-100. Please type that.") If a student does not make an error on a problem, and therefore receives no corrective feedback of any sort, then the student has not participated in either the control condition or the experimental condition for that problem. For each experiment "time on task" is controlled, whereby a student is given problems until a timer has gone-off and then is advanced to a posttest after completing the problem they are currently working on.

The Ms. Lindquist's curriculum is composed of five sections, starting with relatively easy one-operator problems (i.e., "5x"), and progressing up to problems that need four or more mathematical operations to correctly symbolize (i.e., "5x+7*(30-x)"). Few students make it to the fifth section, so the experiments we report on are only in the first two curriculum sections. At the beginning of each curriculum section, a tutorial feedback strategy is selected that will be used throughout the exercise whenever the student needs assistance. Because of this setup, each student can participate in five separate experiments, one for each curriculum section. We would like to learn which tutorial strategy is most effective for each curriculum area

Since its inception in September 2000, over 17,000 individuals have logged into the tutoring system via the website, and hundreds of individuals have stuck around long enough (e.g., 30 minutes) to provide potentially useful data. The system's architecture is constructed in such a way that a user downloads a web page with a Java applet on it, which communicates to a server located at Carnegie Mellon University. Students'

responses are logged into files for later analysis. Individuals are asked to identify themselves as a student, teacher, parent or researcher. We collect no identifying information from students. Students are asked to make up a login name that is used to identify them if they return at a later time. Students are asked to specify how much math background they have. We anticipate that some teachers will log in and pretend to be a student, which will add additional variance to the data we collect, thereby making it harder to figure out what strategies are most effective; therefore, we also ask at the end of each curriculum section if we should use their data (i.e., did they get help from a teacher, or are they really not a student). Such individuals are removed from any analyses. We recognize that there will probably be more noise in web based experiments due to the fact that individuals will vary far more than would normally occur in individual classroom experiments (Ms. Lindquist is used by many college students trying to brush up on their algebra, as well as by some students just starting algebra), nevertheless, we believe that there is still the potential for conducting experiments studying student learning. Even though the variation between individuals will be higher, thus introducing more noise into the data, we will be able to compensate for this by generalizing over a larger number of students than would be possible in traditional laboratory studies.

In all of the experiments described below, the items within a curriculum section were randomly chosen from a set of problems for that section (usually 20-40 such problems per section). The posttest items (which are the exact same as the pretest items) were fixed (i.e., all students received the same two-item posttest for the first section, as well as the same three-item posttest for the second section, etc.) We will now present the experiments we performed.

## 3.2 Experiments: Differential Learning

Thirteen experiments were conducted to see if there was a difference in learning gain (measured by the difference in posttest and pretest score) according to the tutoring strategy provided by the tutor. To determine if the difference in learning gain between the tutoring strategies was statistically significant an ANOVA was conducted. The measure of learning gain was considered to be a "lightweight" evaluation due to the brevity of the pretest and posttest.

Each experiment involved two tutoring strategies given at random to a group of students. Each student participating in the experiment answered at least one problem incorrectly during the curriculum section, causing the tutor to intervene. Students receiving a perfect pretest were eliminated from some of the experiments in an attempt to eliminate the "ceiling effect" caused by the shortness of the pretest and the large number of students scoring perfectly.

The experiments can be divided into two groups, the first examining the difference between the Inductive Support (IS) and Cut-to-the-chase (Cut) strategy and the second examining the difference between the IS and Verbal strategy. If students reported that they were students and were required to use the tutor, they were given either the IS or Cut strategy (we consider these students to be in the "forced" group). If students reported that they were students and were not required to use the tutor, they were given either the IS or Verbal strategy (these students are referred to as the "non-forced" group). Each experiment was conducted over a single curriculum section. In some cases there were multiple experiments for the same curriculum section and strategy comparison, which was made possible by having several large but distinct sets of students coming from

different versions of the tutor where time on task had been modified. The thirteen experiments, which are indicated in Table 6, will now be described along with their results.

### 3.2.1 Experiment 1 & 2 (Section 1, Verbal vs. Cut)

An early version of the tutor provided the Verbal and Cut strategies on Section 1 to forced students, so these two experiments are based on those students. In experiment 1, 64 students received Verbal, whereas 87 students received Cut. Since approximately 2/3 of these students obtained a perfect pretest, experiment 2 was conducted with the same students, but removing those students receiving a perfect pretest. The reason for keeping the first experiment is that reporting on overall learning is only possible if all students taking the pretest are accounted for even if they received a perfect score. Due to the large number of students receiving a perfect pretest, it is obvious that a longer pretest would have helped eliminate this problem, but may also have reduced the number of students completing the entire curriculum section.

### 3.2.2 Results for Experiment 1 & 2

The first experiment showed no evidence of a differential learning gain between the Verbal and Cut strategies with the learning gain for Verbal being 13% and for Cut being 14%. This was not surprising since 2/3 of the students had received a perfect pretest, which was our motivation for creating experiment 2, having those students eliminated. For the second experiment, there was also no evidence of a differential learning gain, although the learning gain for Verbal was 41% and Cut was 35%. For each of these experiments the number of problems solved by strategy was statistically different ($p<.0001$). This is not particularly surprising as the Cut strategy simply provides the correct answer, whereas the Verbal strategy is more time consuming by using menus and intelligent dialog, which results in fewer problems being completed on average. Another observation is that the time on task for each strategy was statistically different ($p<.0001$). This is explained by a design decision to allow students to finish the problem they are working on before advancing to the posttest, which means more time consuming tutoring strategies result in a slightly longer average time on task.

### 3.2.3 Experiment 3 (section 1, IS vs. Cut)

For the first section forced students, the IS and Cut strategies were provided on the latest version of the tutor. Although the number of forced students was substantially less than non-forced students (due to the tutor being available online rather than used just in a classroom setting), both experimental conditions had over 60 students. Only enough data was available for a single experiment on the first section involving the IS and Cut strategies since the tutor previously provided the Verbal and Cut strategies on that section as seen in Experiments 1 and 2.

### 3.2.4 Results for Experiment 3

For this experiment, a differential learning gain between the IS and Cut strategy was observed as being statistically significant ($P=.0224$). Students with the IS strategy had a learning gain of 53% and those with the Cut strategy 36%. The pretest scores were surprising in that those students given the IS strategy had a lower score, on average 22% correct with those given the Cut strategy having on average 34% correct. Interestingly, the students given the IS strategy not only had lower performance on the pretest, but also

had a higher performance on the posttest, which explains the statistically significant learning gain observed.

### 3.2.5 Experiment 4 & 5 (section 2, IS vs. Cut)

On the second curriculum section, the IS and Cut strategies were given to the forced students. Two experiments were conducted using a set of students that were controlled for time. The students in Experiment 5 were given twice as much time as those in Experiment 4 (1200 seconds vs. 600 seconds).

### 3.2.6 Results for Experiment 4 & 5

Both Experiment 4 and 5 showed no evidence of differential learning by tutoring strategy. For Experiment 4, the learning gain was 18% for IS and 14% for Cut. This is confounded by the learning gain on Experiment 5, which was 19% for IS and 23% for Cut. Since both experiments contained relatively few students in each condition, it is not surprising that the results from Experiment 4 and 5 would be contradictory.

### 3.2.7 Experiment 6-11 (section 1, IS vs. Verbal)

These six experiments compared differential learning for the IS and Verbal strategies on the first section, which were given to non-forced students. It was noticed for Experiment 6 that approximately 2/3 of the students received a perfect pretest. To prevent a ceiling effect of student's not demonstrating learning, those students receiving a perfect pretest were removed to produce Experiment 7. Experiment 8 involved a much smaller group of students (approximately 30 per condition) receiving the same amount of time on Section 1 as those in the previous experiment. Although the students in Experiment 8 had very high pretest scores, those students receiving a perfect score were not removed due to the much smaller sample size. Experiments 9 and 10 both involved separate groups of students who had those students receiving a perfect pretest removed from the sample. Experiment 11 was the combination of the students from
Experiments 9 and 10, as both of those experiments provided the same time on task.

### 3.2.8 Results for Experiment 6-11

Experiments 6-11 all showed that students given the IS strategy as having a higher learning gain than those receiving the Verbal strategy. Experiment 8 had a p-value suggesting the difference in learning gain was not statistically significant, which could partially be explained by the small sample size (approximately 30 students given each condition) and due to the high pretest scores (75% for IS and 84% for Cut), which resulted in a ceiling-effect. Looking at the posttest scores, those given IS received 89% correct, whereas those given Cut received 93% correct. It should be noted that Experiment 11, which was the combination of students from Experiments 9 and 10 increased the statistical significance for learning gain from (P=0.1030) and (P=0.0803) consecutively to (P=0.0210).

### 3.2.9 Experiments 12 & 13 (section 2, IS vs. Verbal)

These two experiments compared differential learning for the IS and Verbal strategies on the second section, which were given to non-forced students. Both experiments involved a separate group of students having a different time on task. In Experiment 12 the average problem solving time was approximately 700 seconds, whereas in Experiment 13 the average problem solving time was approximately 1200 seconds. The sample size

used for experiment 13 (having approximately 100 students) contained almost twice as many students as that used for experiment 13.

### 3.2.10 Results for Experiment 12 & 13

Experiments 12 and 13, which are both on the second curriculum section did not show statistical evidence of differential learning gain. For Experiment 12 the learning gain of those students given the IS strategy, which was 22% was slightly higher than those students given the Cut strategy, which was 18%. The results of Experiment 13, which had twice the number of students and double the amount of time on task had a learning gain of 30% for those given the IS strategy and 33% for those given the Cut strategy. Although the difference in learning gain was insignificant for both of these experiments, it was odd that such a large number of students would show nothing significant after 20 minutes of problem solving. It was observed that the pretest score between condition in Experiment 13 was statistically significant (P=.0465), which indicates that the lightweight evaluation method may be partially responsible.

## 3.3 Experiments: Student Motivation

Four experiments were conducted to determine if there was a difference in student motivation determined by the tutorial strategy (either IS or Verbal) offered by the tutor. For the first experiment, students received either the IS or Verbal strategy on the first section and all students were given the Cut strategy on the second section. We were not particularly interested in student motivation involving the Cut strategy as we already examined this [7] found that if students were given Cut they left the web site at much higher rates. For the second experiment, students received the same tutorial strategy (either IS or Verbal) on both the first and second sections. In this experiment, only students completing the first section and starting work on the second section were analyzed for their completion rate on the second section. The third experiment looked at only those students working on the first section to determine if there was a difference in completion rate for that section. The fourth and final experiment looked at those students skipping the first section due to getting a perfect pretest and starting work on the second section.

The number of students within each control condition is indicated in the *count* column of Table 5. Experiment 3 has the largest number of students, because it included those students starting work on section 1, which is the majority of students. Experiment 4 also contains a large number of students, which results from a large number of students skipping the first section due to a perfect pretest. An ANOVA was conducted for each of the four experiments to see if the difference in motivation (section completion rate) by tutorial strategy was statistically significant.

### 3.3.1 Results of Student Motivation

For the first experiment, with approximately 150 students in each condition, the percent of students completing the second section was 50% and 49% for IS and Verbal consecutively which was not statistically different. For the second experiment, with approximately 65 students in each condition, the section completion rate was 55% and 65% for IS and Verbal consecutively, which was also not statistically different. The third and four experiments contained an even larger number of students, but for both of these

experiments no difference in motivation was seen for the given tutorial strategy. The motivation experiments are summarized in Table 5.

| | Count | | Section Completed (%) | | P-Value |
|---|---|---|---|---|---|
| Experiment | IS | Verbal | IS | Verbal | |
| 1 | 189 | 121 | 50 | 49 | .6166 |
| 2 | 73 | 60 | 55 | 65 | .2362 |
| 3 | 697 | 428 | 61 | 59 | .6005 |
| 4 | 275 | 187 | 47 | 51 | .4123 |

**Table 5 Experiments on Student Motivation**

From these four experiments, it would appear that student motivation is not influenced by giving either the IS or Verbal strategy. Possibly student motivation is not seen, because students starting the second section after finishing the first have nearly the same motivation. It would be interesting to see if student motivation on the first section was dependent on strategy given, which will most likely be looked at in a future study.

## 3.4 Discussion: Web-Based Experiments

However, these results should be taken with a grain of salt given that students are taking a two or three item pretest and posttest, which is due to our decision to provide only a lightweight evaluation as previously mentioned. Web-based evaluation for the most part makes this lightweight evaluation useful given the large collection of data that is produced.

## 3.5 Conclusion

In earlier work, Heffernan (2001) presented evidence that suggested that students learned more when they engaged in a dialog with the Ms. Lindquist tutoring system, but did not investigate if it was worth the extra time spent. Heffernan (2003) reported some web-based results that suggested that the Cut to the chase strategy was inferior to the IS strategy in terms of learning gain.

From the experiments reported in this work conducted on differential learning by tutorial strategy, it appears that the benefit to using one strategy over another is sometimes seen on the first curriculum section. In particular, experiment three is something of a replication of the work from Heffernan (2003). This could partially be explained by the tutorial dialogs on the second section being longer and requiring more time to read. It should be noted that a student can spend a great deal of time on a single problem, and these results are making us consider setting a time cut-off for a dialog so that students don't spend too much time on any one dialog.

Next we turn to comparing IS with Verbal. It appears that providing the IS strategy is a better choice than Verbal on the first curriculum section as seen by the significant difference in learning gain on experiment 7, 9, 10 and 11. We were pleasantly surprised that we could detect differences in learning rates in only 8-10 minutes suing such crude (2 item pre and posttests).

The strong evidence for the IS strategy being better than the Cut strategy was not particularly surprising. Heffernan (2003) previously reported seeing a similar result, but this was for students working on the second curriculum section. We have to study this further to better understand these results.

Finally, it should be reiterated that no differences in motivation could be found between the IS and Verbal strategies. This could possibly be explained by both of these strategies being advanced, in that they keep a participant more involved than the naïve Cut strategy. This result is also consistent with Heffernan (2003). Given that students seemed to learn a little better with the IS strategy than the Verbal strategy, we thought we might see a motivation benefit for the IS strategy, but we did not.

# 4 Future Directions

The methodology for constructing and evaluating transfer models described in Chapter 2 can be used to improve the cognitive model used by the Ms. Lindquist tutor by finding new knowledge components. Doing this requires analyzing more student log files and comparing more models in a semi-automated manner. A model search implemented by Jonathan Freyberger and Larissa Orlova makes use of the described methodology, but incorporates an intelligent search to find the "best fitting" transfer model. Using the concept of a transfer model is also being employed for the Assistments project (*www.assistment.com*), a research project funded by the US Department of Education to build computer based assisting into an assessment system.

From the web-based evaluation in Chapter 3, we can see the usefulness of conducting web-based evaluations, which suggests that further studies should investigate other tutoring systems. Also formalizing procedures for comparing learning rates and motivation for various tutoring strategies lends to future work.

Many concepts were investigated during this thesis that would be ideal topics for further study. The first topic to be mentioned is the Ski-Slope Metric (see Appendix B.4), which is a concept mentioned to my advisor by Ken Koedinger. Another topic relates to that of problem difficulty (see Appendix C), which consists of analyzing the individual problems to determine why some are substantially more difficult than others. Another topic of study relates to better accounting for student errors (see Appendix D), which involves clustering or classifying common mistakes. Improving the cognitive model to detect and respond to these pitfalls would benefit the tutoring system as well as enhance our understanding of the difficulty associated with solving algebra word problems. The last topic to be mentioned relates to using the concept of "model mispredictions" (see Appendix E), which could potentially be used to find problems that require a modification to the current model. The approach to using the model's mispredictions for stepwise refinement is in part investigated by Jonathan Freyberger who mines the incorrectly classified instances using the AprioriSetsAndSequences algorithm as part of the intelligent search used to find the best fitting transfer model.

# 5 Final Remarks

This thesis has addressed the problem of constructing and evaluating models that predict learning transfer. In the process of investigating the methodology employed, a log file parser was written for the Ms. Lindquist tutor. Additionally code for constructing and evaluating models based on various transfer models had to be written. The parser facilitated the web-based evaluation and also made it possible to easily determine the individual problem difficulties (see Appendix C), the most common incorrect responses (see Appendix D) and model mispredictions (see Appendix E).

Producing models of student learning has greatly increased my knowledge of statistical modeling techniques and I have benefited from spending over a year working with the S-Plus statistical software package. Also my understanding of Intelligent Tutoring Systems has greatly increased by routinely analyzing the interactions of those students working with the Ms. Lindquist tutor.

Developing techniques for constructing and evaluating transfer models is essential to the software lifecycle of an Intelligent Tutoring System as it aims to improve the existing cognitive model from such a system. The task of analyzing tutorial strategies is also of great interest as it is essential to refining existing tutorial strategies and creating new tutorial strategies. The methods employed by this research will surely lead to producing better tutoring systems that are able to adapt their tutorial strategies based on the results from tutoring students.

| Exp. | | Sec | Str1 | Str2 | Count | | Problems Solved | | | Time (sec) | | | Pretest Correct (%) | | | Posttest Correct (%) | | | Learning Gain (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Str1 | Str2 | Str1 | Str2 | P-Value | Str1 | Str2 | P-Value | Str1 | Str2 | P-Value | Str1 | Str2 | P-Value | Str1 | Str2 | P-Value |
| Forced | 1 | 1 | Verbal | Cut | 64 | 87 | 4.9 | 7.9 | <.0001 | 477 | 404 | <.0001 | 71 | 65 | .3151 | 84 | 79 | .3258 | 13 | 14 | .8396 |
| | 2 | 1 | Verbal | Cut | 27 | 47 | 3.5 | 6.5 | <.0001 | 498 | 407 | <.0001 | 32 | 35 | .5278 | 72 | 70 | .8035 | 41 | 35 | .5328 |
| | 3 | 1 | IS | Cut | 61 | 68 | 7.4 | 11.2 | <.0001 | 602 | 593 | .6695 | 22 | 34 | .0072 | 74 | 70 | .4262 | 53 | 36 | .0224 |
| | 4 | 2 | IS | Cut | 33 | 39 | 4.5 | 6.4 | .0089 | 639 | 580 | .1436 | 43 | 44 | .9070 | 62 | 58 | .6750 | 18 | 14 | .5234 |
| | 5 | 2 | IS | Cut | 26 | 35 | 9.7 | 14.3 | .0066 | 1241 | 1199 | .4469 | 47 | 43 | .4841 | 67 | 65 | .8073 | 19 | 23 | .7382 |
| | | | | | | | | | | | | | | | | | | | | | |
| Non-Forced | 6 | 1 | IS | Verbal | 189 | 173 | 5.8 | 4.0 | <.0001 | 425 | 495 | <.0001 | 52 | 55 | .3815 | 83 | 81 | .5162 | 31 | 25 | .2014 |
| | 7 | 1 | IS | Verbal | 117 | 105 | 5.4 | 3.4 | <.0001 | 421 | 484 | .0005 | 21 | 26 | .1520 | 77 | 72 | .2253 | 56 | 46 | .0560 |
| | 8 | 1 | IS | Verbal | 30 | 28 | 7.4 | 5.4 | .0180 | 394 | 477 | .0015 | 75 | 84 | .3192 | 89 | 93 | .3887 | 13 | 9 | .5946 |
| | 9 | 1 | IS | Verbal | 60 | 32 | 6.5 | 5.5 | .2265 | 630 | 713 | .0141 | 22 | 24 | .7481 | 73 | 95 | .0602 | 51 | 36 | .1030 |
| | 10 | 1 | IS | Verbal | 39 | 35 | 10.2 | 7.3 | .0298 | 640 | 641 | .9618 | 31 | 41 | .0429 | 85 | 82 | .6402 | 54 | 40 | .0803 |
| | 11 | 1 | IS | Verbal | 99 | 67 | 8.0 | 6.5 | .0531 | 634 | 676 | .0838 | 25 | 33 | .0535 | 77 | 71 | .2016 | 52 | 38 | .0210 |
| | 12 | 2 | IS | Verbal | 55 | 65 | 4.1 | 3.6 | .1994 | 786 | 723 | .1441 | 35 | 42 | .2519 | 58 | 60 | .7205 | 22 | 18 | .4271 |
| | 13 | 2 | IS | Verbal | 116 | 90 | 10.0 | 7.7 | .0039 | 1250 | 1290 | .3029 | 43 | 36 | .0465 | 73 | 69 | .2986 | 30 | 33 | .5298 |

**Table 6 Experiments on Differential Learning Gain**

# References


Aleven V., Popescu, O. & Koedinger, K. R. (2001). Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In Moore, Redfield, & Johnson (Eds.), *Proceedings of Artificial Intelligence in Education 2001.* Amsterdam: IOS Press.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought.* Lawrence Erlbaum Associates, Mahwah, NJ.

Baker, R.S., Corbett, A.T., & Koedinger, K.R. (2003). Statistical Techniques for Comparing ACT-R Models of Cognitive Performance. Presented at 10th Annual ACT-R Workshop.

Birnbaum, M.H. (Ed.). (2000). "Psychological Experiments on the Internet." San Diego: Academic Press. http://psych.fullerton.edu/mbirnbaum/web/IntroWeb.htm

CIRCSIM-Tutor (2002). (See http://www.csam.iit.edu/~circsim/)

Corbett, A. T. & Anderson, J. A. (1992). Knowledge tracing in the ACT programming tutor. In: Proceedings of 14-th Annual Conference of the Cognitive Science Society.

Corbett, A. T., Anderson, J. R., & O'Brien, A. T. (1995). Student modeling in the ACT programming tutor. Chapter 2 in P. Nichols, S. Chipman, & R. Brennan, *Cognitively Diagnostic Assessment*. Hillsdale, NJ: Erlbaum.

Draney, K. L., Pirolli, P., & Wilson, M. (1995). A measurement model for a complex cognitive skill. In P. Nichols, S. Chipman, & R. Brennan, *Cognitively Diagnostic Assessment.* Hillsdale, NJ: Erlbaum.

Embretson, S. E. & Reise, S. P. (2000). Item Response Theory for Psychologists Lawrence Erlbaum Assoc.

Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., & the TRG (in press). (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*.

Heffernan, N. T. & Croteau, E. A. (2003). A Methodology for Evaluating Predictions of Transfer and an Empirical Application to Data from a Web-Based Intelligent Tutoring System: How to Improve Knowledge Tracing in Dialog Based Tutors. Worcester Polytechnic Institute Technical Report: WPI-CS-TR-03-27.

Heffernan, N. T. (2003). Web-Based Evaluations Showing both Cognitive and Motivational Benefits of the Ms. Lindquist Tutor. *11th International Conference Artificial Intelligence in Education. Syndey*. Australia.

Heffernan, N. T., & Koedinger, K. R. (2002). An intelligent tutoring system incorporating a model of an experienced human tutor. *Sixth International Conference on Intelligent Tutoring Systems*.

Heffernan, N. T. (2001). *Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of an Experienced Human Tutor.* Dissertation & Technical Report. Carnegie Mellon University, Computer Science, http://www.algebratutor.org/pubs.html.

Heffernan, N. T., & Koedinger, K. R. (1998). A developmental model for algebra symbolization: The results of a difficulty factors assessment. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Societ*y, (pp. 484-489) Hillsdale, NJ: Lawrence Erlbaum Associates.

Heffernan, N. T., & Koedinger, K. R. (1997). The composition effect in symbolizing: the role of symbol production versus text comprehension. In *Proceeding of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 307-312). Hillsdale, NJ: Lawrence Erlbaum Associates.

Junker, B., Koedinger, K. R., & Trottini, M. (2000). Finding improvements in student models for intelligent tutoring systems via variable selection for a linear logistic test model. Presented at the Annual North American Meeting of the Psychometric Society, Vancouver, BC, Canada. http://lib.stat.cmu.edu/~brian/bjtrs.html

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.

Laurillard, D. (1993). *Rethinking university teaching: A framework for the effective use of educational technology*. London: Routledge.

Nathan, M. J., Kintsch, W. & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning environments. *Cognition & Instruction* 9(4): 329-389.

Newell, A. & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In Anderson (ed.), *Cognitive Skills and Their Acquisition*., Hillsdale, NJ: Erlbaum.

Raftery, A.E. (1995). Bayesian model selection in social research. *Sociological Methodology* (Peter V. Marsden, ed.), Cambridge, MA: Blackwells, pp. 111-196.

Rosé, C., Jordan, P., Ringenberg, M, Siler, S., VanLehn, K. and Weinstein, A. (2001). Interactive conceptual Tutoring in Atlas-Andes. In *Proceedings of AI in Education 2001 Conference*.

# Appendix A: Integrating JAVA with S-PLUS

The following software configuration was used to integrate Java with S-Plus:

- Insight S-PLUS 6.1 for Windows, Student Edition, Release 1
- JAVA version 1.4.1_02
- Windows XP Home

The Student Edition of S-PLUS can be obtained from the S-PLUS Student Center.

With S-PLUS 6, Insight provides CONNECT/JAVA, which is intended to make accessing the S-PLUS engine through JAVA a trivial task. This web page details some of the difficulties I encountered and how I overcame these when using CONNECT/JAVA. When I first attempted to integrate JAVA and S-PLUS, I was unable to locate information on the web. I am providing this information as it would have helped me and hopefully will help you with your own project requiring the integration of JAVA with S-PLUS.

**STEP 1**: *Install S-PLUS 6 for Windows*. The home directory for S-PLUS on my computer is "C:\Program Files\Insightful\splus61se". I will use **SHOME** to refer to the S-PLUS home directory, so substitute your path accordingly whenever that variable is used.

**STEP 2**: *Read javadoc for CONNECT/JAVA*. The javadoc can be located at "SHOME\java\javadoc\index.html".

**STEP 3**: *Locate the CONNECT/JAVA class files*. These are the classes referred to in the javadoc and are located at "SHOME\java\jre\lib\ext\Splus.jar". You might want to reference these files later, so extract the jar file (using Winzip or a similar tool) to a directory, such as "C:\STest\src".

**STEP 4**: *Locate the CONNECT/JAVA example files*. These are compiled examples that illustrate concepts using CONNECT/JAVA and are located at "SHOME\java\jre\lib\ext\examples.jar". You might want to reference these files later, so extract the jar file (using Winzip or a similar tool) to a directory, such as "C:\STest\examples".

**STEP 5**: *Verify both JAVA and S-PLUS work independently*. You should be able to type "java" or "splus" from the command line, which would display the java usage screen or the splus application respectively.

**STEP 6**: *Run an example*. Change to the example directory "C:\STest\examples" and type:

```
java -Dsplus.shome="SHOME" -classpath ".;SHOME\java\jre\lib\ext\Splus.jar" TextOutputExample
```

If running the example “TextOutputExample” was successful, you should have seen the following displayed to stdout:

```
Sending expression 1:100
Result from S-PLUS:
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
 [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
 [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
 [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
 [91]  91  92  93  94  95  96  97  98  99 100
```

Note that the system property “splus.shome” needs to be set for CONNECT/JAVA to work properly. Additionally it is necessary to indicate the class path of the jar file containing the classes for CONNECT/JAVA as shown above.

**STEP 7**: *Create your own simple example*.

The two JAVA/CONNECT classes used in the following example are *SplusDataResult* and *SplusUserApp*. Read about both of these in the javadoc mentioned in STEP 2.

```
/* Coercion Example
 * Illustrating the simplicity of CONNECT/JAVA
 *
 * See: S-PLUS 6 for Windows Programmer's Guide, page 23 (Coercion of Values)
 */

import com.insightful.splus.SplusDataResult;
import com.insightful.splus.SplusUserApp;
import java.io.PrintStream;

public class Coercion
{
    public static void main(String args[])
    {
        try
        {
            String s = "c(T, F, pi, 7)";
            System.out.println("Sending \"" + s + "\" to S-PLUS");
            SplusDataResult splusdataresult = SplusUserApp.eval(s + "\n", true, false,
false, false, false);
            System.out.println("Result from S-PLUS:");
            System.out.println(splusdataresult.getOutput());
        }
        catch(Exception exception)
        {
            System.out.println(exception);
        }
        System.exit(0);
    }
}
```

To compile this example type:
javac -classpath ".;SHOME\java\jre\lib\ext\Splus.jar" Coercion.java

To run this example type:
java -Dsplus.shome="SHOME" -classpath ".;SHOME\java\jre\lib\ext\Splus.jar" Coercion

If running this example was successful, you should have seen the following displayed to stdout:

```
Sending "c(T, F, pi, 7)" to S-PLUS
Result from S-PLUS:
[1] 1.000000 0.000000 3.141593 7.000000
```

**Final remarks**: If you have found any of this information useful, please let me know. It appears that CONNECT/JAVA is capable of integrating JAVA with S-PLUS and is certainly easier (in my opinion) than integrating with the CONNECT/C++ library (although this is the library used by CONNECT/JAVA). If you would like to make a contribution to this tutorial, please send email ecroteau@wpi.edu.

# Appendix B: Code Documentation

## B.1 Log file Parser

The Ms. Lindquist log file parser was named *TMiner*, which stands for "Tutor Miner". The goal of the parser was to capture the entire contents of each student file that was available so that the parsed data could be used to model student learning and evaluate various aspects of the tutor. The parser was written in Java™ and consisted of eight primary classes. Each class will be briefly described so that an understanding of the entire parser is conveyed. The discussion will start with the classes having the least number of dependencies on the other classes.

- **MathEvaluator** – The MathEvaluator class is responsible for building and evaluating parse trees for mathematical expressions (i.e. 3x+7-y). This class has no dependencies on the rest of the parser and would be useful for any application requiring the ability to parse and evaluate mathematical expressions.

- **ProblemFile** – The ProblemFile class is responsible for encapsulating the template used by the tutor for a particular problem. Each problem file consists of the dialog that describes the problem and a series of dialogs and expected responses that correspond to each question associated with the problem. Each question has two possible responses, one which contains a variable (requiring the student to symbolize) and the other corresponding to the evaluated expression without variables.

- **Problem** – The Problem class is responsible for encapsulating a single scenario from a student log file. Each problem captures the tutor/student interaction by providing the questions asked to the student and the responses made by the student. Associated with each problem is a tutorial strategy, curriculum section, problem name, problem dialog and the resulting conversation. A conversation consists of a students response, correct response, elapsed time and the dialog itself. This class has dependencies on both the MathEvaluator and ProblemFile classes. The MathEvaluator class is used to determine if the student provided a correct answer for a question different from what was asked by the tutor. Since the tutor provides credit for a student's knowledge of these "higher-level" problems, the MathEvaluator and ProblemFile classes allow the question type to be changed to the question type associated with the response provided by the student.

- **StudentReader** – The StudentReader class is responsible for reading an entire student log file and constructing a problem using the Problem class for each scenario appearing in the student log file. Additionally, this class captures responses to questions provided by the tutor. Examples of questions are if the student was required to use the tutor, how helpful the tutor was and if the student received help from someone while using the tutor. The StudentReader provides several reporting functions, such as being able to determine the difference in score between the posttest and pretest for a given curriculum section.

- **TMiner** – The TMiner class is responsible for creating a StudentReader object for each student file in a particular directory, or for each student listed in a given text file.

- **Filter** – The Filter class is used to exclude certain instances/rows of data from the generated dataset (created by the DataDumper) based on a set of rules. For example, a filter could be used to eliminate pretest/posttest problems from the dataset, all problems for a particular curriculum section or even students not having completed a predetermined number of problems in a section.

- **LFAnalysis** – The LFAnalysis class makes use of a transfer model to provide the DataDumper with the skills required by a question based on the previously answered questions for a given problem scenario. Additionally, this class keeps a running total of successful skill usage, which is required to perform a learning factors analysis.

- **DataDumper** – The DataDumper class is responsible for writing the contents of each StudentReader object read by the TMiner class to a file. The DataDumper iterates over each StudentReader object and writes the student's attempts, required skills and successful skill usage for each problem. The DataDumper is the main entry point for the parser and has dependencies on all of the classes previously listed.

The previous discussion of class dependencies is illustrated by the following figure:

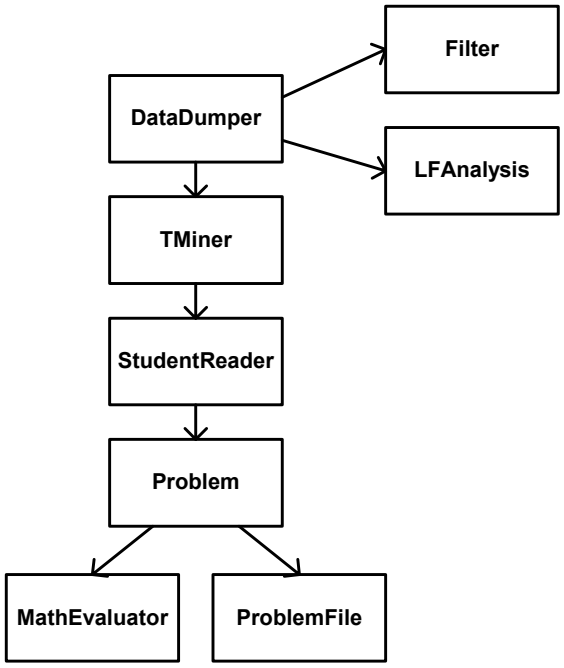


**Figure 1 Class Hierarchy for the Logfile Parser**

### B.1.1 Building the Parser

The parser can be built from the directory containing its source .java files by typing: "javac DataDumper.java".

### B.1.2 Executing the Parser

The parser can be executed within the same directory it was built by typing: "java DataDumper".

### B.1.3 Parser Output

The output of the parser is a file named "test". This file is tab-delimited and currently produces 46 columns. The header for these 46 columns in addition to the methods populating these columns can be located in the source file DataDumper.java.

### B.1.4 Transfer Model

The current transfer model being applied to the parsed log files can be located in the source file LFAnalysis.java. The implemented transfer model contains six question types and has eight knowledge components. **Question types can be added** to the existing transfer model by modifying the method *factorHelper*. **Additional knowledge components** could be added to the existing transfer model by increasing the size of the integer arrays *factor8* and *factor16* such that the size of *factor8* is equal to the number of knowledge components and *factor16* is equal to twice the number of knowledge components. Currently the array *factor8*'s size is eight whereas the array *factor16*'s size is sixteen. The reason for using two arrays with one containing twice the number of knowledge components is due to the "subtracting-out" method used to determine the difficulty knowledge components for a particular question. The method *factorHelper* returns a string indicated by the regular expression: $(0 + 1)^{m*n}$, where *n* is equal to the number of knowledge components and *m* is equal to the highest degree of any knowledge component. The current implementation has $m = 2$, which is the reason *factor16* is twice the size of *factor8*. *m* number of 0's or 1's are specified for each knowledge component, which allows the subtracting out method to provide partial credit for the difficulty knowledge components.

### B.1.5 Directory Structure

The parser's class files are all located in the distribution's highest level directory (e.g. C:\TMiner). This directory has six subdirectories. Only two of those subdirectories (*problems* and *SystemFiles*) are used by the parser. The *problems* subdirectory contains the tutor's problem files and the *SystemFiles* subdirectory contains the student log files. An illustration of the TMiner directory structure is as follows:

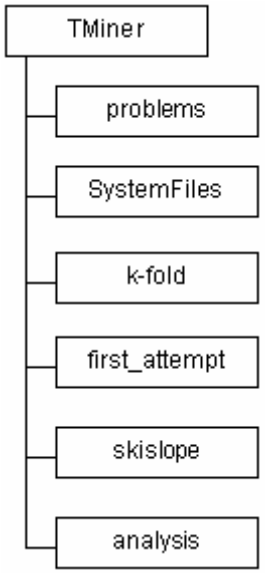


**Figure 2 TMiner Directory Structure**

A partial listing of the files contained in the *problems* and *SystemFiles* directories appear in the following figure. These screenshots serve as a point of reference for someone unfamiliar with the parser directory structure wanting to parse their own collection of student log files. The easiest way to parse your own student files would be to create a new directory under the *TMiner* directory, which contains your student files. The file TMiner.java could then be modified to read the students from that new directory. An alternative would be to simply replace the student files contained in the "SystemFiles" directory with the student files you would like to parse.

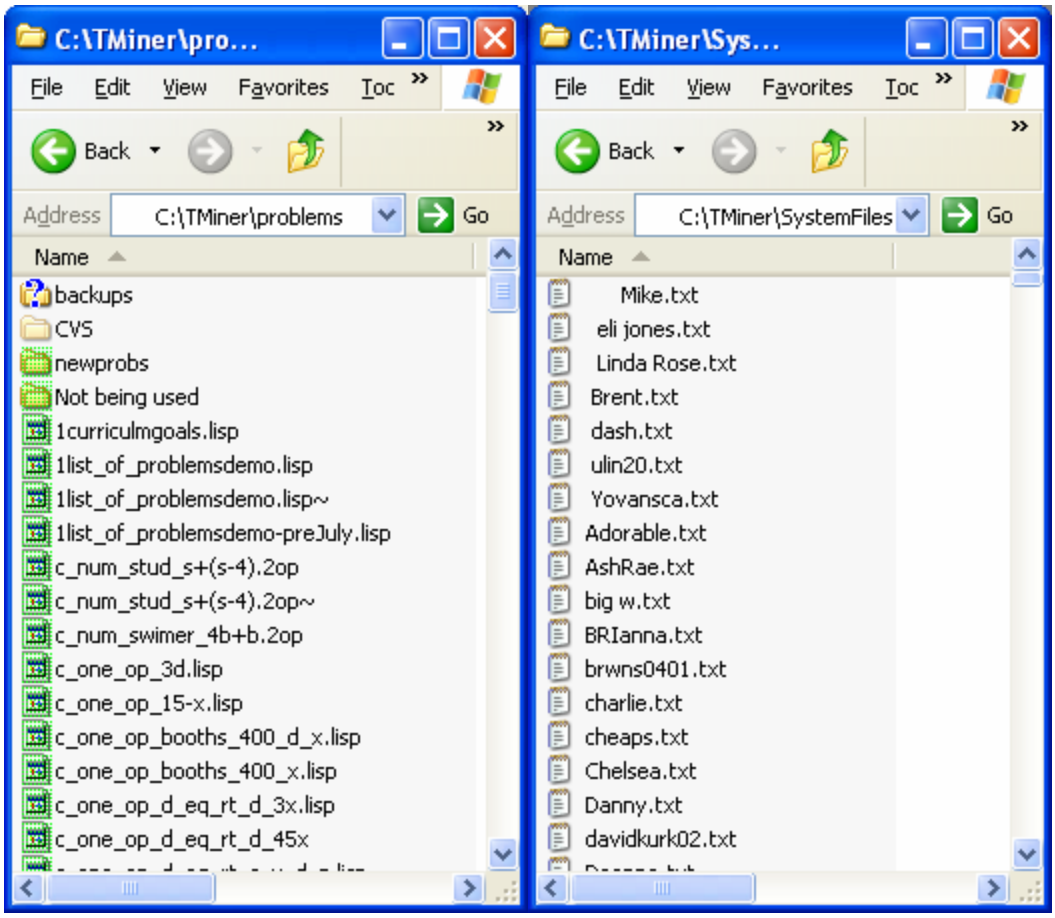


**Figure 3 Partial Directory Listing of TMiner Subdirectories**

## B.2 First Attempt Extractor

The first attempt extractor is located in the *first_attempt* subdirectory within the TMiner distribution. The purpose of this utility is to remove instances from the output file created by the parser such that only instances referred to as "first attempts" remain. A first attempt is defined as being the first time a question is given with a unique set of difficulty factors for scenario problem. If the "subtracting-out" method was not employed, a first attempt would correspond with the first appearance of a question type for a scenario problem. Constructing models that only accounts for a user's first attempts is considered to be standard practice. The first attempt extractor adds an additional column to the dataset, which indicates the number of previous questions (hint levels) seen the current scenario problem. This additional parameter could be used to model the number of hints required to get a correct responses.

The first attempt extractor can be compiled by typing "javac FirstAttempt.java" within the */TMiner/first_attempt* directory. To execute this program, type "java FirstAttempt". It is assumed that the input dataset is named "test" (the output from the parser) and the output dataset containing only the first attempts is called "firstatm.txt". It would be a good idea to make the input and output filenames user specifiable from the command line.

## B.3 k-fold Evaluator

The k-fold evaluator is located in the *k-fold* subdirectory within the TMiner distribution. The purpose of this utility is to construct a parsimonious model based on a dataset containing "first attempts" and on specified knowledge components. The model construction procedure eliminates learning knowledge components that have been fit with negative coefficients, because this would be saying that the student is unlearning. The procedure then eliminates coefficients that are statistically insignificant. The resulting model is parsimonious.

The k-fold evaluator makes use of a statistical software package called S-PLUS (see Appendix A). The k-fold evaluator communicates with the S-PLUS engine to construct the models. The Bayesian Information Criterion (BIC) is the metric obtained from the model to allow it to be compared with other models. Additionally the model construction makes use of a 10-fold cross validation, which allows a close approximation of the model's predictive performance to be determined.

The k-fold evaluator can be compiled by typing "make" within the */TMiner/k-fold* directory. To execute this program, type "run". The parameters (learning and difficulty knowledge components) included in the model are specified within the file CrossDriver.java. The model construction process is recorded in a log file that is specified within CrossDriver.java. The last model appearing within the log file is a parsimonious model resulting from the model construction procedure. A screenshot of a log file with an intermediate (non-parsimonious) model appears in the following figure.

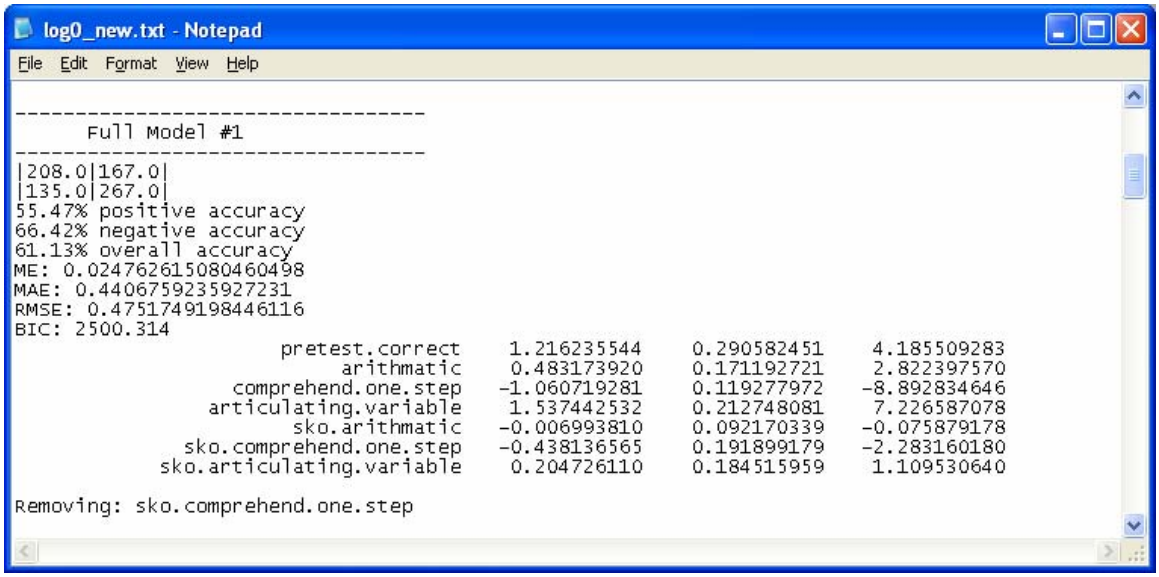


**Figure 4 Log file excerpt showing the model construction process**

## B.4 Ski-Slope Metric

The *skislope* subdirectory within the TMiner distribution contains a single source file: SkiSlope.java. This was an experimental implementation to investigate the usefulness of employing what Heffernan and his advisor Koedinger refer to as a Ski-Slope metric.

The Ski-Slope metric provides an additional column within the dataset, which is intended to replace the dependent "response" variable. The new is variable referred to as the Ski-Slope metric and is not restricted to being equal to either a 1 or 0 (correct or

incorrect). Instead, as the hint-level is increased (do to an incorrect response), a student's correct response would be give less than full credit (some real value between 0 and 1 as a function of the hint-level). A detailed discussion of the ski-slope metric employed will not be provided here, however additional information can be found at the website http://www.cs.wpi.edu/~ecroteau/LFA/

## B.5 Analysis

The *analysis* subdirectory within the TMiner distribution contains a single source file: OverUnderFit.java. This was an experimental procedure used to illustrate which problems the model had a tendency to "over" or "under" predict the actual problem difficulty (see Appendix E). By carefully looking at those problems being over or under predicted, it was possible to determine some modifications to be made to the model construction procedure. In particular, it was decided that the notion of a "first attempt" would be necessary from this analysis.

# Appendix C: Problem Difficulty

The difficulty of each problem given by the Ms. Lindquist tutor was investigated. The approximate difficulty was determined by the percentage of students answering the problem correctly on their first attempt. Intuitively problems having a lower percent correct are more difficult than those with a higher percent correct. The percentage of students answering each problem correctly on their first attempt is based on evidence provided by over 17,000 students. Due to randomization of the curriculum, not all students received the same set of problems. Also, students solved problems for a variable amount of time, which resulted in the number of students solving any particular problem being much less than 17,000.

The following table indicates each problem by its name and provides the percentage of students answering it correctly based on the number of students attempting that problem. The pretest and posttest problems were attempted by the greatest number of students, because these problems were used by each student taking the pretest and posttest. The problems are sorted in order of difficulty, those at the top being answered correctly with the highest frequency and those lower in the table becoming progressively more difficult.

| Problem Name | Percent Correct | Attempts |
|---|---|---|
| h_bar-b-que.lisp | 0.91573 | 178 |
| p_waiterTips.2op.lisp | 0.884393 | 519 |
| h_phonebill_dec2x-8.lisp | 0.854209 | 487 |
| n4_farmer_feet.3op.c_like.lisp | 0.839572 | 187 |
| c_one_op_d_eq_rt_d_3x.lisp | 0.838275 | 742 |
| h_mow_19x-35.lisp | 0.819838 | 494 |
| h_cats_12+x.lisp | 0.816883 | 770 |
| h_gum_x-3.lisp | 0.814448 | 706 |
| p_one_op_5x.lisp | 0.811153 | 789 |
| posttest_lemondae.like2op | 0.808588 | 2189 |
| h_bananas_20-x.lisp | 0.80597 | 737 |
| p_waiterTips.2op.like | 0.804435 | 496 |
| c_plumber_15+30h.2op.lisp | 0.803607 | 499 |
| n_MrHenson_.25w+1.5t.3op.lisp | 0.798742 | 159 |
| c_one_op_15-x.lisp | 0.792923 | 763 |
| p_CancerFundRaiser.3op.lisp | 0.791286 | 2846 |
| h_tree_3x.lisp | 0.786382 | 749 |
| h_scuba_25x+91.lisp | 0.782222 | 450 |
| h_pateoh_5h.lisp | 0.7787 | 723 |
| p_one_op_500-x.lisp | 0.771838 | 767 |
| c_one_op_3d.lisp | 0.771613 | 775 |
| h_annie_200-x.lisp | 0.766578 | 754 |
| h_babysitting_5h+6_h-2.lisp | 0.766082 | 171 |
| pat_subscrips_5x-100.2op.lisp | 0.763676 | 457 |
| h_annie2_200+x.lisp | 0.761491 | 805 |
| h_airplane_325h.lisp | 0.761214 | 758 |
| h_snowcones_1s-50.lisp | 0.759766 | 512 |

| Problem Name | Percent Correct | Attempts |
|---|---|---|
| h_cookies_100_x.lisp | 0.757781 | 739 |
| h_magazine_5x-100.lisp | 0.756757 | 481 |
| h_soccer_12+x.lisp | 0.75 | 732 |
| p_anne_AlgebraSymb.lisp | 0.749221 | 5455 |
| p_maryAge_is_albert-(Bob-c).lik | 0.740331 | 362 |
| h_doubloons_500_x.lisp | 0.739691 | 776 |
| c_paint_y+3y.2op.lisp | 0.737968 | 374 |
| h_farm_605-118x.lisp | 0.737374 | 99 |
| pat_30-2d.lisp | 0.73673 | 471 |
| h_shoes_x+3.lisp | 0.733681 | 766 |
| h_bamboo_3x+15.lisp | 0.733607 | 488 |
| h_fundraiser_x_4.lisp | 0.732725 | 767 |
| h_candy_35_x.lisp | 0.732304 | 777 |
| h_crayons_25_x.lisp | 0.73219 | 758 |
| h_test_25-p.lisp | 0.72053 | 755 |
| p_one_op_3d.lisp | 0.718894 | 10231 |
| c_shipping_bears_8+10x.2op.lisp | 0.716484 | 455 |
| h_telephone_5000x.lisp | 0.716235 | 733 |
| h_family_150_x.lisp | 0.714876 | 726 |
| p_one_op_d_eq_rt_s_100_d_x.lisp | 0.714286 | 770 |
| c_one_op_g+r.lisp | 0.71164 | 756 |
| c_one_op_d_eq_rt_s_y_d_z.lisp | 0.709287 | 743 |
| Posttest_cancer_like.3op | 0.700149 | 1344 |
| h_party_3x+6.lisp | 0.6893 | 486 |
| pat_phonecredit_10m-20.2op.lisp | 0.688596 | 456 |
| h_roses_4x.lisp | 0.686301 | 730 |
| h_shovel_8w-x.lisp | 0.685106 | 470 |
| n_lotterydivdiv.2op.lisp | 0.683616 | 354 |
| h_phonebill_2x-8.lisp | 0.682203 | 472 |
| h_freshman_400-40x.lisp | 0.68 | 475 |
| n_whale2op.lisp | 0.679272 | 714 |
| h_brownies_3x-50.lisp | 0.678112 | 466 |
| h_rainforests_9h.lisp | 0.674548 | 719 |
| h_tire_32-2x.lisp | 0.668776 | 474 |
| h_math_rocks_30+x.lisp | 0.665835 | 802 |
| p_one_op_d_eq_rt_s_x_d_6.lisp | 0.66033 | 789 |
| midtest_mcd-7.2op | 0.660118 | 6923 |
| midtest_tiger_dis_div_s.1op | 0.657439 | 3179 |
| p_one_op_d_eq_rt_s_x_d_h.lisp | 0.656697 | 769 |
| p_divide_gift_parens.lisp | 0.640751 | 373 |
| h_license_35+40t.lisp | 0.635802 | 486 |
| p_LemonadeStand.2op.lisp | 0.614379 | 459 |
| p_parachute5k-200s.lisp | 0.612705 | 488 |
| n_rt-d.2op.lisp | 0.605691 | 492 |
| p_one_op_x-10.lisp | 0.596257 | 748 |
| h_painting_x-5.lisp | 0.590296 | 742 |
| h_video_2x+v.lisp | 0.59009 | 444 |
| n_diff_tween_distnaces.3op.lisp | 0.587879 | 165 |
| h_car_t-3h.lisp | 0.582809 | 477 |

| Problem Name | Percent Correct | Attempts |
|---|---|---|
| p_its_demo_problem.lisp | 0.582353 | 170 |
| n_movinglawn_2_3_t.2op.lisp | 0.574713 | 174 |
| h_tickets_12+45t.lisp | 0.573499 | 483 |
| h_paint_20756-1250x.lisp | 0.568826 | 494 |
| c_one_op_booths_400_d_x.lisp | 0.56725 | 9435 |
| h_pool_i-1h.lisp | 0.566239 | 468 |
| h_company_150w-l.lisp | 0.552752 | 436 |
| p_one_op_d_eq_rt_t_12_d_s.lisp | 0.548473 | 753 |
| h_computer_x-1234.lisp | 0.538462 | 741 |
| h_liquid_-2x-35.lisp | 0.534335 | 466 |
| h_steelers_7+3c.lisp | 0.53252 | 492 |
| p_one_op_d_eq_rt_t_x_d_60.lisp | 0.528947 | 760 |
| post_combine_divide.2opdis | 0.515082 | 2188 |
| n_strawberries.3op.lisp | 0.497076 | 171 |
| pat_brick_laying.3op.lisp | 0.493827 | 162 |
| p_one_op_m_d_5.lisp | 0.492958 | 781 |
| h_cancer_145000y.lisp | 0.490642 | 748 |
| h_eggs_x_12.lisp | 0.48913 | 736 |
| n2_add_speeds.3op.lisp | 0.487342 | 158 |
| p_32FistOuncePlus20Cents.like | 0.48503 | 167 |
| h_film_7x-15.lisp | 0.484914 | 464 |
| post_total time.3op | 0.475872 | 1347 |
| n3_rate_of selling.lisp | 0.475763 | 557 |
| h_turnpike_55X+56.lisp | 0.473913 | 460 |
| h_company_450-150x.lisp | 0.473573 | 473 |
| h_population_233-177x.lisp | 0.467742 | 62 |
| p_one_op_d_eq_rt_t_x_d_r.lisp | 0.459603 | 755 |
| p_num_stud_in_class.2op | 0.458135 | 3774 |
| p_num_stud_in_class.2op.like | 0.444668 | 497 |
| midtest_dis_div_r.1op | 0.442006 | 3190 |
| midtest_ann_analogue.2op | 0.438943 | 6928 |
| p_two_jobs.lisp | 0.437804 | 2058 |
| c_num_stud_s+(s-4).2op | 0.437086 | 453 |
| h_skyscrapers_x-3.lisp | 0.436185 | 713 |
| h_teachers_879-23x.lisp | 0.429224 | 438 |
| c_num_swimer_4b+b.2op | 0.422998 | 487 |
| n_dropped_speed2op.lisp | 0.419098 | 377 |
| h_temperature_x-16.lisp | 0.414853 | 781 |
| post_debbie_like_dis.4op | 0.401932 | 1346 |
| h_bicycle_m-80t.lisp | 0.391304 | 483 |
| n_diff_speed.3op.lisp | 0.382857 | 175 |
| p_hourssaved(x-b)divSpd.2o.lisp | 0.377465 | 355 |
| n3_rate_of_invitations.lisp | 0.37464 | 347 |
| midtest_boys_girls.2op_dis | 0.373241 | 6966 |
| post_dfa5_savedtime.2opdis | 0.3701 | 2194 |
| h_NHS_6x-99.lisp | 0.339207 | 454 |
| p_32FistOuncePlus20Cents.3op | 0.289598 | 2538 |
| h_brownies2_x_5.lisp | 0.284138 | 725 |
| h_firstcar_x_10.lisp | 0.276243 | 724 |

| Problem Name | Percent Correct | Attempts |
|---|---|---|
| p_two_jobs.like | 0.272222 | 180 |
| p_maryAge_is_albert-(Bob-c).2op | 0.25876 | 371 |
| slope_candle_rate.3op.lisp | 0.256098 | 164 |
| p_biketrip_AlgebraSymb.lisp | 0.244444 | 360 |
| h_tin_10-2x.lisp | 0.240166 | 483 |
| n_plane_time_saved.2op.lisp | 0.232432 | 370 |
| n_how_much_faster2op.lisp | 0.215909 | 352 |
| n_hill_3u-10d.3op.lisp | 0.167665 | 167 |
| p_garage_two_rates_4op.lisp | 0.159763 | 169 |
| p_hourssaved(x-b)divSpd.like | 0.140957 | 376 |
| h_submarine_-2x-5.lisp | 0.133909 | 463 |

**Table 7 Approximate problem difficulty determined by percent correct**

# Appendix D: Most Common Incorrect Responses

In addition to investigating individual problem difficulty (see Appendix C), the most frequent errors made for each problem were determined. For each problem, the most common error (see Error #1) and the second most common error (see Error #2) were determined. The following table indicates the most common and second most common errors made for each problem. Additionally the correct answer is provided for comparison. This evidence is based on the responses provided by over 17,000 students. Not all students made errors and not all students received the same questions, so the number of students doing any particular problem is much less than 17,000.

| Problem Name | Error #1 | Error #2 | Correct Answer |
|---|---|---|---|
| c_num_stud_s+(s-4).2op | s-4 | 4-s | s+(s-4) |
| c_num_swimer_4b+b.2op | 4n | 4*n | n+4*n |
| c_one_op_15-x.lisp | 15/x | x-15 | 15-x |
| c_one_op_3d.lisp | p/2 | 2/p | 2*p |
| c_one_op_booths_400_d_x.lisp | x/400 | 400x | 400/x |
| c_one_op_d_eq_rt_d_3x.lisp | 3/x | x/3 | 3*x |
| c_one_op_d_eq_rt_s_y_d_z.lisp | y*z | z/y | y/z |
| c_one_op_g+r.lisp | r*g | rg | r+g |
| c_paint_y+3y.2op.lisp | y*3 | 3y | y+3*y |
| c_plumber_15+30h.2op.lisp | 30+15*h | 30*x+15 | 15+30*h |
| c_shipping_bears_8+10x.2op.lisp | 8*w | 10x+8 | 8+10w |
| h_airplane_325h.lisp | 325/h | h/325 | h*325 |
| h_annie2_200+x.lisp | 200*x | $200+x | 200+x |
| h_annie_200-x.lisp | x-200 | $200-x | 200-x |
| h_babysitting_5h+6_h-2.lisp | (h*5)+(h*6)-2 | (h*5)+(6*h-2) | 5*h+6*(h-2) |
| h_bamboo_3x+15.lisp | 15*x | 3x | 15+3*x |
| h_bananas_20-x.lisp | x-20 | 20/x | 20-x |
| h_bar-b-que.lisp | (y*2)+x*y | 4x+2y=z | 4*x+2*y |
| h_bicycle_m-80t.lisp | 80*t | 80t-m | m-80*t |
| h_brownies2_x_5.lisp | x/4 | 4/x | x/5 |
| h_brownies_3x-50.lisp | 50+3x | 50-3*x | 3*x-50 |
| h_cancer_145000y.lisp | 145,000/y | 145000/y | y*145000 |
| h_car_t-3h.lisp | 3*h-t | 3h-t | t-3*h |
| h_cats_12+x.lisp | 12*x | 12x | 12+x |
| h_company_150w-l.lisp | x-150*w | x-150w | 150*w-x |
| h_company_450-150x.lisp | 15000*x | 450000/x | 450000-15000*x |
| h_computer_x-1234.lisp | 1234-x | x-$1234 | x-1234 |
| h_cookies_100_x.lisp | x/100 | 100*x | 100/x |
| h_crayons_25_x.lisp | x/64 | 64*x | 64/x |
| h_doubloons_500_x.lisp | p/500 | 500/x | 500/p |
| h_eggs_x_12.lisp | 12x | 12*x | x/12 |
| h_family_150_x.lisp | x/150 | 150/x | 150/x |
| h_farm_605-118x.lisp | 6,051,000-x | 118250*x-6051000 | 6051000-118250*x |
| h_film_7x-15.lisp | 15-7x | 15-7*x | 7*x-15 |
| h_firstcar_x_10.lisp | 10*x | 10x | x/10 |
| h_freshman_400-40x.lisp | 40*x-400 | 40x-400 | 400-40*x |

| Problem Name | Error #1 | Error #2 | Correct Answer |
|---|---|---|---|
| h_fundraiser_x_4.lisp | x/3 | 4x | x/4 |
| h_gum_x-3.lisp | 3-x | x+3 | x-3 |
| h_license_35+40t.lisp | 40*t | 40t | 35+40*t |
| h_liquid_-2x-35.lisp | -35+2h | -35+2*h | -2*h-35 |
| h_magazine_5x-100.lisp | 100-5s | 5*x-100 | 5*s-100 |
| h_math_rocks_30+x.lisp | 30*x | $30+x | 30+x |
| h_mow_19x-35.lisp | 50-19*x | 50-19x | 19*x-35 |
| h_NHS_6x-99.lisp | 6*x | 6x | 6*x-99 |
| h_painting_x-5.lisp | 5-x | 5/x | x-5 |
| h_paint_20756-1250x.lisp | 20756/1250*x | 1250*x-20756 | 20756-1250*x |
| h_party_3x+6.lisp | 3+4+2*x | 3x+5 | 3*x+6 |
| h_pateoh_5h.lisp | 5+h | $5*h | 5*h |
| h_phonebill_2x-8.lisp | 8-2*m | 8-2m | 2*m-8 |
| h_phonebill_dec2x-8.lisp | 20*m-8 | 20m-8 | 0.2*m-8 |
| h_pool_i-1h.lisp | 2*h-i | 2*h | i-2*h |
| h_population_233-177x.lisp | 2236000-17740*10 | 2236000-17740 | 2236000-17740*x |
| h_rainforests_9h.lisp | 9/h | 9-h | 9*h |
| h_roses_4x.lisp | $4*r | 4*x | 4*r |
| h_scuba_25x+91.lisp | 25x-91 | 25+91*x | 25*x+91 |
| h_shoes_x+3.lisp | s*3 | 3s | s+3 |
| h_shovel_8w-x.lisp | x-8w | x-8*w | 8*w-x |
| h_skyscrapers_x-3.lisp | x+3 | 3*x | x-3 |
| h_snowcones_1s-50.lisp | 1x-50 | 50-1*s | 1*s-50 |
| h_soccer_12+x.lisp | 12*x | 12x | 12+x |
| h_steelers_7+3c.lisp | 2p+7 | 2*p+7 | 7+3*p |
| h_submarine_-2x-5.lisp | 2*h-5 | 2*h+5 | -5-2*h |
| h_teachers_879-23x.lisp | 23x | 23*x | 879-23*x |
| h_telephone_5000x.lisp | 5000/x | 5,000/x | 5000*x |
| h_temperature_x-16.lisp | x+16 | 16+x | x-16 |
| h_test_25-p.lisp | p-25 | 25/p | 25-p |
| h_tickets_12+45t.lisp | 45+12 | 45+12*t | 12+45*t |
| h_tin_10-2x.lisp | 10,000,000-2x | 10/2*x | 10-2*x |
| h_tire_32-2x.lisp | 2*x-32 | 32/x | 32-2*x |
| h_tree_3x.lisp | 3+x | x+3 | 3*x |
| h_turnpike_55X+56.lisp | 56-50x | 50x-56 | 56+50*x |
| h_video_2x+v.lisp | v+2.00*x | v+2.00x | 2*x+v |
| midtest_ann_analogue.2op | 500/60-m | 500/60 | 500-60*m |
| midtest_boys_girls.2op_dis | 62-x | 62+x | 62+(62-x) |
| midtest_dis_div_r.1op | 55*x | 55/x | x/55 |
| midtest_mcd-7.2op | 5x-7 | 5*x-7 | 5*h-7 |
| midtest_tiger_dis_div_s.1op | s/200 | 200s | 200/s |
| midtest_x-20.1op | ansx-20wer | 20*x | x-20 |
| n2_add_speeds.3op.lisp | (w*t)+(m*s) | wt+ms | w/t+m/s |
| n3_rate_of selling.lisp | p-50/6 | p/6 | (p-50)/6 |
| n3_rate_of_invitations.lisp | 130-x/3 | 130/3 | (130-x)/3 |
| n4_farmer_feet.3op.c_like.lisp | r+h | 4r+4h | 4*h+2*r |
| n_diff_speed.3op.lisp | (y/30)-(x/5) | y/30-x/5 | x/5-y/30 |
| n_diff_tween_distnaces.3op.lisp | (120/g)-(100/f) | 120/g-100/f | 120*g-100*f |
| n_dropped_speed2op.lisp | (k-5)/h | k*h-5 | k/h-5 |

| Problem Name | Error #1 | Error #2 | Correct Answer |
|---|---|---|---|
| n_hill_3u-10d.3op.lisp | 7x-4x | x/7-x/4 | x/4-x/7 |
| n_how_much_faster2op.lisp | k/t-90 | t-(k/90) | 90-k/t |
| n_lotterydivdiv.2op.lisp | x/4 | .5x/4 | (x/2)/4 |
| n_movinglawn_2_3_t.2op.lisp | t+(t*2)*3 | (t*2)*3 | (t*2+t)*3 |
| n_MrHenson_.25w+1.5t.3op.lisp | w/2+t/4 | (w/2)+(t/4) | 2*w+4*t |
| n_plane_time_saved.2op.lisp | 300/s | (300/s)-2 | 2-300/s |
| n_rt-d.2op.lisp | 100*h | h-300 | 100*h-300 |
| n_strawberries.3op.lisp | s+(s+7) | (s*2)+7 | s+(2s+7) |
| n_whale2op.lisp | 2x | m*m | 2*m |
| pat_30-2d.lisp | 30+2x | 2x-30 | 30-2*x |
| pat_brick_laying.3op.lisp | 45h | (45/h)+(60/h) | 45*h+60*(h-3) |
| pat_phonecredit_10m-20.2op.lisp | 20-3*k | 20-3k | 3*k-20 |
| pat_subscrips_5x-100.2op.lisp | 100-5m | 100-5*m | 5m-100 |
| Posttest_cancer_like.3op | 3s+2m | s+m | 3*m+2*s |
| posttest_lemondae.like2op | 15-(5*w) | 15-5w | 5*w-15 |
| post_combine_divide.2opdis | 972+p/5 | (972+p)/7 | (972+p)/5 |
| post_debbie_like_dis.4op | (3*40)+(h-3*55) | 120+55h | 40*3+55*(h-3) |
| post_dfa5_savedtime.2opdis | 550/h-2 | 550/h | 550/(h-2) |
| post_total time.3op | 30x+40y | (x*30)+(y*40) | x/30+y/40 |
| p_32FistOuncePlus20Cents.3op | 33+20x | 33+20*x | 33+20*(x-1) |
| p_32FistOuncePlus20Cents.like | 3+2h | 3+(2*h) | 3+2*(h-1) |
| p_anne_AlgebraSymb.lisp | 680 | 800-40*3 | 800-40*m |
| p_CancerFundRaiser.3op.lisp | s+t | 7t+10s | 7*s+10*t |
| p_divide_gift_parens.lisp | x-20/4 | x/4-20 | (x-20)/4 |
| p_garage_two_rates_4op.lisp | 2y+6x | (y*2)+(x*6) | 6x+2(12-y) |
| p_hourssaved(x-b)divSpd.2o.lisp | m/n-2 | (m/n)-2 | m/(n-2) |
| p_hourssaved(x-b)divSpd.like | 26/(208-x) | (208-x)/26 | 26/(208+x) |
| p_its_demo_problem.lisp | 600/11h | 600/(11*h) | 600-11*h |
| p_LemonadeStand.2op.lisp | 35-2g | 2*g | 2*g-35 |
| p_maryAge_is_albert-(Bob-c).2op | p-(i-3) | p-i-3 | p-(j-3) |
| p_maryAge_is_albert-(Bob-c).lik | (m+3)-s | m-3/s | (m-3)-s |
| p_num_stud_in_class.2op | g+12 | 12+g | g+(g+12) |
| p_num_stud_in_class.2op.like | x-7 | 7-x | x+(x-7) |
| p_one_op_3d.lisp | 3+x | x+3 | 3*x |
| p_one_op_500-x.lisp | x-500 | 500/x | 500-x |
| p_one_op_5x.lisp | 4/x | $4*x | 4*x |
| p_one_op_d_eq_rt_s_100_d_x.lisp | x/100 | x*100 | 100/x |
| p_one_op_d_eq_rt_s_x_d_6.lisp | 6/x | x*6 | x/6 |
| p_one_op_d_eq_rt_s_x_d_h.lisp | x*h | h/x | x/h |
| p_one_op_d_eq_rt_t_12_d_s.lisp | 12s | 12*s | 12/s |
| p_one_op_d_eq_rt_t_x_d_60.lisp | 60/x | 60*x | x/60 |
| p_one_op_d_eq_rt_t_x_d_r.lisp | x*r | r/x | x/r |
| p_one_op_m_d_5.lisp | 5/m | m-5 | m/5 |
| p_one_op_x-10.lisp | 10-x | 10/x | x-10 |
| p_parachute5k-200s.lisp | 214*s-5000 | 5000-214x | 5000-214*s |
| p_two_jobs.like distance | 55i+65(40-i) | 55i | 55*j+65*(40-j) |
| p_two_jobs.lisp | 5g+30+g/7 | 5g+7g | 5*g+7*(30-g) |
| p_waiterTips.2op.like | 55-2f | f*2 | 55+2*f |
| p_waiterTips.2op.lisp | h+40 | h*5=40 | 40+5*h |

| Problem Name | Error #1 | Error #2 | Correct Answer |
|---|---|---|---|
| slope_candle_rate.3op.lisp | (y-4)/(8-x) | 8-x/y-4 | (8-x)/(y-4) |

**Table 8 Most common incorrect responses for each problem**

# Appendix E: Model Mispredictions

It is important to have the ablility to analyze a model's performance in detail. Using the BIC and predictive accuracy of a model are useful metrics, but they do not indicate the performance on a problem-by-problem level. To account for this level of granularity, the concept of a misprediction for a particular problem was defined as being the average difference in the model's prediction and the actual response for each attempt at the problem. The misprediction for a particular problem is given by the formula: $\frac{\sum_{i=1}^{n}(P_i - A_i)}{n}$, where $P_i$ is the model's prediction for the $i^{th}$ attempt, $A_i$ is the actual response for the $i^{th}$ attempt and $n$ indicates the number of attempts at the problem. Each student attempting a problem contributes to the misprediction associated with that problem. Using this definition, a number (misprediction) on the interval [-1, 1] can be produced for each problem by analyzing a collection of student responses. If the misprediction is close to zero, it can be said that the model is extremely well fit for that particular problem. A misprediction in the positive direction indicates the model is predicting too high for the responses associated with that problem, which means that the model is predicting a student's success too frequently for that problem. A misprediction in the negative direction indicates that the model has a tendency to under predict the responses associated with that problem.

The usefulness of mispredictions is seen when looking for trends or patterns giving indication of why a model is poorly predicting certain problems. An example of this could be when a problem is written ambiguously or possibly has an incorrect answer. So a problem may be more than just difficult (see Appendix C), it could also be poorly fit due to human or modeling error. The following table indicates the misprediction using one of the generated models for each problem. This evidence is based on the responses provided by the students in Mr. X's class.

| Problem Name | Model Misprediction |
|---|---|
| h_bamboo_3x+15.lisp | 0.32526496 |
| h_phonebill_dec2x-8.lisp | 0.3246356 |
| c_plumber_15+30h.2op.lisp | 0.30296338 |
| p_waiterTips.2op.lisp | 0.27914256 |
| h_magazine_5x-100.lisp | 0.2791076 |
| h_cats_12+x.lisp | 0.25960073 |
| h_test_25-p.lisp | 0.2596001 |
| h_gum_x-3.lisp | 0.25953308 |
| pat_subscrips_5x-100.2op.lisp | 0.25566489 |
| p_waiterTips.2op.like | 0.2548535 |
| h_snowcones_1s-50.lisp | 0.23318931 |
| h_video_2x+v.lisp | 0.23233204 |
| h_license_35+40t.lisp | 0.23214418 |
| h_car_t-3h.lisp | 0.21069542 |
| midtest_mcd-7.2op | 0.20536073 |
| c_one_op_g+r.lisp | 0.20074794 |

| Problem Name | Model Misprediction |
|---|---|
| h_soccer_12+x.lisp | 0.18264595 |
| c_one_op_15-x.lisp | 0.17620605 |
| h_painting_x-5.lisp | 0.16866349 |
| c_shipping_bears_8+10x.2op.lisp | 0.16210718 |
| h_mow_19x-35.lisp | 0.16105089 |
| p_one_op_5x.lisp | 0.14843409 |
| p_one_op_3d.lisp | 0.1463316 |
| h_teachers_879-23x.lisp | 0.14365368 |
| h_party_3x+6.lisp | 0.14183629 |
| h_cookies_100_x.lisp | 0.13456534 |
| h_family_150_x.lisp | 0.13455689 |
| h_bananas_20-x.lisp | 0.13454747 |
| h_candy_35_x.lisp | 0.13450995 |
| h_farm_605-118x.lisp | 0.1296043 |
| h_math_rocks_30+x.lisp | 0.1262812 |
| h_phonebill_2x-8.lisp | 0.124909185 |
| h_pool_i-1h.lisp | 0.12022697 |
| h_tree_3x.lisp | 0.10571594 |
| p_one_op_500-x.lisp | 0.092924036 |
| h_scuba_25x+91.lisp | 0.084204085 |
| h_doubloons_500_x.lisp | 0.08306465 |
| h_crayons_25_x.lisp | 0.07773938 |
| n_whale2op.lisp | 0.07771478 |
| c_one_op_booths_400_d_x.lisp | 0.07586609 |
| h_tickets_12+45t.lisp | 0.061534338 |
| h_tin_10-2x.lisp | 0.05642311 |
| c_one_op_3d.lisp | 0.045276534 |
| h_film_7x-15.lisp | 0.04288682 |
| h_shoes_x+3.lisp | 0.037335522 |
| n_rt-d.2op.lisp | 0.023348259 |
| h_freshman_400-40x.lisp | 0.022026509 |
| h_shovel_8w-x.lisp | 0.018680813 |
| midtest_ann_analogue.2op | 0.009863825 |
| pat_phonecredit_10m-20.2op.lisp | -0.001077952 |
| p_anne_AlgebraSymb.lisp | -0.009606479 |
| h_roses_4x.lisp | -0.013116392 |
| h_rainforests_9h.lisp | -0.013216132 |
| h_steelers_7+3c.lisp | -0.019116707 |
| h_cancer_145000y.lisp | -0.026140217 |
| h_telephone_5000x.lisp | -0.034575358 |
| h_tire_32-2x.lisp | -0.034818117 |
| p_LemonadeStand.2op.lisp | -0.038796194 |
| h_company_450-150x.lisp | -0.044377565 |
| h_pateoh_5h.lisp | -0.04815449 |
| h_bicycle_m-80t.lisp | -0.057469375 |
| h_annie2_200+x.lisp | -0.058604434 |
| c_num_stud_s+(s-4).2op | -0.060327716 |
| h_paint_20756-1250x.lisp | -0.0625076 |
| h_company_150w-l.lisp | -0.06494704 |

| Problem Name | Model Misprediction |
|---|---|
| midtest_boys_girls.2op_dis | -0.06657256 |
| h_turnpike_55X+56.lisp | -0.06752801 |
| h_annie_200-x.lisp | -0.07375089 |
| p_one_op_m_d_5.lisp | -0.07378295 |
| pat_30-2d.lisp | -0.08029953 |
| p_parachute5k-200s.lisp | -0.08500492 |
| h_computer_x-1234.lisp | -0.09333609 |
| h_liquid_-2x-35.lisp | -0.09899583 |
| h_submarine_-2x-5.lisp | -0.10298692 |
| c_num_swimer_4b+b.2op | -0.112069875 |
| p_one_op_x-10.lisp | -0.1154356 |
| h_temperature_x-16.lisp | -0.12509955 |
| h_population_233-177x.lisp | -0.12860289 |
| h_brownies2_x_5.lisp | -0.14041558 |
| h_skyscrapers_x-3.lisp | -0.1404366 |
| h_fundraiser_x_4.lisp | -0.14951557 |
| p_num_stud_in_class.2op | -0.16162327 |
| h_NHS_6x-99.lisp | -0.20493731 |
| h_eggs_x_12.lisp | -0.21108305 |
| h_brownies_3x-50.lisp | -0.2214626 |
| h_firstcar_x_10.lisp | -0.24043237 |
| p_num_stud_in_class.2op.like | -0.24707532 |

**Table 9 Example of model mispredictions**