Reaching More Students: A Web-based Intelligent Tutoring System with support for Offline Access

by

Paul Kehrer

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

May 26, 2012

APPROVED:

Professor Neil Heffernan, Major Advisor

Professor Joseph Beck, Thesis Reader

Professor Craig Wills, Head of Department

Abstract

ASSISTments is a web-based intelligent tutoring system that can provide students with immediate feedback when they are doing math homework. Until now, ASSISTments required internet access in order to do nightly homework. Without ASSISTments, students do their work on paper and are not told if they are correct or given help for wrong answers until the next morning at best. We've developed a component that supports "offline-mode", enabling students without internet access at home to still receive immediate feedback on their responses. Students with laptops download their assignments at school, and then run ASSISTments at home in offline mode, utilizing the browser's application cache and Web Storage API. To evaluate the benefit of having the offline feature, we ran a randomized controlled study that tests the effect of immediate feedback on student learning. Intuition would suggest that providing a student with tutoring and feedback immediately after they submit an answer would lead to better understanding of the material than having them wait until the next day. The results of the study confirmed our hypothesis, and validated the need for "offline mode."

Acknowledgements

For the development of the “offline tutor”, I would like to thank the ASSISTments GWT development team of David Magid (Project manager, WPI), Thomas Hays (Junior at Clarkson University), Zach Broderick (MS, WPI), Skyler Whorton (MS, WPI), and Corey Belhumeur (MS, WPI).

I would also like to thank Kim Kelly (PhD Candidate, WPI), whose students were the participants in the Feedback Immediacy study. She was critical in the design and running of the study.

For his research on the “Response Tabling” paper, I would like to thank Qing Yang Wang (MS, WPI).

I would also like to thank Neil Heffernan and Cristina Heffernan for creating and managing the ASSISTments project.

Funding for this research was provided primarily by the National Science Foundation’s GK-12 program, of which I am a fellow as part of the Partnership In Math and Science Education (PIMSE) grant. Additional sources of funding include the Department of Education, the Office of Naval Research, and the Spencer Foundation.

Contents

Tables

Figures

1. Introduction

With the use of technology in grade school education becoming more and more mainstream, educators are faced with the economic challenge of providing high-tech educational solutions that are available to all students. ASSISTments is a web-based intelligent tutoring system that is free to use by anyone, provided that they have internet access. As part of the Maine Learning Technology Initiative (MLTI), Maine's department of education provided all 7th and 8th grade students and teachers with laptop computers for use at home and school. We have developed a new component of the ASSISTments platform, called "offline mode", that allows all students in the Maine laptop program to benefit from ASSISTments, even without home internet access.

ASSISTments(www.assistments.org) was developed in 2003 by Neil Heffernan. It is a free service that has been developed and maintained since then by graduate students of Prof. Heffernan. The service has stayed free through grants by various organizations including the National Science Foundation and the Department of Education. The site is used by about 2,500 students each day, primarily middle school math students in Massachusetts. Our number of active users has approximately doubled every year for the last four years. "ASSISTments" as a word is a combination of "assistance" and "assessment", blending the benefits to students with the data collection for teachers [14]. This research focuses on the "assistance" aspect of the system, and how the ASSISTments platform can improve student learning. This assistance comes in three main forms. Immediate feedback tells the student whether or not they were correct after they enter an answer, hints can be requested giving the student extra information about solving the problem, or scaffolding will break the main problem into sub-steps giving the student a step-by-step approach to the problem. The evaluation portion of my research focuses on the benefits of immediate feedback.

Maine's one-to-one laptop program hopes to bring the benefits of educational technology to its middle school students and teachers. We believe that ASSISTments will make a perfect complement to this program. In 2009, Mendicino and Razzaq reported significant learning improvement in ASSISTments web-based homework when compared to traditional "paper and pencil" homework [12]. With evidence that computer based educational tools are effective, we set out to implement "offline mode," so that all students in Maine would have access to ASSISTments for nightly homework even without home internet access. The offline mode provides students with all of the features of regular ASSISTments, provided they can download their assignments at school. So we are making the claim that the immediate feedback provided by "offline-mode" leads to better learning, so therefore with ASSISTments and offline mode, we can benefit all students in the Maine one-to-one laptop program, regardless of internet access.

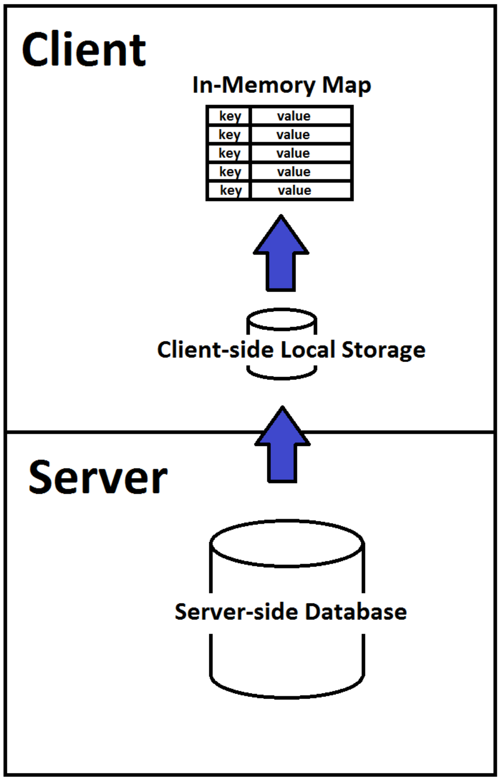
2. Offline Tutor Design

In order for us to study its effectiveness, we needed to actually implement the new ASSISTments client with support for “offline mode.” Due to various issues with the existing client, and the general desire for a more extensible piece of software, we decided to develop an entirely new client-side application. We believe the investment of time in redesigning the client was worth it because of the future benefits of being able to make additions to a more stable and well-architected system[16].

For the new client-side component, or the “Offline Tutor” as it became referred to in development, we decided to use the Google Web Toolkit (GWT). GWT is a toolkit that allows for development using Java source code that compiles into optimized JavaScript. It allows us to do the vast majority of our development in Java, which is helpful because we find that generally new developers joining the ASSISTments team are more comfortable using Java than JavaScript. GWT also provides automatic cross-browser support, so we don’t need to worry about the frustrating differences between web browsers. Lastly, it provides easy-to-use abstractions for some key HTML5 features which make “offline mode” possible. The two essential features being the Application Cache, which makes our JavaScript client available offline, and persistent Local Storage, which allows us to store information about a student’s progress in the web browser until it can be uploaded to the ASSISTments servers.

2.1 The Persistence Manager

At the core of our new client's architecture is the abstraction we refer to as the Persistence Manager. It is a component that handles all transactions involving persistent entities. These include content objects like Problems and Problem Sets as well as student progress, which is stored as a list of Student Actions. We wanted our new tutor to be able to seamlessly transition between online and offline modes, so our persistence manager is a 3-tier component that stores data on both the client and the server when possible. The first tier is an in-memory key-value mapping of our persisted objects. The second tier is the client-side Local Storage, which persists key-value strings even after the web browser is closed. The final tier is the server side database, where the teacher can view data.



key	value
key	value
key	value
key	value
key	value

FIGURE 1: DESIGN OF THE "PERSISTENCE MANAGER"

When objects need to be retrieved, the system looks in the first tier. If it doesn't exist there, it looks in the second tier. If it doesn't exist in either of the client-side tiers, it makes a request of the server. When storing objects, the client attempts to write to all three tiers. If the server side tier is unavailable because the application doesn't have internet access, then the persisted data is stored in a pending queue, which is sent as soon internet access is restored.

With this persistence model, as long as the client receives all of the content it needs for an assignment before it loses internet access, it can be a fully functioning tutor with no access to the web. That is to say, that the student experience of doing problem sets in "offline mode" will be identical to the online experience. They will receive immediate feedback and tutoring help. The student's teacher will also be able to see all of the work the student completed in offline mode once the student "hands in" their work at school.

2.2 Usage model for "Offline Mode"

The way students will use "offline mode" has been designed to be as simple as possible. We wanted "offline mode" to be as similar as possible to the classic ASSISTments usage model. In classic ASSISTments, the students log into their class, and click on an assignment. They then enter the answers for the problem set in the browser. When they are finished, they navigate back to the assignments page, and can close their browser. Offline mode requires two more simple steps in order to do homework without internet access.

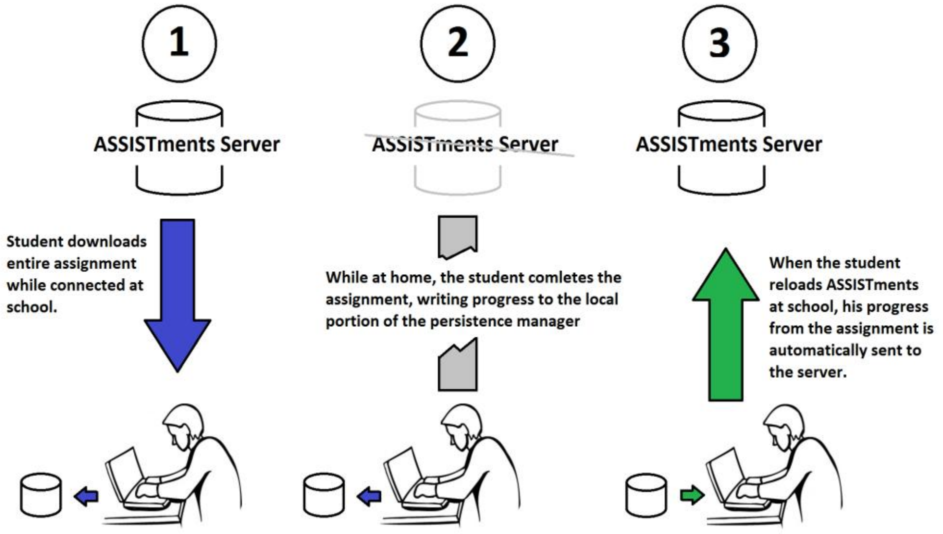
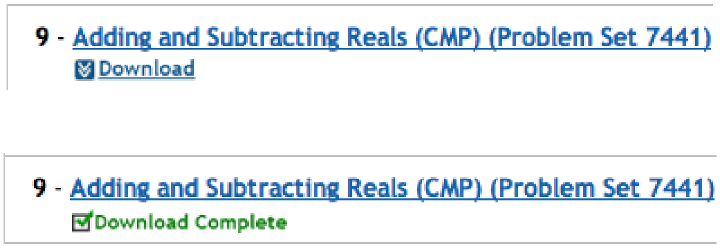

1
2
3
ASSISTments Server
~~ASSISTments Server~~
ASSISTments Server
Student downloads entire assignment while connected at school.
While at home, the student comletes the assignment, writing progress to the local portion of the persistence manager
When the student reloads ASSISTments at school, his progress from the assignment is automatically sent to the server.

FIGURE 2: STANDARD USE CASE FOR ASSISTMENTS "OFFLINE MODE"

First, the student must download the required assignments. They do this by simply clicking the "Download" button, which now appears under each assignment link in the student's assignments page. After the downloading dialog completes, the students will see the "Download Complete" notification underneath the downloaded assignment. We expect that the student will download a few different assignments that are due the following day, so that he can work on them that night.



9 - Adding and Subtracting Reals (CMP) (Problem Set 7441)

9 - Adding and Subtracting Reals (CMP) (Problem Set 7441)

FIGURE 3: UI FOR DOWNLOADING ASSINGMENTS FOR USE IN "OFFLINE MODE" (BEFORE AND AFTER DOWNLOADING)

After downloading his assignments, the student takes his laptop home to do his homework. During the download process, our application will have already stored itself in the browser's application cache. This way, when the student goes to www.assistments.org/offline, which is a URL we instruct them to use, their browser will load our application out of the application cache. The student will then pick from the downloaded assignments. When doing the assignments, the user experience will be identical to the online tutor. We have downloaded all of the questions, the answers, and also the tutoring help if it exists. Once the student finishes the work, he can close his browser.

The final step in the offline usage model is "handing in" the offline work. The student does this by logging onto their ASSISTments account at school where they have internet access. On their landing page, if they did offline work, there will be a prominent link that reads, "You have offline work to hand in." Clicking on this sends all of the work they did since their last hand in to the server. At this point, the teacher can view everything they did the night before with the same level of detail as if they had done the work online.

2.3 Additional Design Decisions

There were other considerations made during the design of the tutor that we believed would allow us to make a more extensible and robust application.

2.3.1 Separation of problem set structure and content

When downloading problem sets, the application must be aware of the entire structure of the problem set. If it is a "Random Child Order" problem set, the application needs to know about every problem that it has to choose from. However, we did not want our application to have to download every single problem body, every piece of tutoring, and every correct and incorrect answer just to be able to start the problem set. For this reason, we separated the definition of

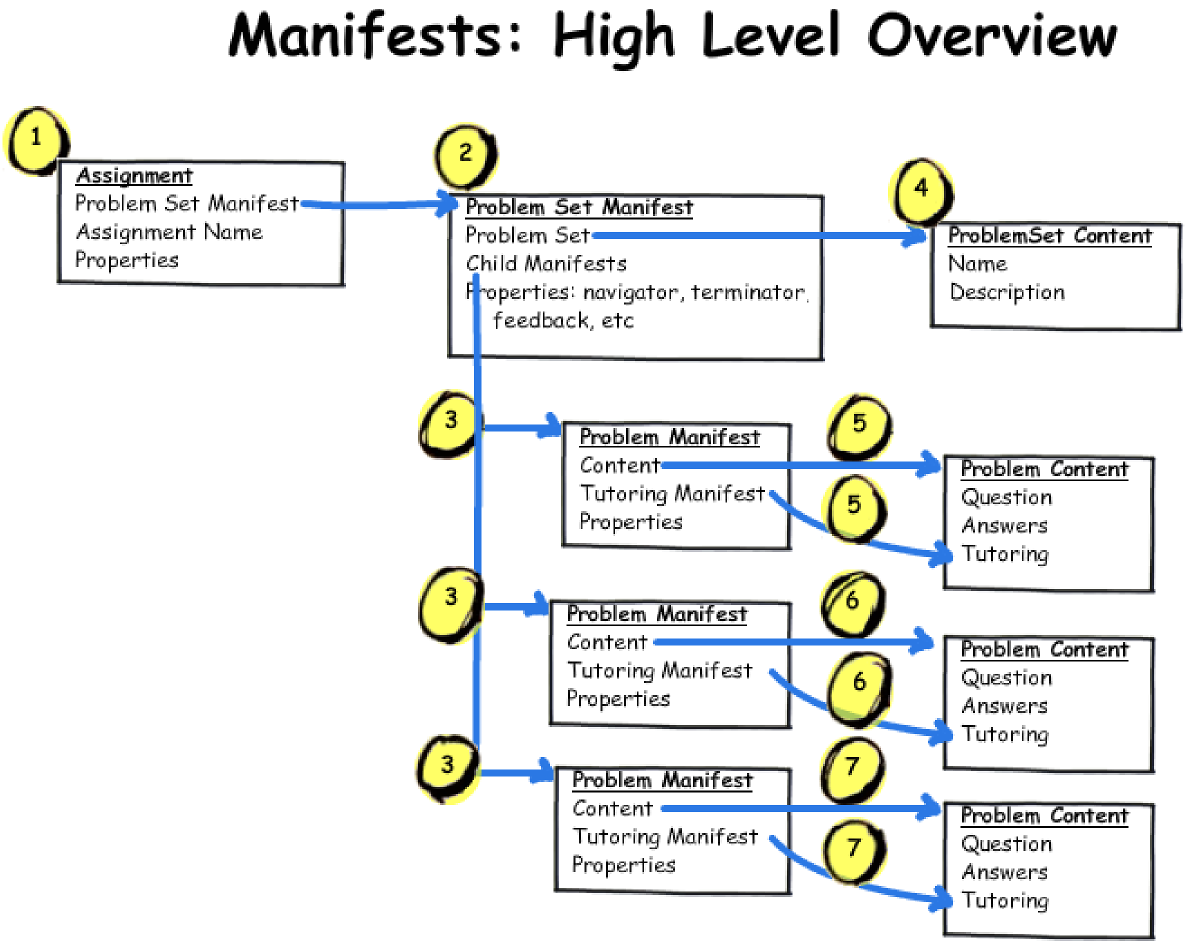


FIGURE 4: MANIFEST STRUCTURE

problem set structure and problem content. We define structure to be which problems are part of a problem set, as well as what order those problems will be presented in. Content, on the other hand, is the actual text and html of the problem body, and any tutoring that may be included. This division of data is outlined in Figure 4. Each assignment consists of a tree of “manifests”, which describe the structure. Every node of this tree also has a pointer to “content” objects, which include the problems and the tutoring.

With this manifest and content structure, our application can start by just downloading the manifest tree. This tells the application everything it needs to know about traversing the problem set. Then the application can request content as it needs it. If the user is downloading the assignment to run offline, then of course the application must download the manifest tree as well as every piece of content on the nodes of that tree. Because of the abstraction of the persistence manager, whether the content is downloaded initially or only when needed does not affect the general behavior of the tutor.

2.3.2 Performance considerations: Look-ahead & Asynchronous Logging

The new tutor can run with offline-mode either on or off. If offline mode is not being used, the client doesn’t need to download every single bit of content before starting. I did, however, want the tutor to run smoothly, and not make the user wait every time the tutor needs to request a new problem from the server. In order to accomplish this, we implemented a look-ahead policy. This takes advantage of our layered persistence manager. As soon as you start a problem, the look-ahead policy tells the persistence manager to request the next problem from the server. This way, when the user completes the problem they are currently working on and the tutor asks for the next one, it is already downloaded to the client. Because the client-server distinction is abstracted by the persistence manager, the component that requests the problem doesn’t need to know where it is. The request is automatically fulfilled by data that has been saved on the client. The look-ahead policy makes it so that as soon as the first problem is loaded, the client will run without any delays from making requests of the server.

The other action that could potentially have a negative impact on performance is logging student progress at the server. If we had to wait for the server to respond after sending our progress at every problem, there would be an irritating pause after submitting each answer. To avoid this, we log progress asynchronously. We maintain a queue of student actions, and based on a time interval, and also specific student actions, we send these to the server in a background thread of our application. This way, the server is always up to date to within 10 seconds of the student’s progress, and the client application never has to wait on the server’s response.

3. Evaluation – Feedback Immediacy Study

The task of evaluating the need and effect of the offline tutor is a difficult one. The offline functionality is intended to only affect a small fraction of the students that use the redesigned tutor. For this reason, it seemed unfeasible to run a randomized controlled study on students who do not have access to the internet at home. We simply didn't have access to a large enough group of students without internet access.

We decided instead to evaluate the benefits of having offline tutor access. Before the offline tutor, teachers would use ASSISTments with their classes, and if students didn't have internet access, they would have to create some alternative for those students. Usually this consists of giving the students a worksheet to fill out at home. When the student returns to school, he can either hand in the worksheet to be corrected by the teacher, or he can use a school computer to enter his answers in ASSISTments. We believe that this student is at a disadvantage without the offline tutor. If he enters his answers at school he is still receiving feedback on his answers, but the feedback is delayed a day. We think that students with access to the internet, and who do their work in ASSISTments will benefit from the immediate feedback, and therefore learn more from their homework.

3.1 Overview of the Feedback Immediacy Study

The motivation for the Feedback Immediacy study came from a question that a parent asked of Kim Kelly, a middle school teacher in Grafton, MA. The parent, who preferred the more traditional "pen and paper" homework style, asked, "Is it okay if my student does his work on a piece of paper, and then just enters the work in ASSISTments after?" Mrs. Kelly's intuitive reaction to the question was "No!" because that would deprive the student of ASSISTments' immediate feedback. We wanted to figure out if this intuition is correct. Does getting feedback as you go lead to better learning than receiving the same feedback after you have already worked through the problems on your own?

This study will serve as method of evaluating the need for the offline tutor as well. With "offline mode", a student with no internet access can still receive immediate feedback. The tutor was specifically designed to provide the offline user with the same user experience when doing homework. Since the offline tutor provides this immediate feedback, the question of "Is immediate feedback important?" will also lead us to know if there is a true need for the offline tutor.

3.2 Study Design

3.2.1 Participants

The study was run with 8th grade middle school students from Grafton Middle School in Grafton, MA. Their teacher was Kim Kelly. The students were from 3 separate class periods, but they were all part of the same Pre Algebra course. The work was done in class using the ASSISTments system, or a worksheet and pencil.

3.2.2 Subject Matter

There were two separate homework assignments given to the students. The first covered Surface Area of 3 dimensional figures, and the second assignment covered Volume of 3 dimensional figures. Mrs. Kelly believes that the difficulty of the two assignments were approximately the same, however our data analysis accounts for the possible effect of the different assignments.

3.2.3 Procedure

The overall goal of the study is to evaluate student knowledge of a subject before their homework, then have the students do their homework either with immediate feedback, or with delayed feedback, and then we will evaluate their knowledge of the subject again after homework. To do this we will provide the students with three different problem sets for each assignment:

Surface Area:

- Surface Area Pre-Test (4 questions)
- Surface Area Homework (15 questions)
- Surface Area Post-Test (4 questions)

Volume:

- Volume Pre-Test (5 questions)
- Volume Homework (14 questions)
- Volume Post-Test (5 questions)

We will then break the students into two separate conditions, and then switch the conditions as part of a crossover design. Group A will be the ASSISTments condition, evaluating the standard ASSISTments use case, and Group B will be the Worksheet condition. The point of the worksheet condition is to simulate the scenario where a student doesn't have internet access at home, but is assigned ASSISTments work by their teacher. The best they can do is to do the work on a printed worksheet, and then enter their answers in ASSISTments the next day.

In Figure 5 you can see the basic breakup of conditions across assignments.

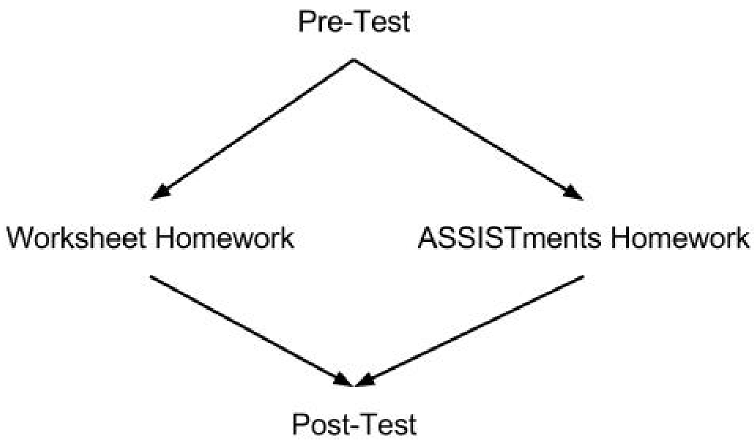


FIGURE 5: CONDITIONS BREAKUP ACROSS ASSIGNMENTS

At the start of the assignment, all students will take a pre-test on the subject. That night, the ASSISTments condition will be presented with an ASSISTments homework assignment in the regular "Tutor mode", in which they will be presented with helpful feedback whenever they answer incorrectly. The Worksheet condition will be given the same homework assignment, but in the form of a paper worksheet. They will not have access to ASSISTments. The following day in class, the students who were not given feedback on the homework will be presented with a paper version of the feedback that the other group received. On the following day in class, students in the Worksheet condition are given an opportunity to enter their answers in ASSISTments. At this time, they will see the same assignment that the ASSISTments condition saw on the previous night. In this way, the students who didn't receive immediate feedback will still receive "delayed" feedback. We will then give the students a post-test on the next day of class, evaluating what they had learned doing the homework. We hypothesized that students who receive their feedback immediately will improve more on the post test than students who received delayed feedback.

Below is the schedule for the study:

TABLE 1: STUDY SCHEDULE

Time:	**ASSISTments Condition**	**Worksheet Condition**
Monday in class	All students **Pre-test** on “Surface Area”	
Monday night homework	Students do **the “Surface Area” assignment on ASSISTments.**	Students do the same “Surface Area” assignment as a **paper and pencil worksheet.**
Tuesday at the beginning of class	Students do unrelated activity.	Students are given a short period of time in which they can **enter their answers into ASSISTments**, getting the same correctness feedback and answer hints that the ASSISTments group got the night before.
Tuesday during class	All students **Post-test** on “Surface Area”	

We then reversed the conditions as part of the crossover design, making Group A the Worksheet condition, and Group B the ASSISTments condition, and ran the same study schedule with the Volume assignment. This way we had data points for both conditions for every student.

3.3 Study Results

The focus of our data analysis was on the student’s pre-test and post-test scores. Below is a sample of our data results (full results can be found in the appendix). The table shows all of the data points for 3 students. The actual results included data for all 64 students involved in the study.

TABLE 2:SAMPLE RESULTS DATA

Student Name	Condition	Assignment	Pre-test	Post-test
student_1	ASSISTments	Surface Area	0%	100%
student_2	ASSISTments	Surface Area	25%	75%
student_3	ASSISTments	Surface Area	0%	75%
…				
student_35	Worksheet	Surface Area	25%	75%
student_36	Worksheet	Surface Area	25%	0%
student_37	Worksheet	Surface Area	0%	50%
…				
student_1	Worksheet	Volume	0%	40%
student_2	Worksheet	Volume	20%	80%
student_3	Worksheet	Volume	0%	60%
…				
student_35	ASSISTments	Volume	0%	80%
student_36	ASSISTments	Volume	0%	40%
student_37	ASSISTments	Volume	0%	60%
…				

With the study design that crosses over the conditions across two assignments, we were unable to simply evaluate the gain scores from the pre-test to the post-test. In order to account for the possible effect of the assignment as well as the student's pretest score, we decided run an ANCOVA using SPSS to test which of the variables had an effect on student performance.

The ANCOVA fits a linear model to predict a dependent variable of student post-test scores based on the independent variables of individual student effect, the student's pretest score, the condition, and the assignment. The result of this analysis provides us with parameter estimates and their significance. The table below shows the parameter estimates for Condition, Assignment, and Pre-test score. The effect of the individual students vary and while they are important to the integrity of the model, their values are not important to our analysis.

TABLE 3: PARAMETER ESTIMATES FROM ANCOVA

Parameter	Estimate Value	P-value
Condition = ASSISTments	0.098	0.021
Condition = Worksheet	0	
Assignment = Surface Area	0.123	0.026
Assignment = Volume	0	
Pre-test Score	0.143	0.639

The estimate for the condition parameter supports our hypothesis. It suggests a 9.8% improvement in post-test scores for students in the ASSISTments condition. The assignment parameter also has a significant effect. It says that the Surface Area assignment will lead to 12.3% percent post test score increase, suggesting that the assignments are not equal in difficulty. The ANCOVA reports that pre-test scores do not have a reliable effect on post-test scores. This is likely either due to the fact that our pretest scores are very low, and have little variance, or that the effect of pretest is accounted for in the individual student effect. The average pretest score across the entire data set is 9.18%, with a standard deviation of 12.56%. With such low scores, there is no observable effect from the pre-test scores in our data set.

3.4 Study Conclusions

The effect of the condition parameter supports our hypothesis. We suspected that having immediate feedback when doing homework would lead to better learning than only receiving delayed feedback. The results from our dataset suggest that there is a 9.8% post-test score increase from having immediate feedback as opposed to doing all of the work on a worksheet and then getting delayed feedback. Our intuition for this result is that immediate feedback helps to correct misconceptions in student learning as soon as they are made. In the work sheet condition, it is possible for a student to reinforce a misconception of the content by making the same mistake over and over without being corrected by ASSISTments' immediate feedback.

Since immediate feedback leads to increased learning gains, we can say that there is a genuine need for the offline tutor where it can be used. The work-around of giving students a worksheet in place of ASSISTments is an inferior alternative. If these students were given access to the ASSISTments "Offline Mode", they would have full access to ASSISTments tutoring and immediate feedback, giving them the same advantage seen in the ASSISTments condition of our study.

4. Response Tabling Experiment

Along with the direct benefits to student learning that have already been discussed, ASSISTments also provides a platform for educational research. Whereas the experiment in chapter 3 was a study on two different teaching techniques, the following is a study of a new student behavior model that can be used to predict student responses. I assisted in the research of Qing Yang Wang, a MS student in Computer Science at WPI, running the residual analysis of our results, and writing up our findings. We had the following paper accepted into the KDD 2011 Workshop on Educational Data.

4.1 Introduction

Intelligent Tutoring Systems (ITS) aim to improve student learning using reliable assessment, while providing students and teachers immediate feedback on student performance. The fine-grained data produced by these tutoring systems provide an opportunity to model student behavior using various methods. One of the most proven and accepted methods in the ITS field is Knowledge Tracing [6] which uses a Dynamic Bayesian Network to track student knowledge. Knowledge tracing provides both the ability to predict future student response values, as well as providing an addition parameter: the probability of student knowledge. For this reason, KT provides insight that makes it useful beyond the scope of simple response prediction. However, KT can be computationally expensive. Model fitting procedures, which are used to train KT, can take hours or days to run on large datasets [15][2][11]. Is this extra computation necessary or can performance be as effectively predicted by calculating simple percent correct features of past response data? This approach was shown to be effective when based on past hint count and response time information [18]. We propose an alternative to KT that matches KT's predictive accuracy with minimal computational cost. We suggest that if the task at hand is strictly prediction such as predicting end of year student outcomes or within tutor responses (KDD Cup), our simple tabling model, which ensembles percent correct response predictions, offers a fast and effective solution.

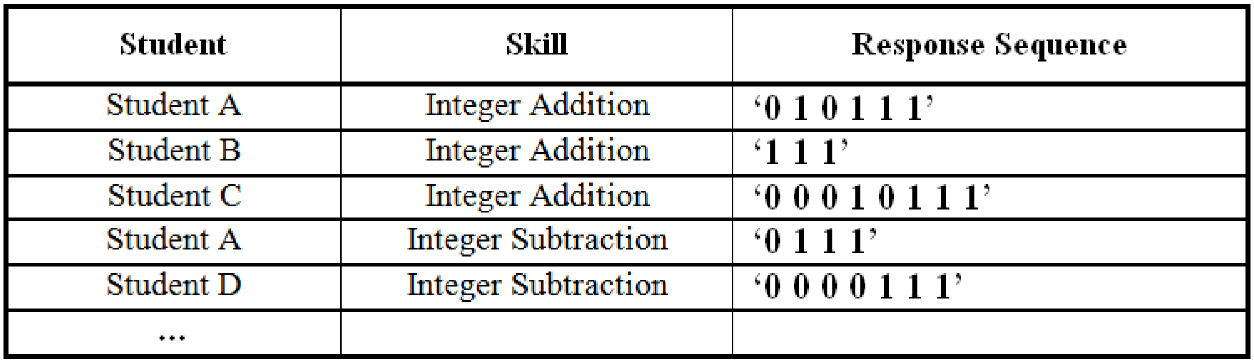
4.2 Dataset Description

The dataset used in this paper came from the ASSISTments Platform, a web-based tutoring system developed at Worcester Polytechnic Institute and used with 4th to 10th grade math students. The responses are all taken from Skill Building problem sets worked on by students in a suburban middle school in central Massachusetts during the 2009-2010 school year. Skill Building is a special type of problem set, where students are presented with problems from a large pool of problems from a single skill. The student completes problems until he either answers 3 questions correct on the first attempt in a row or completes 10 questions in a given day without getting 3 questions correct in a row, in which case he has “exceeded his daily limit” and must return on a different day. Teachers are also able to change the number of correct problems the student must complete in a row as well as the daily limit. For most of our data the number correct in a row was set to either 3 or 5 but a few teachers set this to as many as 15.

The dataset we used includes student responses from 14 different skill types which are each separated into their own dataset. The skills had an average of 800 student responses per skill. Each data point describes whether the student answered correctly or incorrectly on their first attempt of the question. In each skill, we split the student responses into 5 groups in order to run 5-fold cross validation by student. We trained our table model and a KT model on data from 4 of the folds, and then tested the prediction accuracy of our models on the fifth fold. We did this for all 5 folds. The final results reported are combined from all 5 test folds.

Table 1 shows example rows from our dataset.

TABLE 4: EXAMPLE SKILL BUILDING DATA



Student	Skill	Response Sequence
Student A	Integer Addition	'0 1 0 1 1 1'
Student B	Integer Addition	'1 1 1'
Student C	Integer Addition	'0 0 0 1 0 1 1 1'
Student A	Integer Subtraction	'0 1 1 1'
Student D	Integer Subtraction	'0 0 0 0 1 1 1'
…		

4.3 Knowledge Tracing Model Description

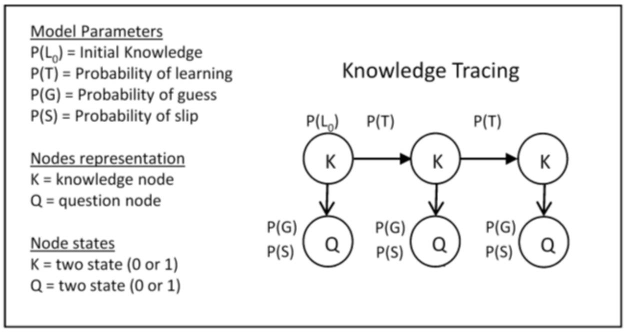


FIGURE 6: KNOWLEDGE TRACING MODEL

The Knowledge Tracing model has been widely used with ITS to model student knowledge and performance. As shown in Fig 1, Knowledge tracing is a typical 2-variable Hidden Markov Model, with one latent and one observable. There are 4 parameters for each skill. The transmission parameters are comprised of the probability that the student knows the skill before starting, $P(L_0)$, and the probability of learning the skill from one problem to the next, $P(Knowledge_{t+1}=T \mid Knowledge_t=F)$ *or* $P(T)$. The emission parameters are the probability of answering correctly while the student doesn't know the skill, "*guessing" P(G)*, and the probability of answering incorrectly when the student does know the skill, "*slip" P(S)*, which can be described as $P(Performance_t|Knowledge_t)$.

In our experiments, we used the Bayes Net Toolbox for Student Modeling[5] to implement Knowledge Tracing which employs expectation maximization (EM) to fit the model parameters to the training data. We set the initial parameters as follow: initial knowledge: $P(L_0)$= 0.50; guess P(G) = 0.14; slip: P(S) = 0.09; learning: P(T) = 0.14, which are found to be the average parameter values across all skills in previous modeling work conducted using ASSISTments.

4.4 Tabling Model Description

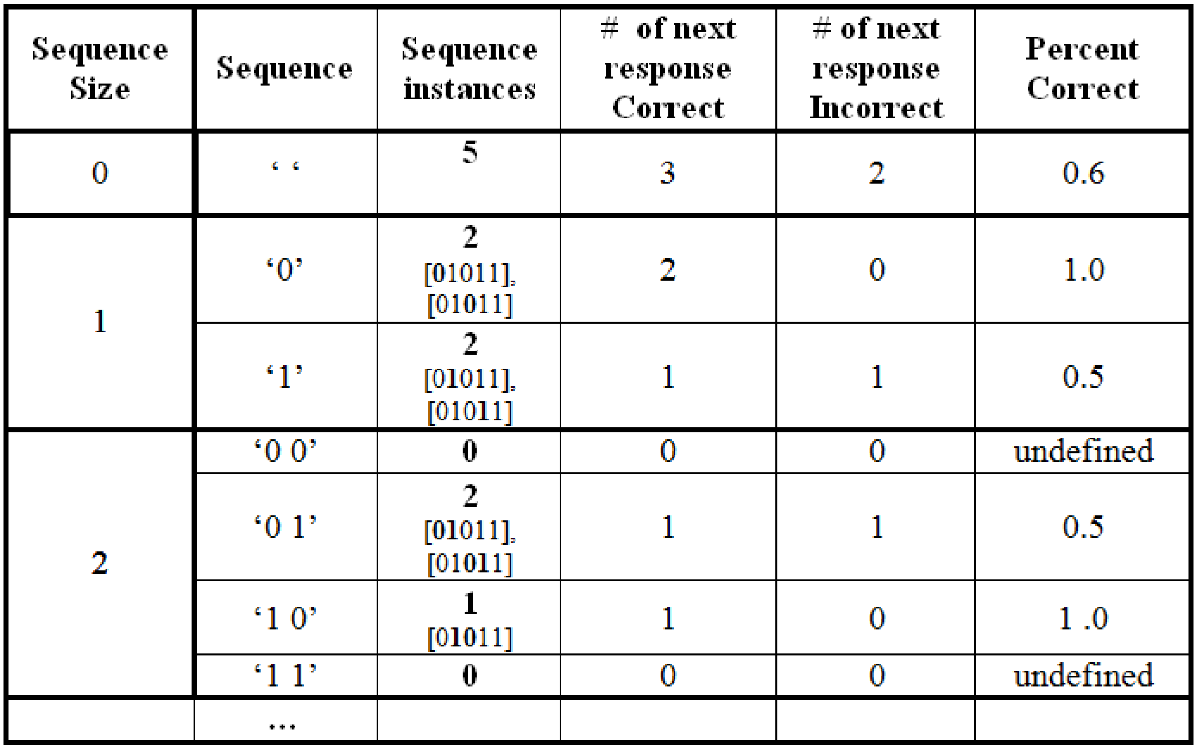
The tabling method evolved out of the simple intuition that common student response sequences may repeat themselves. That is, that overall percent correctness on a problem given a particular past response sequence can predict future performance of other students given the same response sequence. To generalize this idea we take a sequence of correct and incorrect responses, and look at the percentage of correct responses on the next problem. For example, say Student A has answered **'0 1 1'** (0 = incorrect, 1 = correct). We will look at all sequences of student responses that match **'0 1 1'**, observing the response that follows this sequence. So, if 72% of student responses that are preceded by **'0 1 1'** are correct, then we can predict 0.72 for Student A's next response. We describe this prediction as the probability of a correct response given a preceding sequence of **'0 1 1'**, or *P(x=1 | '**0 1 1**') = .72*. Our model becomes a simple table that maps sequences of preceding responses to the percent correct of the follow-up response. We input a sequence of student responses to the table, and use the percent correct from the row corresponding to that sequence as a prediction for the student's next response.

4.4.1 Training the Table

For each skill we train one table. The table has a row for each distinct sequence of preceding responses. To best inform our predictions, we used 5 different sequence lengths, observing 0 through 4 previous responses. The rows are labeled: ‘ ’, ‘**0**’, ‘**1**’, ‘**0 0**’, ‘**0 1**’, ‘**1 0**’, ‘**1 1**’, ‘**000**’, etc. There are a total of 31 rows in each table (1 of length 0, 2 of length 1, 4 of length 2, 8 of length 3, and 16 of length 4). For each row in the table, we record how many instances of this response there are, as well as the number of correct and incorrect responses that follow the sequence. We use these numbers to then calculate the percent correct of next responses.

To maximize the number of data points used in training the model, we attempt to use each response of all the students trained against each different sequence length. That is to say for each student’s first response, we can only train using an empty preceding sequence. For students’ second responses, we can train the empty sequence row as well as the 1-response sequence rows. Table 2 illustrates how the training works, by showing the results of training a table using only one student’s response of **‘0 1 0 1 1’**.

TABLE 5: EXAMPLE FOR A SINGLE STUDENT WITH RESPONSES OF “01011” FOR TRAINING

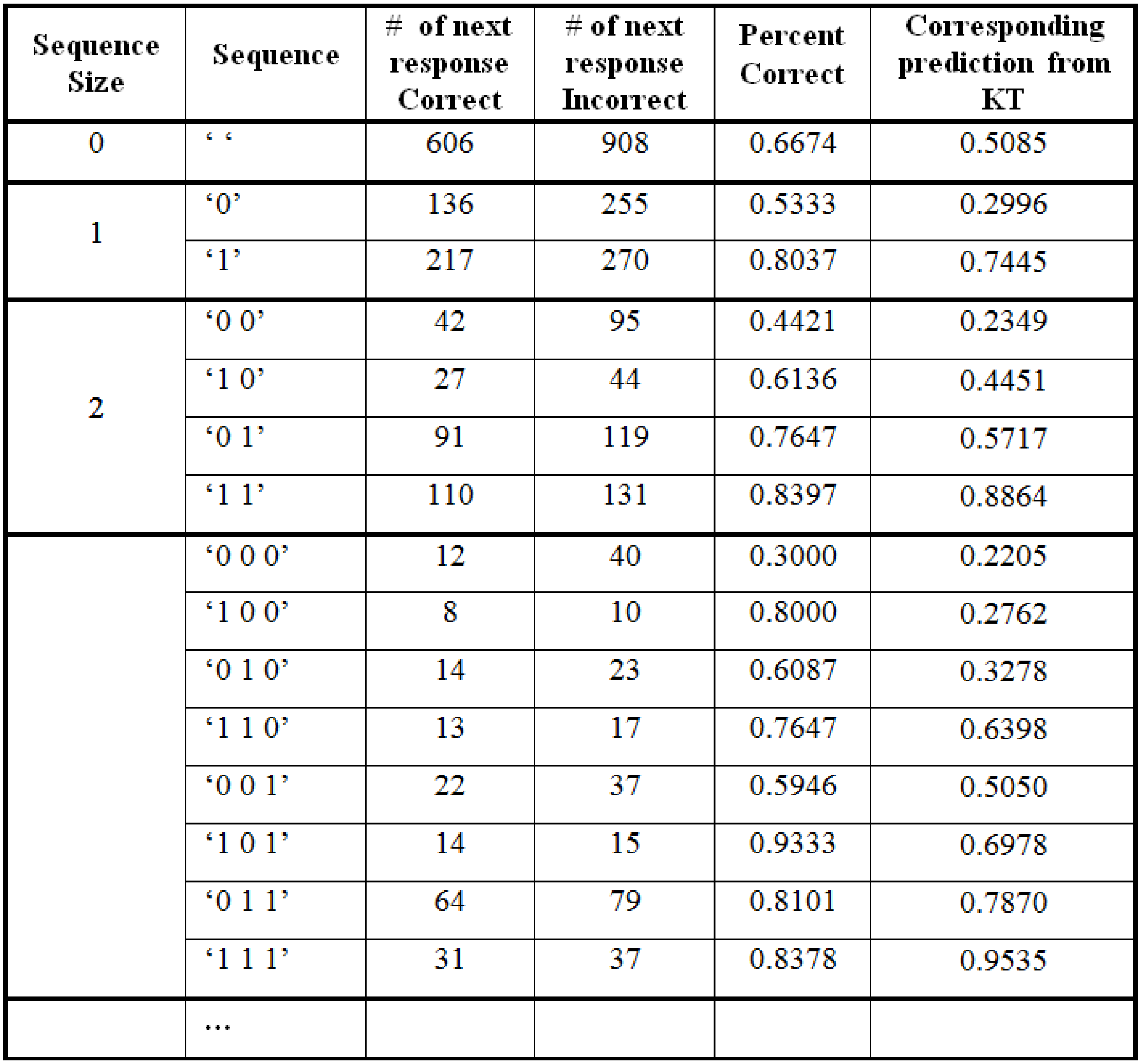


Sequence Size	Sequence	Sequence instances	# of next response Correct	# of next response Incorrect	Percent Correct
0	‘ ‘	5	3	2	0.6
1	‘0’	2 [01011], [01011]	2	0	1.0
	‘1’	2 [01011], [01011]	1	1	0.5
2	‘0 0’	0	0	0	undefined
	‘0 1’	2 [01011], [01011]	1	1	0.5
	‘1 0’	1 [01011]	1	0	1 .0
	‘1 1’	0	0	0	undefined
	…				

The first row of the table shows that given no prior information, all we can do is count the number of correct responses to calculate the percent correct. In the second row, where the preceding sequence is **‘0’**, we see that there are two instances of a **‘0’** sequence in the response **‘0 1 0 1 1’**. Since the response after the **‘0’** is **‘1’** in both cases, we record a percent correct of 1.0 for that sequence. We continue this process until we have covered all sequence sizes, and all possible sequence combinations. By counting the number of sequence instances in the table, we see that this single sequence of 5 responses provides our table model with 15 data points.

We counted responses in this manner for every student response in each skill, which leaves us with a table like Table 3, which shows a fully trained table for one of our skills.

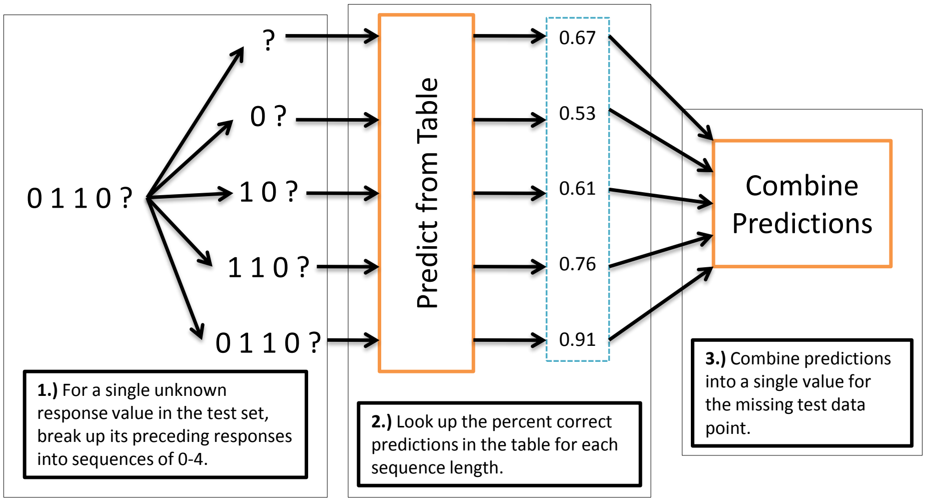
TABLE 6: FULLY TRAINED TABLE FOR SEQUENCE LENGTHS 0-3



Sequence Size	Sequence	# of next response Correct	# of next response Incorrect	Percent Correct	Corresponding prediction from KT
0	‘ ‘	606	908	0.6674	0.5085
1	‘0’	136	255	0.5333	0.2996
	‘1’	217	270	0.8037	0.7445
2	‘0 0’	42	95	0.4421	0.2349
	‘1 0’	27	44	0.6136	0.4451
	‘0 1’	91	119	0.7647	0.5717
	‘1 1’	110	131	0.8397	0.8864
	‘0 0 0’	12	40	0.3000	0.2205
	‘1 0 0’	8	10	0.8000	0.2762
	‘0 1 0’	14	23	0.6087	0.3278
	‘1 1 0’	13	17	0.7647	0.6398
	‘0 0 1’	22	37	0.5946	0.5050
	‘1 0 1’	14	15	0.9333	0.6978
	‘0 1 1’	64	79	0.8101	0.7870
	‘1 1 1’	31	37	0.8378	0.9535
	…				

4.4.2 Making Predictions

To test the predictive power of our tabling method, we made predictions for every student response in the test data set. For each response, we used the preceding 4 responses as evidence (or fewer if 4 did not exist such as when predicting the first four responses), and made a prediction. Figure 2 outlines the full prediction process for a single student’s response.



1.) For a single unknown response value in the test set, break up its preceding responses into sequences of 0-4.

2.) Look up the percent correct predictions in the table for each sequence length.

3.) Combine predictions into a single value for the missing test data point.

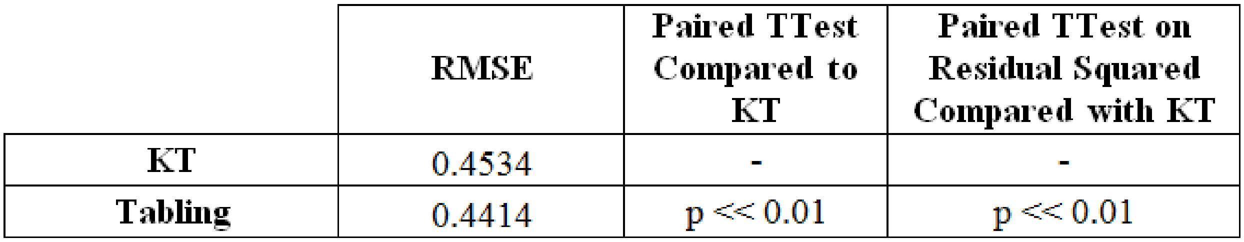
FIGURE 7: TEST DATA PREDICTION PROCESS

First, we would create up to 5 different evidence sequences based on the student's preceding responses. Then, we would look up each of the 5 sequences in the trained table, yielding 5 different percent correct values. We then took these 5 prediction values, and combined them to come up with a prediction of correctness based on the student's preceding responses. For the combination step of the process, we did a simple average of our table's predictions.

4.4.3 Tabling results compared with knowledge TRACING

Table 4 shows RMSE values for our two models predicting student responses on our test data. The table also shows p values from a paired T-Test of each prediction, as well as a T-Test of the squares of the residuals. We found an RMSE of 0.4414 for our tabling model, which is reliably better than the Knowledge Tracing model for this dataset.

TABLE 7: RMSE OF OUR TWO MODELS



	RMSE	Paired TTest Compared to KT	Paired TTest on Residual Squared Compared with KT
KT	0.4534	-	-
Tabling	0.4414	p << 0.01	p << 0.01

4.5 Ensembling Table Predictions with Knowledge Tracing

We had the intuition that since our tabling model is so different from the Knowledge Tracing approach, there would be a potential improvement from combining our results from tabling with the predictions from our KT model. The tabling model takes only the previous four responses into account when predicting, and makes a simple mapping of response sequences to percent correct, whereas Knowledge Tracing can use a longer sequence of responses and models the student’s probability of knowledge while also making predictions. We hoped that the strengths of the two methods could be ensemble into more accurate prediction.

We combined our results in two ways. We first tried taking the simple average of the tabling model’s prediction and KT’s prediction for each response. We refer to the second method in the results as “Average-Min-Max”, shown in equation 1, motivated by the thought that for each prediction, we would use the model that gave the most definitive response. That is to say, if both predictions were above 0.5, we chose the higher of the two. If both were below 0.5 we chose the lower prediction. If one prediction was above our threshold and the other was below, we chose the average of the two predictions.

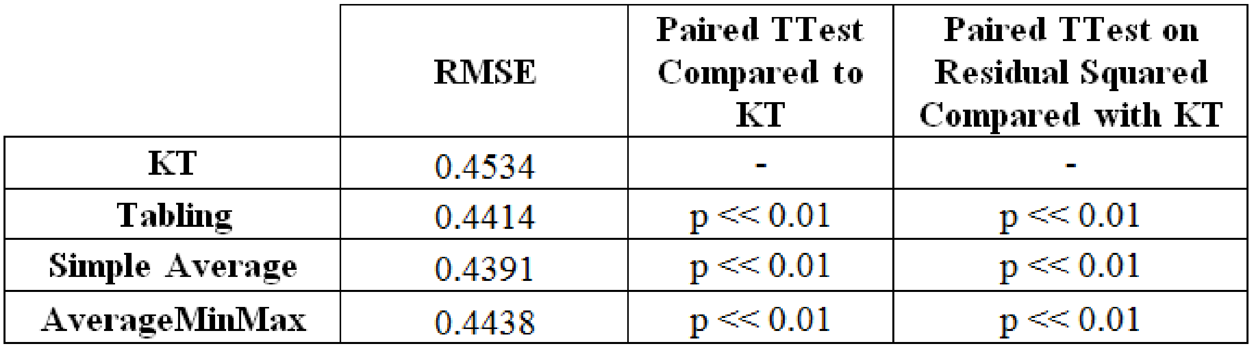
$$avgMinMax(P_t, P_k) = \begin{cases} \min(P_t,P_k) \text{ if } (P_t < 0.5) \text{\textasciicircum} (P_k < 0.5) \\ \max(P_t,P_k) \text{ if } (P_t > 0.5) \text{\textasciicircum} (P_k > 0.5) \\ \text{avg}(P_t,P_k) \text{ if } (P_t < 0.5) \text{\textasciicircum} (P_k > 0.5) \\ \text{avg}(P_t,P_k) \text{ if } (P_t > 0.5) \text{\textasciicircum} (P_k < 0.5) \end{cases}$$

FIGURE 8: "AVERAGE MIN MAX" ENSEMBLING METHOD

4.5.1 Ensembling Results

After ensembling our predictions, we found that taking a simple average gave us a significant improvement in RMSE, while using our AverageMinMax technique gave no reliable improvement.

TABLE 8: RMSE OF OUR ENSEMBLED PREDICTIONS



	RMSE	Paired TTest Compared to KT	Paired TTest on Residual Squared Compared with KT
KT	0.4534	-	-
Tabling	0.4414	p << 0.01	p << 0.01
Simple Average	0.4391	p << 0.01	p << 0.01
AverageMinMax	0.4438	p << 0.01	p << 0.01

4.6 Residual Analysis

We wanted to learn more about exactly how our table model was performing better than KT. We decided to track residuals and RMSE on a per opportunity basis. Figures 3 and 4 show the two graphs for the first 10 student responses. It should be noted that the majority of our student response sequences are 5 or 6 responses long. The behavior of the graphs from 7-10 is based on fewer data points than the rest of the graph.

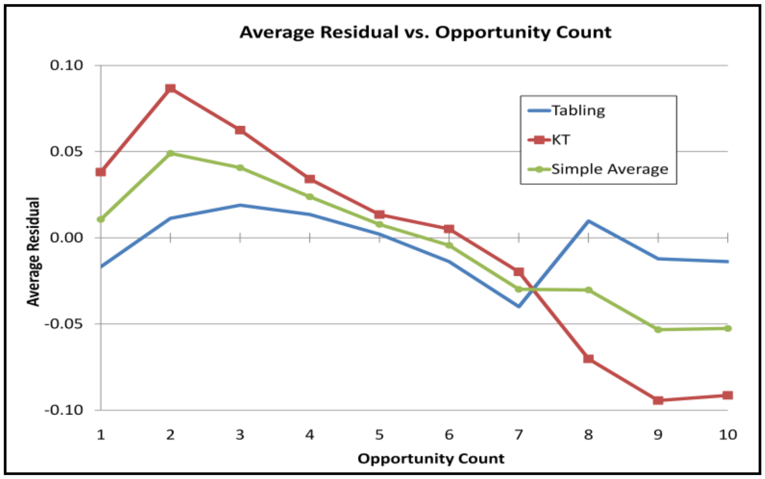


FIGURE 9: AVERAGE RESIDUAL ANALYSIS

It is interesting to note from the residual graph in Figure 3 that KT is underpredicting early in the response sequence. Our intuition is that KT is more conservative in its likelyhood of learning increases, and takes a few responses before it can confidently predict correctness. On the other hand, in the later responses of the residual graph, you can see that KT is over predicting, being overly convinced that the student has learned the skill, when he still may be answering questions incorrectly. This fits with our intuition that tabling, which has no concept of a predecided learning curve, would be resistant to drastic over or underpredictions.

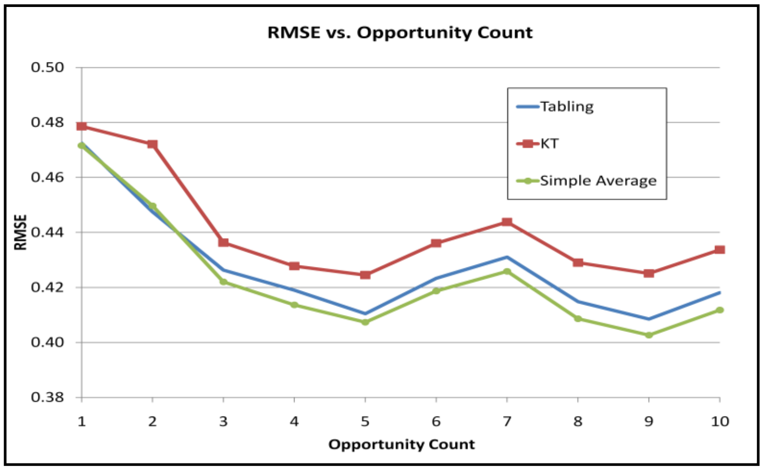


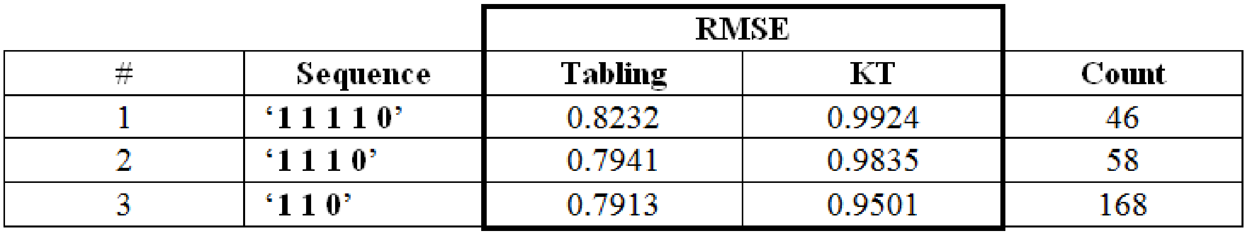
FIGURE 10: RMSE ANALYSIS

We see in the graph of RMSE in Figure 4, that tabling has a sharp decrease in error on the second response, while KT only has a slight improvement from the first response. This suggests again that KT is underpredicting, and cannot confidently predict correctness with only one response. Tabling, whose average residual at response 2 is close to 0, has a much more accurate prediction for the second response in the sequence.

4.6.1 Best and Worst Predicted Sequences

To further examine the performance of tabling versus KT, we found the sequences with the least accurate and the most accurate predictions from our two models. We found that the 3 worst sequences, and the 4 worst sequences were the same for Tabling and KT.

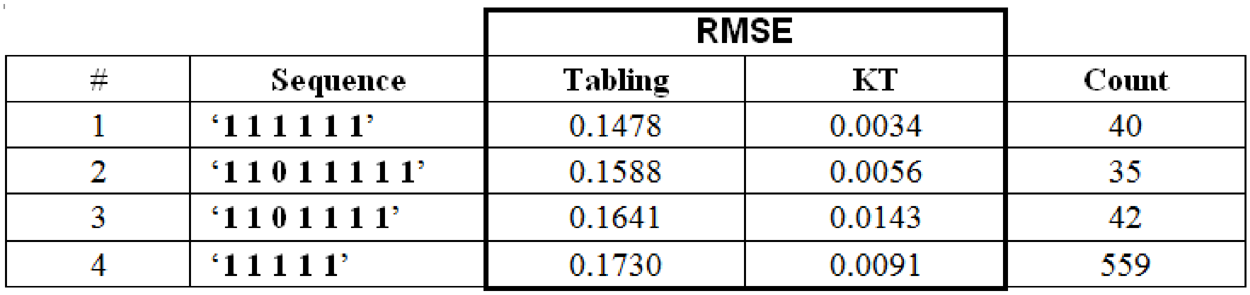
TABLE 9: TOP 3 WORST-PREDICTED SEQUENCES FOR TABLING AND KT



		RMSE		
#	Sequence	Tabling	KT	Count
1	‘1 1 1 1 0’	0.8232	0.9924	46
2	‘1 1 1 0’	0.7941	0.9835	58
3	‘1 1 0’	0.7913	0.9501	168

Table 6 shows the 3 worst sequences for predicting the next response for Tabling and KT. The sequences make intuitive sense, showing responses that have a number of correct responses followed by one incorrect response. It makes sense that it would be difficult to predict the next response, because the previous response could either be an indication that the student doesn’t know the skill, or simply a mistake that would be followed by a correct answer. It is worth noting that for these worst sequences, tabling is more accurate than KT.

TABLE 10: TOP 4 BEST-PREDICTED SEQUENCES FOR TABLING AND KT*



		RMSE		
#	Sequence	Tabling	KT	Count
1	'1 1 1 1 1 1'	0.1478	0.0034	40
2	'1 1 0 1 1 1 1 1'	0.1588	0.0056	35
3	'1 1 0 1 1 1 1'	0.1641	0.0143	42
4	'1 1 1 1 1'	0.1730	0.0091	559

**(ordered by Tabling's top 4, KT had #3 and #4 swapped)*

Table 7 shows the best-predicted sequences. As expected, both models are very accurate when the student has gotten a few questions correct in a row before the current question. It makes intuitive sense that the next question is very likely correct. For the best sequences, KT is more accurate than tabling. This is likely because if a student has gotten 6 questions correct in a row, he definitely knows the skill and will very likely answer correctly. KT will have a very high probability of correctness for this sequence, but Tabling will only look at 4 of the responses, and will average the 5 sequence length predictions. The shorter length predictions will generally be much lower than a longer sequence of correct responses.

4.7 Discussion

What we are suggesting with our Table model is a simple method for doing effective response prediction. Unlike Knowledge Tracing, the Table model offers no interpretability or domain insight. Knowledge tracing can be used for predicting student responses, but it also models a student's probability of knowledge. Interpreting this parameter can be useful in various educational applications. For instance, the Cognitive Tutor uses Knowledge Tracing to determine if a student is finished with the current skill and can move on in the curriculum. The Table model doesn't track any parameters, and doesn't even model students as their own entity. We enter a sequence of evidence responses as input into the table model, and it returns a prediction for the next response. There is nothing else to be learned from the table.

For this reason, we see the table model as having a useful yet limited application. It has the distinct advantage over Knowledge Tracing of being computationally inexpensive. Both training the model and making predictions took seconds, whereas training a dynamic Bayesian Network using Expectation Maximization can take hours using a large enough dataset. If there is the need for interpretability, where predicting the student's response is not enough, our model is not appropriate. However, when the goal is strictly predicting the next response, we believe we have a lightweight compliment to KT that can increase student response prediction accuracy.

4.8 Future Work

The model used in the paper relied strictly on previous responses to make predictions. The tabling idea is a simple idea that could easily be expanded to leverage more features into the prediction. We could conceivably have additional rows to our table which show the percentage of correct responses where the student took longer or shorter than a certain amount of time to answer. We could also calculate the percent correct based on the number of previously answered questions on this skill. This way, in our prediction step of testing shown in Figure 2 would include the 5 predictions based on previous responses as well as predictions based on additional features from the dataset. Wang & Heffernan explored the impact of hints on student performance by finding the percent correct on items based on the number of hints requested [18]. This is another feature that could be added into our tabling model.

Another area for improvement of the table modeling is the step of combining our predictions from the different table entries. For our results in this paper we simply averaged all of the predictions. Perhaps there is a way to better ensemble these predictions so that the more accurate predictions end up having more weight. The method used to ensemble would have to allow for missing values, because not every response tested will have 4 preceding responses.

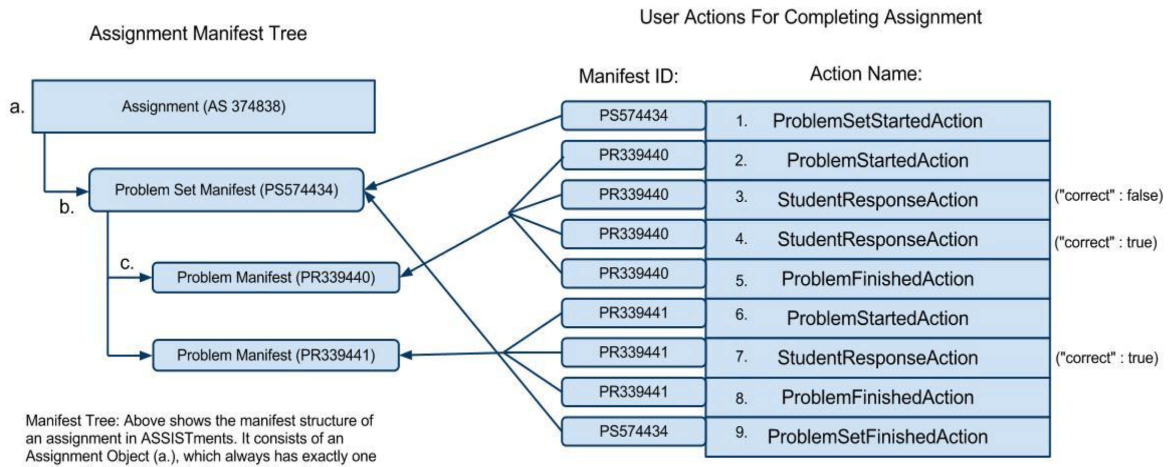
4.9 Tabling Experiment Acknowledgements

This research was supported by the National Science foundation via grant “Graduates in K-12 Education” (GK-12) Fellowship, award number DGE0742503 and Neil Heffernan’s CAREER grant. We would like to thank the organizers of the 2010 KDD Cup at the Pittsburg Science of Learning Center for the Cognitive Tutor datasets and Matthew Dailey for his data preparation assistance.

We acknowledge the contributions of Cristina L. Heffernan, Program Manager and School Liaison for the ASSISTments program. We also acknowledge the many additional funders of ASSISTments Platform found here: http://www.webcitation.org/5ym157Yfr

A.1 New Tutor Specification Documents

A.1.1 Actions Structure


Assignment Manifest Tree
a.
Assignment (AS 374838)
b.
Problem Set Manifest (PS574434)
c.
Problem Manifest (PR339440)
Problem Manifest (PR339441)
User Actions For Completing Assignment

Manifest ID:	Action Name:	
PS574434	1. ProblemSetStartedAction	
PR339440	2. ProblemStartedAction	
PR339440	3. StudentResponseAction	("correct" : false)
PR339440	4. StudentResponseAction	("correct" : true)
PR339440	5. ProblemFinishedAction	
PR339441	6. ProblemStartedAction	
PR339441	7. StudentResponseAction	("correct" : true)
PR339441	8. ProblemFinishedAction	
PS574434	9. ProblemSetFinishedAction	

Manifest Tree: Above shows the manifest structure of an assignment in ASSISTments. It consists of an Assignment Object (a.), which always has exactly one Problem Set Manifest (b.). Each Problem Set manifest has one or more child manifests, which in this case are Problem Manifests (c.). Problem Manifests do not have children.

User Actions: Every action taken by our users are recorded as a "UserAction" Every action records an action type, a user ID, an assignment ID a timestamp of when the action was performed, along with any other action-specific information that is required. Each action is also associated with a single manifest from the Assignment Manifest Tree. For example, a Problem Set Started Action is associated with a Problem Set Manifest, whereas a Student Response Action, which includes the student's answer to a problem, will be associated with a problem manifest. NOTE: See "ASSISTments Tutor JSON" for full JSON text of all of these Actions.

A.1.3 Actions JSON Specification

This is the full JSON of each of the actions from the “ASSISTments Tutor Action Sequence” Diagram.

1. ProblemSetStartedAction(PS574434)

```
{
     "key": "@org.assistments.gwt.student.tutor.event.action.ProblemSetStartedAction/92389-AS374838-1311600IXigqQels",
    "timestamp": 1311600841937,
    "isScaffolding": false,
    "userID": "92389",
    "assignmentID": "AS374838",
    "manifest": "@org.assistments.core.domain.content.Manifest/PS574434"
}
```

2. ProblemStartedAction(PR339440)

```
{
    "key": "@org.assistments.gwt.student.tutor.event.action.ProblemStartedAction/92389-AS374838-1311600842034",
    "userID": "92389",
    "assignmentID": "AS374838",
    "isScaffolding": false,
    "manifest": "@org.assistments.core.domain.content.Manifest/PR339440",
    "timestamp": 1311600842034
  }
```

3. StudentResponseAction(PR339440)

```
  {
    "key": "@org.assistments.gwt.student.tutor.event.action.StudentResponseAction/92389-AS374838-1311600848009",
    "userID": "92389",
    "assignmentID": "AS374838",
    "isScaffolding": false,
    "manifest": "@org.assistments.core.domain.content.Manifest/PR339440",
    "timestamp": 1311600848009,
    "correct": false,
    "problemClassName": "org.assistments.core.domain.content.assess.FillInProblem",
    "response": {
      "valueParts": [
        {
          "type": "ALGEBRA",
          "value": "3"
        }
      ]
    }
  }
```

4. StudentResponseAction(PR339440)

```
 {
    "key": "@org.assistments.gwt.student.tutor.event.action.StudentResponseAction/92389-AS374838-1311600850448",
    "userID": "92389",
    "assignmentID": "AS374838",
    "isScaffolding": false,
    "manifest": "@org.assistments.core.domain.content.Manifest/PR339440",
    "timestamp": 1311600850448,
    "correct": true,
    "problemClassName": "org.assistments.core.domain.content.assess.FillInProblem",
    "response": {
      "valueParts": [
        {
          "type": "ALGEBRA",
          "value": "2"
        }
      ]
    }
  }
```

5. ProblemFinishedAction(PR339440)

```
 {
    "key": "@org.assistments.gwt.student.tutor.event.action.ProblemFinishedAction/92389-AS374838-1311600850451.0",
    "userID": "92389",
    "assignmentID": "AS374838",
    "isScaffolding": false,
    "manifest": "@org.assistments.core.domain.content.Manifest/PR339440",
    "timestamp": 1311600850451
  }
```

6. ProblemStartedAction(PR339441)

```
 {
     "key": "@org.assistments.gwt.student.tutor.event.action.ProblemStartedAction/92389-AS374838-1311600851612",
     "userID": "92389",
     "assignmentID": "AS374838",
     "isScaffolding": false,
     "manifest": "@org.assistments.core.domain.content.Manifest/PR339441",
     "timestamp": 1311600851612
  }
```

7. StudentResponseAction(PR339441)

```
 {
     "key": "@org.assistments.gwt.student.tutor.event.action.StudentResponseAction/92389-AS374838-1311600857991",
     "userID": "92389",
     "assignmentID": "AS374838",
     "isScaffolding": false,
     "manifest": "@org.assistments.core.domain.content.Manifest/PR339441",
     "timestamp": 1311600857991,
     "correct": true,
     "problemClassName": "org.assistments.core.domain.content.assess.FillInProblem",
     "response": {
       "valueParts": [
         {
           "type": "ALGEBRA",
           "value": "4"
         }
       ]
     }
  }
```

8. ProblemFinishedAction(PR339441)

```
 {
     "key": "@org.assistments.gwt.student.tutor.event.action.ProblemFinishedAction/92389-AS374838-1311600857993.0",
     "userID": "92389",
     "assignmentID": "AS374838",
     "isScaffolding": false,
     "manifest": "@org.assistments.core.domain.content.Manifest/PR339441",
     "timestamp": 1311600857993
  }
```

9. ProblemSetFinishedAction(PS574434)

```
  {
     "key": "@org.assistments.gwt.student.tutor.event.action.ProblemSetFinishedAction/92389-AS374838-13116008dqZsxU",
     "timestamp": 1311600858019,
     "userID": "92389",
     "isScaffolding": false,
     "manifest": "@org.assistments.core.domain.content.Manifest/PS574434",
     "assignmentID": "AS374838"
  }
```

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4 (2) 167-207.

[2] Bahador, n., pardos, z., heffernan, n. T. And baker, r. 2011. Less is more: improving the speed and prediction power of knowledge tracing by using less data. Educational data mining 2011 conference

[3] Breiman, l., friedman, j. H., olshen, r. A., and stone, c. J. 1984. Classification and regression trees. Wadsworth international group, belmont, california

[4] Caurana, r, niculescu-mizil, a. Crew, g., ksikes. 2004. Ensemble selection from libraries of models. Proceedings of the 21st international conference on machine learning(icml ’04)

[5] Chang, k., beck, j., mostow, j., and corbett, a. 2006. A bayes net toolkit for student modeling in intelligent tutoring systems. Intelligent tutoring systems, 8th international conference, its 2006, 104-113

[6] Corbett, a. T., and anderson, j. R. 1995. Knowledge tracing: modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4, 253–278.

[7] Corbett, A., & Anderson, J. R. (2001) Locus of feedback control in computer-based tutoring: Impact on learning rate, achievement and attitudes. *In Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems (J. Jacko, A. Sears, M. Beaudouin-Lafon & R. Jacob, eds.),* pp. 245-252, ACM Press, New York.

[8] Koedinger, k.r., anderson, j.r., hadley, w.h., and mark, m.a. 1997. Intelligent tutoring goes to school in the big city. International journal of artificial intelligence in education, 8, 30–43

[9] Pardos, z. A. And heffernan, n.t. 2010. Modeling individualization in a bayesian networks implementation of knowledge tracing. Proceedings of the 18th international conference on user modeling, adaptation, and personalization

[10] Pardos, z. A. And heffernan, n. T. 2010. Navigating the parameter space of bayesian knowledge tracing models: visualizations of the convergence of the expectation maximization algorithm. In proceedings of the 3rd international conference on educational data mining, 161-170.

[11] Pardos, z. And heffernan, n. In press. Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. Journal of machine learning research

[12] Mendicino, M., Razzaq, L. & Heffernan, N. T. (2009) Comparison of Traditional Homework with Computer Supported Homework. Journal of Research on Technology in Education, 41(3), 331-359

[13] Quinlan, r. 1993. C4.5: programs for machine learning. Morgan kaufmann publishers inc., san francisco, ca, usa.

[14] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A., Rasmussen, K.P. (2005). The Assistment Project: Blending Assessment and Assisting. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, pp. 555-562. Amsterdam: IOS Press.

[15] Ritter, s., harris, t., nixon, t., dickison, d., murray, c., towle, b. 2009. Reducing the Knowledge tracing space. In proceedings of the 2nd international conference on Educational data mining, cordoba, spain, 151–160

[16] Ryan, Ray. (2009) Google Web Toolkit Architecture: Best Practices For Architecting Your GWT App. Talk from Google I/O Developer Conference 2009. Moscone Center, San Francisco, http://www.google.com/intl/de-DE/events/io/2009/sessions/GoogleWebToolkitBestPractices.html

[17] Silvernail, D. L., & Lane, D. M. M. (2004). The impact of Maine's one-to-one laptop program on middle school teachers and students (Report #1). Gorham, ME: Maine Education Policy Research Institute, University of Southern Maine Office.

[18] Wang, y. & Heffernan, n. T. 2011. The "assistance" model: leveraging how many hints and attempts a student needs. The 24th international flairs conference