Handwriting Interaction for Math Tutors: Lessons for HCI in Education

Lisa Anthony
UMBC
Information Systems
1000 Hilltop Circle
Baltimore, MD 21250
lanthony@umbc.edu

Jie Yang
Carnegie Mellon University
HCII
5000 Forbes Avenue
Pittsburgh, PA 15213
yang+@cs.cmu.edu

Kenneth R. Koedinger
Carnegie Mellon University
HCII
5000 Forbes Avenue
Pittsburgh, PA 15213
koedinger@cmu.edu

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract
We discuss our position on key issues in HCI in education based on lessons learned during our work on incorporating handwriting interaction into intelligent tutoring systems (ITS) for algebra learning. Pen-based computing offers opportunities to support more natural and transparent (invisible) interactions, such as handwriting and sketch, for students in the math domain, allowing them to focus on the learning task. We describe the technical and research challenges we encountered in making an ITS-embedded handwriting recognition system usable by middle and high school algebra students. Our efforts can be informative in designing usable and pedagogically-effective educational technology systems in the future.

Keywords
Intelligent tutoring systems, Cognitive Tutors, algebra, handwriting recognition, human-computer interaction, HCI in education, child-computer interaction.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Human Factors

Introduction

HCI in education often emphasizes only one half of the dual nature of the field: either focusing only on the usability of the system, or only on the pedagogy. Both aspects must be considered to realize robust benefits for students using new technology in their learning experiences. We describe relevant lessons learned in the context of a project to incorporate handwriting interaction into intelligent tutoring systems (ITS). ITS provide individual, targeted learning experiences for students. Although highly effective, they are still not at levels comparable to the best human tutors [4]. Traditional ITS have offered students only typing and mouse interfaces for solving math problems on the computer. Work on alternate modalities has extensively explored spoken tutorial dialogue (e.g., [5]); very few projects have used handwriting or sketch interaction in tutoring systems (but see e.g., [6]). ITS could be more effective for math if their interfaces were more transparent (invisible), allowing students to focus on getting the answer, rather than how to input it.

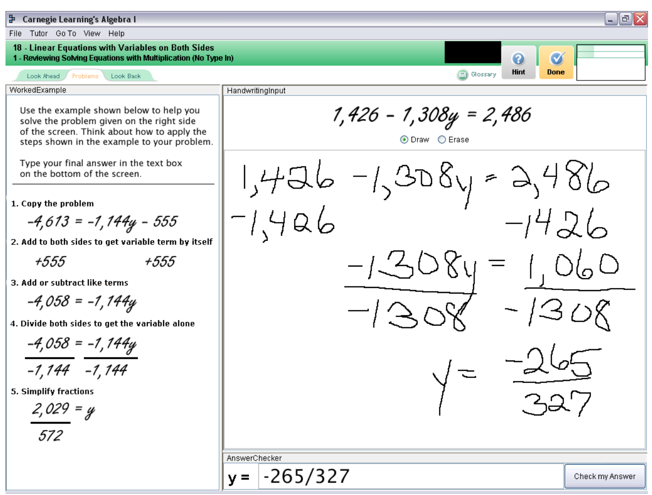

Figure 1. Conceptual prototype of the Cognitive Tutor Algebra lesson enhanced with handwriting interaction.

We have worked on a project incorporating handwriting interaction into a type of ITS called Cognitive Tutors (CT) [4]; see Figure 1. CTs contain a model of student knowledge and are able to provide just-in-time help and tailor the curriculum to each individual student's strengths and weaknesses. We capitalized upon this model of student knowledge to address some of the technical challenges we encountered in developing an algebra ITS with a handwriting input recognizer. We first provide some background on the project itself, and then describe lessons learned and our position on key issues for the future of HCI for educational technology.

The Handwriting-Enabled Algebra Tutor

We began exploring handwriting input for math tutors for several reasons. Typing and mouse interactions do not allow students to easily enter math expressions, which often contain important spatial information such as fractions or exponents. Interfaces to enter such expressions either require children to learn new syntax (e.g., "2^x" for 2^x) or to use complex menu structures. Both of these approaches increase the cognitive burden of the interface, detracting from the learning task. Early studies in the project established that handwriting input is more usable than typing for entering math [3], and that students can achieve the same learning gains in less time with a handwriting-based tutor than a typing one [2], establishing a firm motivation to continue.

Using handwriting interaction in algebra ITS requires the ability to recognize and understand the student's writing. Handwriting recognition approaches vary, but none have achieved 100% accuracy, meaning there will be errors in the system's interpretation of the student's answer. Work on the general usability of handwriting and sketch interactions has focused on methods of recognition error correction [8]. In our case, asking children to correct the system's recognition errors while they are engaged in learning tasks is questionable, given the original goal to reduce cognitive burden imposed by the interface. In fact, this correction process may impose its own cognitive load, whether or not recognition is highly accurate [7]. We prefer a model in which the system's recognition process is performed behind-the-scenes, and the result of this process then generates the instructional feedback the student receives upon entering his/her answer.

In order to achieve this standard, we make use of the tutoring system's contextual information. The Cognitive Tutor's model of the student's knowledge can be used to refine the handwriting recognition process and increase accuracy, enabling more accurate instructional feedback. In our work, we were able to raise the accuracy of a handwriting recognizer nearly 10 percentage points through the use of this context [1]. For more information on the technical approach taken to integrate the Cognitive Tutor contextual information with the handwriting recognition process, please see [1]. To further reduce the impact of recognition errors on the student's learning experience, we used an approach in which the student types his/her final answer, and only if it is wrong does the system process the problem-solving steps to find the error.

Informing HCI in Education

During this project, we encountered several research challenges that may be informative to future HCI researchers and designers of educational systems.

Transparency

It is critical that designers of educational technology consider the impact of the technology on the learning experience. Does the technology streamline the experience, or does it add complexity? Although not in all cases does added complexity interfere with learning, deciding to move forward with such a design should be undertaken only with extreme care. Typically, we recommend a transparent (invisible) interaction, allowing the student to focus on the lesson to be learned rather than (at best) bells and whistles or (at worst) usability disasters of educational technology.

A corollary to this recommendation which applies when designing educational technology with embedded artificial intelligence is to limit the impact that system errors in recognition, understanding, classification, etc, have on the student's learning experience. In our case, we did this by not revealing to the students the system's underlying recognition of their work. This may not be appropriate in all cases, but it should be a conscious design choice and evaluated with students.

Pedagogy

Pedagogical issues must be considered when designing educational software. Teams should include or consult with researchers or practitioners in education and the learning sciences. Too often, systems for learning in the HCI literature fail to do this, leading to a system which is not grounded in good pedagogical science. In our case, one of our team is an experienced researcher in the fields of learning science and educational technology. We also worked closely with teachers in the classroom to ensure that the systems we evaluated satisfied their concerns about effectiveness and probability of success with their own students.

In our system, this collaboration led to the idea of including annotated worked examples in the tutor (see Figure 1), motivated by limitations of the handwriting recognition accuracy in allowing real-time, step-by-step feedback to be provided with high confidence. If we had not considered the pedagogical impact of students on removing this feedback from the tutor for technical reasons, we might not have seen any learning gains.

Validity

Speaking of learning, during an educational technology project, evaluations of both usability *and* learning gains must be conducted. In initial stages, it is appropriate to conduct laboratory studies, both to highlight usability issues early and to prevent negative impact of an immature system on classroom students. When the project matures, the technology must be evaluated in the context of a real classroom / learning environment.

In fact, measuring learning gains quantitatively is sometimes overlooked in the HCI literature when designing and developing educational technology. It is good early on to collect qualitative data on whether the students and teachers like the technology and want to use it again, but as the project matures, it is imperative to evaluate whether or not real learning is occurring. Do the students demonstrate improved knowledge of the targeted concepts after using the technology? Designing such evaluations is an area in which having education researchers or practitioners on the team or to consult can help a great deal.

Conclusion

We have described several key lessons learned in the context of our project on enabling handwriting input in an ITS for algebra. These lessons can be informative to future educational technology systems in terms of balancing pedagogical and technological demands.

Acknowledgements

We thank the teachers who allowed us to test early versions of this system in their classrooms. The tutor project was supported by the Pittsburgh Science of Learning Center (NSF Grant #SBE-0354420).

References

[1] Anthony, L. *Developing handwriting-based intelligent tutors to enhance mathematics learning*. PhD thesis, Carnegie Mellon University, 2008.

[2] Anthony, L., Yang, J. & Koedinger, K.R. Benefits of handwritten input for students learning algebra equation solving. In *Proc. AIEd 2007*, 521-523.

[3] Anthony, L., Yang, J., & Koedinger, K.R. Evaluation of multimodal input for entering mathematical equations on the computer. In *Proc. CHI 2005*, 1184-1187.

[4] Corbett, A.T. Cognitive computer tutors: Solving the two-sigma problem. In *Proc. User Modeling 2001*, 137–147.

[5] Litman, D.J. & Silliman, S. Itspoke: An intelligent tutoring spoken dialogue system. In *Companion Proc. HLT 2004*, 233–236.

[6] Oviatt, S., Arthur, A. & Cohen, J. Quiet interfaces that help students think. In *Proc. UIST 2006*, 191–200.

[7] Read, J.C, MacFarlane, S.J. & Casey, C. A comparison of two on-line handwriting recognition methods for unconstrained text entry by children. In *Proc British HCI 2003*, 29–32.

[8] Wais, P., Wolin, A. & Alvarado, C. Designing a sketch recognition front-end: user perception of interface elements. In *Proc. SBIM 2007*, 99–106.