Toward Next-Generation Intelligent Tutors: Adding Natural Handwriting Input

Lisa Anthony, Jie Yang, Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
+1 (412) 268-8856
{lanthony, yang+, koedinger}@cs.cmu.edu

ABSTRACT
This article describes our work exploring and adapting handwriting recognition-based interfaces in intelligent tutoring systems for students learning algebra equation-solving. The research is motivated by the hypothesis that handwriting as an input modality may provide significant advantages over typing in math learning. We describe our approach and report results to date in exploring the use of handwriting recognition in interfaces for math learning, from both a technical and a pedagogical perspective. We have found that handwriting input can provide benefits to students learning math, and continue to pursue further technical and pedagogical enhancements.

Keywords
Handwriting input, handwriting recognition, mathematics learning, intelligent tutoring systems, algebra equation solving.

1. INTRODUCTION
Mathematic skills are not only essential to successful careers in sciences or engineering, but also crucial to even non-scientists in daily life. For example, a stay-at-home mom might need to use simple algebra skills to figure out her monthly budget. However, algebra is typically one of the subjects in which students suffer the most. Our project aims to improve student performance and engagement in algebra via intelligent tutoring systems that accept natural handwriting input (Figure 1).

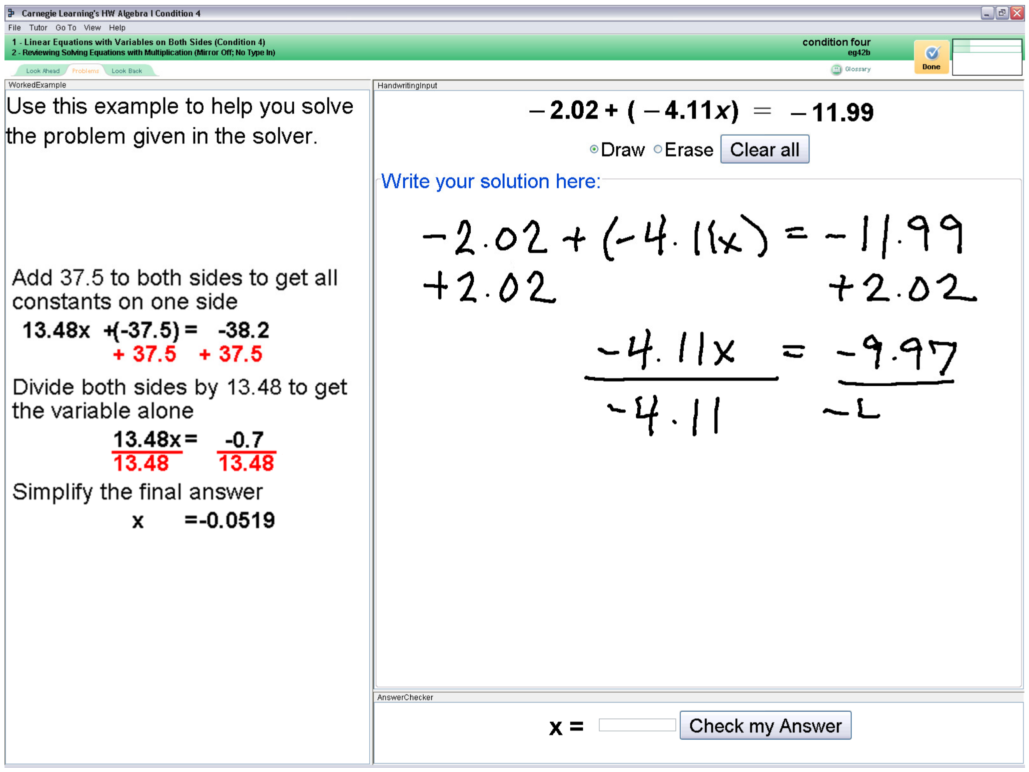
Many schools throughout the United States now incorporate computers including PDAs or TabletPCs and intelligent tutoring systems as a regular part of classroom instruction [16]. An intelligent tutoring system (ITS) is educational software that can monitor the student as she works at her own pace, and tailor feedback, step-by-step hints, and even the curriculum to address a student's particular needs. This self-pacing provides an opportunity for teachers to give more individual attention to students that need it most. One particular type of ITS is known as "Cognitive Tutors". These tutors are quite successful in the classroom—when compared to traditional classroom instruction, Cognitive Tutors raise student achievement one standard deviation [6], turning *C* students into *B* students. Our goal is to improve mathematics learning in Cognitive Tutors even further, via use of multimodal and multimedia interface technologies, turning those *B* students all the way into *A* students.

A user interface bridges information exchanges between a student and an ITS. Different modalities in input and output of the interface can significantly affect efficiency of the ITS. Although *output* modality has been studied with respect to learning, including the use of animations, diagrams and talking heads (*e.g.*, [9]), little attention has been paid to the effects of *input* modality on learning[1] and most ITS rely on standard menu-based GUI interfaces. We believe that the input modality is extraneous to the problem-solving process—it itself is not relevant to the math concept being practiced. The input modality can *interfere* with learning, however, by imposing *extraneous cognitive load* on the student—mental effort not directly related to the learning process. Typing and menu-based interfaces are especially guilty of this, because they provide only cumbersome support for representing and manipulating math. An interface that can more directly support the standard notations for mathematics that the student is learning could reduce extraneous cognitive load and increase learning.

We have developed and tested a prototype ITS using handwriting input for learning algebra equations. Our latest results show that handwriting input has benefits both for general usability and for learning. Though this work is in the domain of high school algebra learning, it is likely to generalize to other types of math and to other levels of students.

[1] Note that input modality here refers to the modality of generation by the student, and the output modality is the modality presented to the student by the system.

Prepared for *IEEE Multimedia*.
Please cite the definitive version at http://www.computer.org/portal/web/csdl/doi/10.1109/MMUL.2008.73



1 - Linear Equations with Variables on Both Sides (Condition 4)
2 - Reviewing Solving Equations with Multiplication (Mirror Off; No Type In)

condition four
eg42b

Use this example to help you solve the problem given in the solver.

Add 37.5 to both sides to get all constants on one side

$$\begin{array}{rcr} 13.48x + (-37.5) & = & -38.2 \\ +\ 37.5 & & +\ 37.5 \end{array}$$

Divide both sides by 13.48 to get the variable alone

$$\frac{13.48x}{13.48} = \frac{-0.7}{13.48}$$

Simplify the final answer

$$x = -0.0519$$

$$-2.02 + (-4.11x) = -11.99$$

Write your solution here:

$$\begin{array}{lcr} -2.02 + (-4.11x) & = & -11.99 \\ +2.02 & & +2.02 \\ \dfrac{-4.11x}{-4.11} & = & \dfrac{-9.97}{-L} \end{array}$$

$x =$

Figure 1. A screenshot of our current prototype system illustrating the use of worked-examples based instruction and handwriting input.

2. MOTIVATION AND APPROACH

Classrooms in America are now supplementing teaching time in the classroom with computer time using intelligent software tutors, in many different courses. Math is one domain tutoring systems can help; a recent survey found that American high school students have a poorer mastery of basic math concepts than their counterparts in other industrialized nations [11].

There is evidence that the use of handwriting interfaces could have advantages in math learning environments [2], [3]. One factor to this advantage may come from more direct manipulations of pen-based input. Furthermore, students may be more *fluent* in use of handwriting when solving algebra equations, which can impose less extraneous cognitive load. Another factor is the improved support for meaningful two-dimensional spatial information in math. For example, the placement of the x in $2x$ *vs.* 2^x significantly changes the meaning of the expression. Handwriting is a more flexible and robust input modality for representing and manipulating such spatial relationships than typing or pointing [2].

Since handwriting recognition is not perfectly accurate, there is a trade-off in this application domain between challenges in improving recognition accuracy and the need for detailed instructional feedback. One strategy is to modify the instructional paradigm. For example, during worked-examples-based instruction, a common instructional method (*c.f.*, [13]) is to let students study example problems with the solutions before solving their own problems. This studying gives the student a low-risk opportunity to practice the problem-solving concepts she is about to solve on her own and provides a sort of instructional *feed-forward*—from the example to the subsequent problem. Such a feed-forward approach perhaps would reduce the need for detailed instructional *feedback* at each step, allowing the system to delay providing a recognition hypothesis until more information is available. This could be extended to the extreme, perhaps even only providing instructional feedback at the final answer. Figure 1 shows the type of annotated worked examples we use in our tutor.

A second strategy is to improve handwriting recognition accuracy. Our current prototype maximizes the *a priori* recognition accuracy by an adaptation technique [5]. In addition, due to the special nature of a learning task and the needs of a learner, students may not require the system to immediately output a recognition hypothesis. Students are not so much interested in what the tutor thinks they wrote as in whether what they wrote was the correct answer. If the students must spend time correcting the system's recognition, we have merely exchanged one type of cognitive load for another, and might not expect to see any differences in learning. This actually leads to a flexibility which allows the system to delay committing to a recognition hypothesis until more information is available. The tutoring system can provide the student with other forms of information (such as worked examples) to alleviate reliance on this type of recognition feedback, and can provide the recognition engine with more information such as the context of the problem or knowledge about the student's skills to improve the confidence of certain recognition hypotheses before these must be revealed to the student (*i.e.*, at the end of the problem).

A fundamental goal of this project is to determine how the use of handwriting input will help student learning. User studies with students engaged in learning are critical to our approach. Our early studies motivating further research were conducted in the laboratory. As part of our affiliation with the Pittsburgh Science of Learning Center (PSLC), *in vivo* studies are emphasized. These studies take place in a real-world classroom setting in which the experimental system is compared to an authentic control condition. The participating classrooms use Cognitive Tutor Algebra all year, and only some classes or students switch to the experimental system during the study. This is an emerging trend in educational research (*e.g.*, [7], [15]). Being part of the PSLC provides us with this unique opportunity to conduct more real-world, applied studies and see the experimental software and technology in use in the field.

3. CURRENT PROTOTYPE

3.1 System Description

We have developed a prototype system using a foundation of state-of-the-art ITS and handwriting recognition components. Cognitive Tutors are a class of ITS that pose authentic problems to students and emphasize *learn-by-doing* [1]. In Cognitive Tutor Algebra, students represent a given scenario algebraically in a spreadsheet, graph functions, and solve equations with a symbol manipulation tool. The Tutor can provide step-by-step feedback and help. Our prototype adds a handwriting interface to already-existing Cognitive Tutoring Algebra lessons that have been previously developed and field-tested extensively.

Figure 1 shows a screenshot of our system in which the student is solving the problem "$4069.64 + 434.17y = 262.47y + (-3804.02)$" by referring to the worked example on the left side of the screen. The student enters his/her solution process in handwriting and types his/her final answer in the text field at the bottom of the screen. The Cognitive Tutor Algebra curriculum has several equation-solving units. We can deploy our prototype for any of these units. Some problem types we use include $ax+b=c$, $x/a + b=c$, $a/x=c$, $ax+bx=c$, $ax+bx+c=d$, etc.

The recognizer we use is the Freehand Formula Entry System (FFES) [12]; rather than developing a robust recognizer from scratch, our approach can blaze a trail for using handwriting engines in more real-world applications without being an expert in recognition algorithms. We chose FFES because it was open-source and had the highest accuracy rates in training experiments on our corpus when compared with several other state-of-the-art recognizers such as JMathNotes [14] and Microsoft's TabletPC SDK[2]. FFES uses the CIT character recognizer, a nearest-neighbor classifier based on a 48-dimensional feature space. FFES also includes a mathematical expression parser (DRACULAE) [17]. The main advantage of this handwriting system for accuracy on our corpus over other engines is that this parser is designed to be effective for the spatial relationships between mathematical symbols and numbers.

3.2 Training a *Walk-Up-and-Use* Engine

In recognition terminology, *writer-independent* means that the system or engine has been trained on samples from a number of different users, in contrast to *writer-dependent*, which means the training samples come from the same user who will be using or testing the system. In general, user-dependent accuracy rates are higher than user-independent rates in most recognition technologies because differences in handwriting (or speech, etc.) vary more widely *across* users than *within* users. But a user dependent system requires more training samples from each individual user. In learning environments, it is not feasible for students to spend time training the system with no learning objectives. On the other hand, it is difficult to embed the handwriting training task into learning-oriented tasks because the system cannot provide adequate feedback on the learning aspects without good *a priori* recognition accuracy. Therefore, we attempt to improve recognition without upfront training for each user.

To train the recognizer and achieve the best *a priori* accuracy *before* incorporating it into the tutoring system, we have collected a corpus of data from over 40 high school and middle school algebra students. The corpus contains 16,191 characters grouped into 1,738 equations. The symbol set includes 21 symbols: the digits, common variable letters, and simple algebraic operators. We have on average 404 samples per user (17 samples per symbol per user). We further studied strategies of training the recognizer. Our experiment results indicate that ensuring an equal representation of each user's handwriting will prevent the system from having a bias toward a particular style of writing. Each of 40 users had to write just *two* samples per symbol to converge to best *a priori* accuracy, averaging 83% on individual symbols or 71% on full equations (accuracy on full equations was determined via Levenshtein string distance formula [7]) [5]. Note that very few samples per symbol per user are needed in this method. Future application developers can use the guidelines in [5] to help determine how much data to collect for their own applications.

4. USER EVALUATIONS

Here we discuss results from two preliminary laboratory studies establishing grounds for further exploration of these issues. Once we have a theoretical basis and good preliminary evidence showing that our interventions would most likely help students, we can move to the classroom.

[2] http://msdn2.microsoft.com/en-us/developercenters/default.aspx

Our first study was designed to explore what advantages, if any, handwriting-based input has for math input on the computer [2]. Forty-eight college-level students entered given algebra and calculus equations on the computer over the course of 45 minutes in several modalities, including typing via Microsoft Equation Editor, and handwriting. No handwriting recognition was used in this study; the aim was to determine what the raw usability of each modality would be in this task without the interference of individual system nuances. The study showed that students who entered math equations via handwriting input were *three times faster*, were less prone to errors in input, and enjoyed their experience more than those who typed [2]. In the classroom, this can translate to increased depth or breadth of coverage by virtue of the extra time afforded, and to improve student motivation by virtue of their increased engagement.

We next took this one step further and applied handwriting-based input to a learning situation [3]. In this study we compared high school students solving problems with a simple type-in interface with a handwriting input space. Both the typing and the handwriting interfaces were simple, unstructured, and unconstrained input spaces. No special-purpose math menus or widgets were provided in either modality. The 48 participants each came to our lab for a 2-hour session. Results from this study showed that students using handwriting finished the learning session in *half the time* of their typing counterparts, yet we found no difference in their learning [3]! In a classroom situation, this would allow teachers to give students more practice or move on to more advanced material in the curriculum sooner. The speed benefits of handwriting only increased as the problems got more complex, for instance, in problems with fractions. In their own words, students commented that handwriting "made it easier" and "takes a shorter time" [3]—statements that lend support to the hypothesis that handwriting involves less extraneous cognitive load. Handwriting also emerged as the winning modality in terms of user preference. All students were exposed to both modalities in a second phase of the study. As the pie chart in figure 2 shows, students showed a strong preference for handwriting. Out of 46 total students, over 80% preferred handwriting [3].

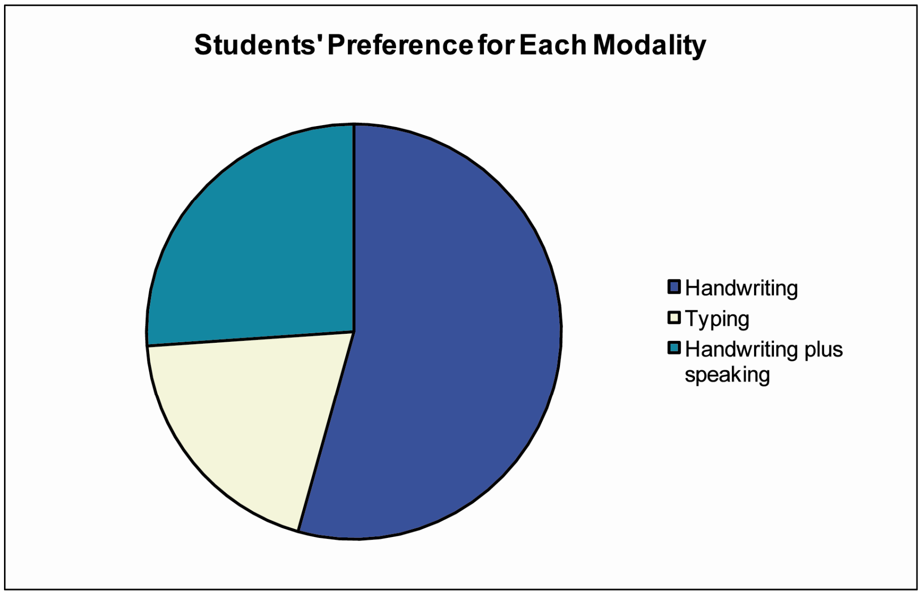


Figure 2. Pie chart distribution of students' preferences for each modality by the three conditions in the study. The majority of students chose one of the two conditions in which handwriting was used.

5. CONCLUSIONS AND FUTURE WORK

We have summarized our ongoing work in adapting, deploying and testing handwriting-based interfaces in ITS for algebra equation solving; some further technical details can be found in [4]. Our studies have provided the first positive evidence in favor of handwriting interfaces for *learning* applications. Students can learn the same amount in half the time using handwriting input *vs.* typing-based interfaces. This, combined with the expected decrease in extraneous cognitive load, can allow students to focus more directly on the mathematics, finishing with a deeper understanding of the material.

We will continue to evaluate later versions of our prototype in classroom studies. Our next steps involve exploring other strategies for enhancing recognition accuracy. Once these engine enhancements are in place, a summative user study will compare the handwriting-based tutor to the standard Cognitive Tutor classroom practice, focusing on differences in learning and cognitive load. In addition to its technical contributions, this work will also help shed light on how to effectively design instructional paradigms that can take advantage of the benefits of handwriting input for learning.

Besides handwriting, we consider using speech as an additional modality for error repair. Systems perform better when the modality of repair is different than the modality of entry, in part because users tend to over-enunciate (in speech) or trace heavily (in writing) and these patterns do not match the system's model (*c.f.*, [10]). Speech seems logical because it can be highly accurate when the symbol vocabulary is small, and because it does not require a return to the keyboard. It may also be pedagogically effective to allow students to self-explain their problem-solving process in speech as they write. We also consider using multimedia output. Our system can present to students animated diagrams helping to explain the problem-solving process when the student needs a hint. Mayer's work exploring the principles of multimedia learning [9] serve as design guidelines for when to include multimedia output and what this output should contain.

While this work is in the domain of high school algebra learning, it is likely to generalize to other types of math and to other levels of students. Future efforts in this line of work will explore other domains such as calculus, geometry, and physics, which rely even more heavily on spatial information in annotations.

6. ACKNOWLEDGMENTS

Work supported by NSF Award #SBE-03554420 and the Pittsburgh Science of Learning Center. The authors are grateful to many people in Carnegie Mellon for their discussions and support.

7. REFERENCES

[1] Anderson, J.R., Corbett, A.T., Koedinger, K.R., and Pelletier, R. (1995) Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences 4*, 167-207.

[2] Anthony, L., Yang, J., and Koedinger, K. R. (2005) Evaluation of Multimodal Input for Entering Mathematical Equations on the Computer. *Proc. CHI'2005*, 1184-1187.

[3] Anthony, L., Yang, J., and Koedinger, K. R. (2007) Benefits of Handwritten Input for Students Learning Algebra Equation Solving. *Proc. AIEd'2007*, 521-523.

[4] Anthony, L., Yang, J., and Koedinger, K. R. (2007) Adapting Handwriting Recognition for Applications in Algebra Learning. *Proc. ACM Multimedia EMME'2007 Wkshp*, 47-56.

[5] Anthony, L., Yang, J., and Koedinger, K. R. Data-Driven Methodology for Handwriting Recognition in Mathematics. Under review.

[6] Corbett, A.T. (2001) Cognitive Computer Tutors: Solving the Two-Sigma Problem. *Proc. of User Modeling*, 137-147.

[7] Koedinger, K. R. and Aleven, V. (2007) Exploring the assistance dilemma in experiments with Cognitive Tutors. *Educational Psychology Review 19*, 239-264.

[8] Levenshtein, V.I. (1966) Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady 10*, 707–710.

[9] Mayer, R. E. (2001) Multimedia Learning. New York: Cambridge University Press, Boston, MA.

[10] Oviatt, S. (2000) Taming Recognition Errors with a Multimodal Interface. *Communications of the ACM*, 45-51.

[11] PISA (2005). International Outcomes of Learning in Mathematics Literacy and Problem Solving. http://nces.ed.gov/pubs2005/2005003.pdf

[12] Smithies, S., Novins, K. and Arvo, J. (2001) Equation Entry and Editing via Handwriting and Gesture Recognition. *Behaviour & Information Technology 20*, 53-67.

[13] Sweller, J. (1988) Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science 12*, 257-285.

[14] Tapia, E. and Rojas. R. (2004) Recognition of On-Line Handwritten Mathematical Expressions using a Minimum Spanning Tree Construction and Symbol Dominance. In *Graphics Recognition: Recent Advances & Perspectives* (eds. Lladós, J. & Kwon, Y.-B.), *Lecture Notes in Computer Science 3088*, 329-340.

[15] White, B. Y. and Frederiksen, J. R. (1998) Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction 16*, 3-118.

[16] Wood, C. (2002) Technology and Education. *PC Magazine*: http://www.pcmag.com/article2/0,4149,15154,00.asp.

[17] Zanibbi, R., Blostein, D., and Cordy, J.R. (2002) Recognizing Mathematical Expressions Using Tree Transformation. *IEEE Trans. on Pattern Analysis & Machine Intelligence 24*, 1455-1467.

BIOGRAPHIES

Lisa Anthony is currently finishing her PhD at the Human-Computer Interaction Institute, Carnegie Mellon University. She has a BS and MS in computer science (Drexel University, 2002), and an MS in Human-Computer Interaction (Carnegie Mellon, 2006). Her research interests include multimodal interfaces such as pen, speech, and gestures. Specifically, she focuses on applying these input modalities to new domains with a user-centered design perspective.

Jie Yang is a Senior Systems Scientist with Human Computer Interaction Institute, Carnegie Mellon University. He received his BS degree from Wuhan University of Technology in 1982, MS degree from Hunan University in 1985, and PhD degree from the University of Akron in 1994, all in electrical engineering. His current research interests include multimodal interfaces, computer vision, and pattern recognition.

Kenneth R. Koedinger is a Professor of Human-Computer Interaction and Psychology at Carnegie Mellon University. He has a BS in Mathematics, a MS in Computer Science (University of Wisconsin, 1984, 1986), and a PhD in Psychology (Carnegie Mellon, 1990). His research goal is to create educational technologies that increase student achievement and he has developed computer simulations of student thinking. Dr. Koedinger is also the Carnegie Mellon Director of the Pittsburgh Science of Learning Center.

CONTACT INFORMATION

Lisa Anthony
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
(412) 268-8856
lanthony@cs.cmu.edu

Jie Yang
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
(412) 268-8020
jie.yang@cs.cmu.edu

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
(412) 268-7667
koedinger@cmu.edu