Incorporating an Intelligent Tutoring System into CS1

Leen-Kiat Soh
National Center for Information Technology in Education (NCITE)
Department of Computer Science and Engineering
University of Nebraska
256 Avery Hall, Lincoln, NE 68588-0115 USA
E-mail: lksoh@cse.unl.edu

ABSTRACT

Intelligent tutoring systems (ITSs) have been used to complement classroom instruction in recent years, and have been shown to facilitate learning. We incorporate an ITS named Intelligent Learning Materials Delivery Agent (ILMDA) into our CS1 course and collect evidence to validate two hypotheses: (1) The ITS improves student learning, (2) The ITS "learns" to tutor the students more efficiently and/or effectively. Our method of inquiry includes collecting data tracked while a student interacts with the ITS, post-test scores, and exam scores. We also use control and treatment groups, as well as different versions of the ILMDA in our experiments. Based on the results, we see indications that support the above two hypotheses.

Categories and Subject Descriptors

K.3 [**Computers and Education**]: Computer Uses in Education, Computer Science Education

General Terms

Design, Experimentation

Keywords

Intelligent Tutoring System, CS1

1 INTRODUCTION

Several Intelligent Tutoring Systems (ITSs) have been tested on humans and have proven to facilitate learning. For example, there have been successful ITSs such as SHERLOCK [9], PACT [8], BodyChat [2], ANDES [6], AutoTutor [4], and SAM [2], and a host of animated pedagogical agents or guidebots (e.g., [7, 10]). In particular, ITSs are motivated and justified from the observation that (1) human tutors produce impressive learning gains (between .4 and 2.3 standard deviation units over classroom teachers) [5] even though the vast majority of tutors have modest domain knowledge, no training in pedagogical techniques, and rarely use the sophisticated tutoring strategies of intelligent tutoring systems, and (2) software AI systems produce learning gains of approximately .3 to 1.0 standard deviations units compared with students learning the same content in a classroom.

In our research, we use an ITS called Intelligent Learning Materials Delivery Agent (ILMDA) [12, 13] in our CS1 course. This ILMDA system is an intelligent agent that delivers learning materials based on the usage history of the learning materials, the student static background profile (such as GPA, majors, interests, and courses taken), and the student dynamic activity profile (based on their interactions with the agent). The agent uses the profiles to decide, through case-based reasoning (CBR) which learning modules (examples and problems) to present to the students [1]. Each learning module consists of three components: (1) a tutorial (e.g., "What is recursion?"), (2) a set of related examples, and (3) a set of exercise problems to assess the student's understanding of the topic. Based on how a student progresses through the module and based on his or her profile, ILMDA chooses the appropriate examples and exercise problems for the student. In this manner, the agent customizes the delivery of the learning materials. Further, this system has a suite of machine learning mechanisms, allowing itself to track its own reasoning steps and scoring decisions that lead to successful student interactions. Indeed, the ILMDA system learns about which problems have been rated incorrectly in terms of its level of difficulty, which instructional strategies have been used successfully, and so on to improve its performance over time.

The ILMDA system differs from traditional ITSs in three aspects: (1) its modularization of the course content, the delivery mechanism, and the knowledge bases, separating them into standalone components, (2) its utilization of agent intelligence where the ILMDA system is able to learn how to deliver its learning module better, and (3) its own accountability of usefulness for evaluation. The agent uses a suite of knowledge bases: instructional strategies—e.g., which problems to select—encoded as cases, adaptation heuristics, and similarity weights for CBR, the quality of learning materials, how the learning materials are used as a set of attributes, and the models of students based on their on their aptitude and motivation background and dynamic interaction with the agent. Most ITSs are able to customize the learning material for different students, but without the machine learning capabilities and a modular instructional expertise-content-system design like the ILMDA's.

The overall, long term goal of our research is to study effectiveness and efficiency of the intelligent tutoring system (i.e., ILMDA) in impacting student learning gains in CS1. This is because the diversity in motivation, self-efficacy, and aptitude of the students in CS1 is high compared to upper-division CS courses and that the ILMDA design is aimed to adapt to such diversity: customizing the tutorials, examples, and problems to fit the different student behaviors and needs. We see this system as a software tutor that also helps the instructor by providing information about which problems that the students do poorly on, which examples that the students spend most time in, and so on.

Towards achieving the aforementioned long-term goal, the objective of our current research is to investigate two hypotheses:

(1) The ILMDA system improves student learning, in terms of their scores in post-tests and exams,

(2) The ILMDA system "learns" to tutor the students more efficiently and/or effectively.

Investigating the first hypothesis will allow us to evaluate the correctness of the ILMDA system, including its reasoning process, the course content (learning materials), and how student participation impacts their performances in the tests and exams. Investigating the second hypothesis will allow us to refine not only the ILMDA system but also our CS1 instruction. By observing how the ILMDA system re-rates the problems, sequences the delivery of problems, and changes its heuristics and weights, the investigation will provide insights to our instructional knowledge for teaching CS1.

2 BACKGROUND

2.1 CS Curriculum

At the Computer Science and Engineering (CSE) Department of the University of Nebraska, the CS1 course is the first requirement course for a computer science or computer engineering major. About half of the students in the course are from other majors as CS1 is a requirement for them as well. Some of our CS1 students have had some programming background in high school or have taken a CS0 course with us. The rest have little or no programming background at all.

Our CS1 course teaches problem solving with computers including problem analysis and specification, algorithm development, program design, and implementation in a high-level programming language. It has 3 weekly lectures and 2-hour weekly laboratory sessions. Laboratory assignments are designed to develop mastery of the high-level programming language and practices [14]. The programming language for our CS1 course is Java.

Our CS1 course is also part of our Reinventing CS Curriculum Project at the CSE Department [11]. This project has designed and developed a placement exam for CS0/CS1, two sets of closed laboratories (CS1 and CS2), and a set of learning objects (for CS1). The laboratories complement the lecture topics, reinforcing on key topics and even introducing topics not covered in the lectures. Each laboratory is structured so that the students have a set of activities to follow and experiment with and a set of problems to solve.

2.2 Enrollment and Demographics

The enrollment for this course has been between 55-90 per semester for the past 2-3 semesters. Most students are Caucasian male, with 1-2 African Americans, 3-5 Asian Americans, and 3-6 female of different ethnicity. Most students are in their first or second semester as a freshmen or their first semester as a sophomore. About half of the students are Computer Science or Computer Engineering majors. There are students from Math Education, Industrial Engineering, and Electrical Engineering. Further, the drop out rate of the course is about 12-15%.

2.3 ILMDA

The ILMDA system is based on a three-tier methodology, as shown in Figure 1. It consists of a graphical user interface (GUI) front-end application, a database backend, and the ILMDA reasoning in between. A student user accesses the learning material through the GUI. The agent captures the student's interactions with the GUI and provides the ILMDA reasoning module with a parametric profile of the student and environment. The ILMDA reasoning module performs case-based reasoning (CBR) to obtain a search query (a vector of search keys) to retrieve and adapt the most appropriate example or problem from the database. The agent then delivers the example or problem in real-time back to the user through the interface.

Each case has a situation, a solution, and an outcome function. The situation specifies the student profile and the characteristics of the learning materials seen thus far, within a session, by the student. The solution identifies the characteristics of the next appropriate examples or problems to be delivered. Thus, basically, a case is an instructional strategy, mapping a student's observed behavior to what he or she should read next.

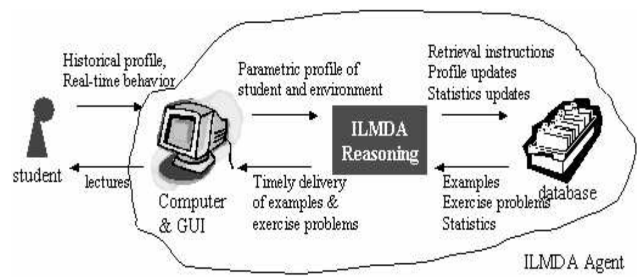

Figure 1. The ILMDA system and environment.

When a student starts the ILMDA application, he or she is first asked to login. This associates the student with his or her profile information. The information is stored in two separate tables. All of their generally static information, such as name, major, interests, etc., is stored in one table, while the student's dynamic information (i.e., how much time, on average, they spend in each section; how many times they click the mouse in each section, etc.) is stored in another table. After a student is logged in, he or she selects a topic and then views the tutorial on that topic. Following the tutorial the agent looks at the student's static profile, as well as the dynamic actions the student took in the tutorial, and searches the database for a similar case. The agent then adapts the output of that similar case depending on how the cases differ, and uses the adapted output to search for a suitable example to give to the student. After the student is done looking at the examples, the same process is used to select an appropriate problem. Again, the agent takes into account how the student behaved during the example, as well as his or her background profile. After the student completes an example or problem, they may elect to be given another.

The ILMDA system also has a suite of machine learning capabilities. Most notably is the case-based learning. It tags cases that have been used successfully (leading to successful interactions with the students) with higher utility and uses those cases more often. It also stores new cases as a result of adapting old solutions to new situations. Other learning capabilities include changing the adaptation heuristics and similarity weights. Interested readers are referred to [12, 13].

2.4 Fall 2004 Results

We deployed the ILMDA system in our CS1 course in the Fall 2004 and Spring 2005 semesters. Initial results gathered from the Fall 2004 deployment were encouraging, prompting the experiments to be described in Sections 3 and 4. Our Fall 2004 evaluation consisted of the following. The Fall 2004 CS1 course had 3 lab sections. We used one lab sections as the control group, and the other two as the treatment group. We conducted the study only on selected lab topics (Table 1). The treatment group was required to go through the ILMDA system to learn about the topic before attending the corresponding lab. Each student in all three sections was required to take the same post-test and participate in the lab activities (and to fill out a worksheet activity form). We used the post-test and worksheet activity scores to assess student performance in conceptual understanding and practical understanding, respectively.

First, we found that with machine learning capabilities, our ILMDA prototype was able to deliver learning materials to the students more efficiently. For example, for the topic *Recursion*, ILMDA agents with activated machine learning capabilities gave fewer examples to the students (2.22 vs. 3.05) and fewer problems to the students (10.95 vs. 15.11), and the average percentage of problems answered correctly was higher (66.2% vs. 60.2%).

Further, based on a Likert-scale survey, we found that students perceived the tutorials, examples, and problems to be useful (> 3.0 out of a 5.0 scale). The students especially thought that the problems to be useful (> 3.5 out of 5.0).

However, the Fall 2004 study did not collect the data with enough resolution: some students spent little time on the ILMDA tutorials; some students did. We should have distinguished these students and conducted separate analyses of their performances. Since we did not separate them out, the results, though encouraging, did not give indications for the validity of ILMDA. We were not able to say whether ILMDA helped students learn.

3 RESEARCH DESIGN

In Spring 2005, we deployed the ILMDA system for five topics in the laboratories: File I/O, Event Driven Programming, Exception, Inheritance, and Recursion. See Table 1 below for the schedule of the labs.

Table 1. Schedule and topics for the five labs used in the study.

Date	Topics and Labs
Feb 14	File I/O
Feb 28	Event Driven Programming1
Mar 7	Exception
Apr 4	Inheritance
Apr 25	Recursion

There were three lab sections:

(1) Static-ILMDA group: In this section, students interacted with the ILMDA system, but without any customization such that the system simply displays problems by the order of their difficulty levels. This served as the baseline group.

(2) Heuristic-ILMDA group: In this section, students interacted with the ILMDA system that delivers problems based on encoded instructional strategies stored as cases. This resembled a traditional ITS without machine learning capabilities.

(3) Learning-ILMDA group: In this section, students interacted with a fully-functional ILMDA system that, in addition to deliver problems based on encoded instructional strategies, learns on its own to improve its performance in guiding students to answer correctly one of the more difficult questions.

The key difference in our experiment between the Fall 2004 and Spring 2005 deployments is the versions of ILMDA. In Fall 2004, there were only two. That is, we used a version that was capable of machine learning and one that was not. We found, to some extent, that the agent that used machine learning was more effective and efficient, as indicated in Section 2.4 earlier. This semester, we added another version: a version that did not reason, and simply provided the set of problems without taking into account ILMDA's interaction with a student. Basically, this version acted as a static "e-book" with set examples and problems.

We define an interactive session between ILMDA and a student as a session between the time when the student logs in and the time when the student logs out. We further define the degree of success of such an interaction in several terms: (1) the number of problems delivered, (2) the number of problems answered correctly, and (3) the difficulty level of the problems answered correctly. The goal of the ILMDA system is to deliver as small as possible a number of problems, have them answered correctly as many as possible, and have one of the most difficult problems answered correctly. We consider an ITS that satisfies the above conditions as one that is efficient and effective.

4 ANALYSIS

In this section, we present and discuss our results. Though we do not have enough data to substantiate our findings in answering the hypotheses posed in Section 1, we are able to draw observations supporting the effectiveness and efficiency of ILMDA.

Table 1 shows a subset of attributes that ILMDA tracked and the averages of each student. We selected 9 students, 3 from each final course grade of A, B, and C for this analysis. We selected these nine due to several reasons. First, not all students agreed to allow their data to be used in our study. Second, students dropped from the course rendering some of the data not usable. Third, not all students used ILMDA though it was required as part of the lab activities. Fourth, out of the remaining students with valid data, we intentionally selected students in three groups: A, B, and C students based on their final course grades so we could compare their interaction with ILMDA.

Table 2 shows a subset of attributes that ILMDA tracked. We see that A and B students spent roughly the same time with ILMDA (except for student A3). The significant difference is between C students and A/B students. C students reviewed only on average 1.1 examples, 0.4 problems, and gotten 0.5 problems correct. Note that the variable #problems indicates the number of different problems a student have seen, and #correct indicates the number of problems a student have answered correctly. It is possible for #correct to be greater than #problems if a student reviews the topic multiple times (for example, before an exam).

Table 3 shows the five post-tests and the total exam scores of each student. From Table 3, we can see that student A3's exam score is lower than all three B students. This is the same student who diligently spent much time going through many problems using ILMDA. This is also the same student who did very poorly in her first exam due to a medical condition. (Later, I allowed her more time to finish the remaining exams. The mental condition that she suffered from was her uncontrollable shaking making it close to impossible to write and type for certain periods of time during the exams.)

Table 2. 9 students of different course grades and how they interacted with ILMDA.

Student	Total Time (s)		
	Tutorial	**Example**	**Problems**
A1	875545	16549	404201
A2	1612450	70490	391575
A3	9215451	4424110	3300442
B1	1193849	33721	170570
B2	1450744	21668	319111
B3	16165	3081	302916
C1	337859	147837	0
C2	13856	9106	33291
C3	261660	32942	0
Mean(A)	3901149	1503716	1365406
Mean(B)	886919	19490	264199
Mean(C)	204458	63295	11097

Student	#exam	#prob	correct	max. difficulty
A1	2.0	7.8	8.2	5.0
A2	1.4	5.0	3.0	2.8
A3	4.0	21.0	25.2	8.0
B1	1.8	3.4	2.2	2.5
B2	0.8	7.2	6.0	3.0
B3	1.4	9.0	4.6	1.4
C1	1	0	0	0
C2	1.2	1	0.8	0.5
C3	1	0.2	0.8	1.4
Mean(A)	2.5	11.3	12.1	5.3
Mean(B)	1.3	6.5	4.3	2.3
Mean(C)	1.1	0.4	0.5	0.6

Table 3. 9 students of different course grades and how they fared in each lab and in the exams. Each post-test's maximum score is 10; the maximum total exam score is 300.

Student	Post-Test Scores					Exam
	T1	**T2**	**T3**	**T4**	**T5**	
A1	8.0	7.0	10.0	7.0	10.00	285.75
A2	8.0	6.0	10.0	6.0	7.00	265.75
A3	10.0	7.0	10.0	10.0	10.00	228.00
B1	5.0	7.0	9.0	8.0	10.00	249.75
B2	9.0	8.0	8.0	7.0	10.00	251.50
B3	10.0	7.0	10.0	6.0	10.00	243.25
C1	8.0	9.0	9.0	6.0	0.00	207.75
C2	0.0	10.0	10.0	0.0	9.00	178.25
C3	0.0	5.0	8.0	9.0	9.00	210.75

4.1 Hypothesis 1

Our first hypothesis is that "The ILMDA system improves student learning, in terms of their performances in post-tests and exams". That means, if a student spent more time on ILMDA, then his or her post-test score for that topic should be higher. Or if a student reviewed more problems or examples, then his or her post-test score for that topic should be higher.

The total amount of time spent by a student on ILMDA slightly correlated with his or her post-test score (0.21). The total amount of time spent by a student on ILMDA had no correlation with his or her total exam score (0.02).

The total number of problems, examples, and correct answers reviewed or produced by a student on ILMDA slightly correlated with his or her total exam score (0.27). The total number of problems, examples, and correct answers reviewed or produced by a student on ILMDA slightly correlated with his or her post-test scores (0.35).

This gives some indications that ILMDA might be worth the trouble. Students who spent a lot of time reading the ILMDA tutorials, but going through only a few examples or problems, did not have higher exam scores. On the other hand, students who went through more examples or problems received better scores on the exams and the post-tests.

Based on only these nine students, though we cannot confidently validate our hypothesis, we do have some evidence to support the hypothesis.

Our next step is to continue to collect data over the next several semesters to obtain a statistically significant analysis.

4.2 Hypothesis 2

Our second hypothesis is that "The ILMDA system 'learns' to tutor the students more efficiently and/or effectively."

To measure this, we had three ILMDA versions at work, as discussed earlier in Section 3.

Since the students who used the static-ILMDA version did not review problems (or reviewed only 1 or 2), we excluded them from our discussions here. There were 3 students who used the learning-ILMDA, and 4 students who used the heuristic-ILMDA. In learning-ILMDA, students reached the average difficulty level of 6.0 after reviewing an average number of 1.7 examples and 6.7 problems. In heuristic-ILMDA, students reached the average difficulty level of 6.2 after reviewing an average number of 2.4 examples and 11.1 problems. This indicates that learning-ILMDA can be more efficient than heuristic-ILMDA: it gave fewer examples and problems while remaining almost as effective as heuristic-ILMDA.

Though this was based on only 7 students, the results were somewhat encouraging. Once again, we cannot confidently validate our hypothesis, but we do have some evidence to support it.

Our next step is to continue to collect data over the next several semesters to obtain a statistically significant analysis.

4.3 Other Observations

Table 4 shows the statistics for individual topics. It is observed that students in general re-visited the topics to review the problems (since #correct > #problem for all topics). Students reviewed the most examples for File I/O even though each topic had 4-5 examples. Similarly, students reviewed the most problems for File I/O even though each topic had about the same number of problems (~20). This could be because File I/O was the first topic delivered by ILMDA in the labs and students were *excited* about using an intelligent tutoring system (ITS) and also had more time to interact with one. Students re-visited exceptions and recursion more often than they did with other topics, probably because these two topics were less intuitive than the others. This is good to see. We were surprised by the lack of re-visits for the Inheritance/Polymorphism topic, and the least amount of time spent on that topic as well, as this is an important topic in CS1.

Table 4. Collected statistics for each ILMDA topic.

Topics	Time Spent (s)	#exam	#prob	#correct	max-diff.
File I/O	4006863	3.00	14.67	13.33	5.00
Event-Driven Programming	3530879	1.50	4.38	7.00	7.00
Exceptions	3649422	2.29	5.71	13.00	7.40
Inheritance/ Polymorphism	1982691	1.00	5.43	8.50	5.00
Recursion	2794626	1.57	4.00	11.00	6.17

5 CONCLUSIONS

Our study set out to test two hypotheses in terms of the impact of an intelligent tutoring system, called ILMDA, on student performance. With limited data, we observed indications that the ILMDA system (1) improves student learning, in terms of their performances in post-tests and exams, and (2) "learns" to tutor the students more efficiently and/or effectively.

Therefore, we believe that the ILMDA system is viable and useful, and is a good start for achieving our long-term goal of creating a highly customizable and adaptive CS1 course for students of diverse backgrounds, aptitudes, and motivations. If the ILMDA can be shown to be effective in improving student learning and efficient in delivering only the necessary learning materials (examples and problems) to the students, then it can be further designed to customize its delivery based on other factors. Further, if the ILMDA can be proved to be capable of learning useful instructional strategies, then the system can actually adapt to its own reasoning process, making it even more autonomous.

Our next step is to collect more data. We will perform data mining such as the use of contrast rules to find causal-relationships between pairs of attributes or parameters collected by the ILMDA system. Our future work includes deploying the system in the next several semesters and extending the system with more topics.

ACKNOWLEDGEMENT

The ILMDA project was partially supported by the Great Plains Software Technology Initiative and the Computer Science and Engineering Department at the University of Nebraska. The educational research was part of a Peer Review in Teaching Project. The author would like to thank Todd Blank and L.D. Miller for their programming work.

REFERENCES

[1] Blank, T., L. D. Miller, L.-K. Soh and S. Person (2004). Case-Based Learning Mechanisms to Deliver Learning Materials, in *Proc. Int. Conf. on Machine Learning & Applications (ICMLA'2004)*, Louisville, KY, December 16-18, pp. 423-428

[2] Cassell, J. and H. Vilhjálmsson (1999). Fully Embodied Conversational Avatars: Making Communicative Behaviors Autonomous, *Autonomous Agents and Multi-Agent Systems*, **2**:45-64.

[3] Cassell, J., M. Annany, N. Basur, T. Bickmore, P. Chong, D. Mellis, K. Ryokai, J. Smith, H. Vilhjálmsson, and H. Yan (2000). Shared Reality: Physical Collaboration with a Virtual Peer, presented at the *ACM SIGCHI Conference on Human Factors in Computing Systems*, April 1-6, The Hague, The Netherlands.

[4] Graesser, A. C., van Lehn, K., Rosé, C. P., Jordan, P. W., and Harter, D. (2001). Intelligent Tutoring Systems with Conversational Dialogue, *AI Magazine*, **22**(4):39-51.

[5] Graesser, A. C., K. Wiemer-Hastings, P. Wiemer-Hastings, R. Kreuz, and the Tutoring Research Group (1999). AutoTutor: A Simulation of a Human Tutor, *J. Cognitive Systems Research*, **1**(1):35-51.

[6] Gertner, A. S. and van Lehn, K. (2000). ANDES: A Coached Problem-Solving Environment for Physics, in *Proc. ITS'2000*, 133-142.

[7] Johnson, W. L., J. W. Rickel, and J. C. Lester (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments, *International Journal of Artificial Intelligence in Education*, **11**:47-78.

[8] Koedinger, K. R., Anderson, J. R., Hadley, W. H., and Mark, M. A. (1997). Intelligent Tutoring Goes to School in the Big City, *J. Artificial Intelligence in Education*, **8**(1):30-43.

[9] Lesgold, A., S. Lajoie, M. Bunzo, and G. Eggan (1992). SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job, in J. H. Larkin and R. W. Chabay (eds.) *Computer Assisted Instruction and Intelligent Tutoring Systems*, 201-238, Hillsdale, NJ: Lawrence Erlbaum.

[10] Rickel, J. and W. L. Johnson (1999). Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control, *Applied Artificial Intelligence*, **13**(4):343-382.

[11] Samal, A., G. Nugent, L.-K. Soh, J. Lang, and S. Person (2005). Reinventing Computer Science Curriculum at University of Nebraska, in PytlikZillig, M. Bodvarsson, and R. Bruning (Eds.) *Technology-Based Education: Bringing Researchers and Practitioners Together*, Chapter 4, pp. 63-82, Information Age Publishing.

[12] Soh, L.-K., T. Blank, L. D. Miller, and S. Person (2005). ILMDA: An Intelligent Learning Materials Delivery Agent and Simulation, in *Proc. Int. Electro-Information Tech. Conf. (EIT'2005)*, May 22-25, Lincoln, NE.

[13] Soh, L.-K., T. Blank, and L.D. Miller (2005). Intelligent Agents that Learn to Deliver Online Materials to Students Better: Agent Design, Simulation and Assumptions, in C. Chaoui, M. Jain, V. Bannore, and L. C. Jain (eds.) *Studies in Fuzziness and Soft Computing: Knowledge-Based Virtual Education*, Chapter 3, Volume 178, pp. 49-80.

[14] Soh, L.-K., A. Samal, S. Person, G. Nugent, J. Lang (2005). Closed Laboratories with Embedded Instructional Research Design for CS1, in *Proc. SIGCSE'2005*, St. Louis, MO, February 23-27, pp. 297-301.