Some Useful Design Tactics for Mining ITS Data

Jack Mostow

Project LISTEN, Carnegie Mellon University, RI-NSH 4213, 5000 Forbes Ave, Pittsburgh, USA
Telephone: 412-268-1330 (voice) / 268-6436 (FAX); email: Mostow@cs.cmu.edu
http://www.cs.cmu.edu/~listen

Abstract. Mining data logged by intelligent tutoring systems has the potential to reveal valuable discoveries. What characteristics make such data conducive to mining? What variables are informative to compute? Based on our experience in mining data from Project LISTEN's Reading Tutor, we discuss how to collect machine-analyzable data and formulate it into experimental trials. The resulting concepts and tactics mark out a roadmap for the emerging area of tutorial data mining, and may provide a useful vocabulary and framework for characterizing past, current, and future work in this area.

Keywords: educational data mining, analyzing tutorial dialogue, Project LISTEN, Reading Tutor

1 Introduction

This paper draws on experience from Project LISTEN, an inter-disciplinary project founded over a decade ago to develop, evaluate, and refine an automated Reading Tutor that listens to children read aloud (Mostow & Aist, 2001). The Reading Tutor displays text on the computer screen, uses automated speech recognition to analyze children's oral reading (Beck *et al.*, to appear; Mostow & Aist, 1997; Mostow *et al.*, 1994), and responds with assistance modeled after human reading experts, but tailored to the strengths and limitations of the technology (Mostow & Aist, 1999).

Controlled studies of students' pre- to post-test gains have demonstrated the effectiveness of the Reading Tutor compared to various alternatives, including independent practice (Mostow *et al.*, 2002b; Mostow *et al.*, under revision; Poulsen, 2004), classroom instruction (Mostow *et al.*, 2004, in press), and human tutors (Mostow *et al.*, 2003a). However, such comparisons tell only how well the Reading Tutor works overall. To understand and improve the Reading Tutor's effectiveness, we have performed (and published) many finer-grained analyses. This paper attempts to distill some lessons from that experience for mining data from intelligent tutors in general.

In general, improving an intelligent tutoring system (ITS) requires instrumenting it appropriately. A bottom-up, data flow view of instrumentation identifies which data to record, how to modify the ITS to generate additional data, and which variables to compute from the raw data in order to produce the analyses and visualizations needed to improve educational outcomes. In contrast, a top-down view starts by identifying what information each audience needs in order to achieve this objective. These informational goals then motivate instrumentation decisions. Faculty, students, administrators, technical support staff, content authors, software developers, researchers, and the ITS itself differ in what content and form of information they are interested in and can understand. Faculty may need simple summaries of student usage and progress; administrators need evidence of ITS effectiveness; technical support staff need problem alerts; content authors need usability indicators; developers need accurate bug reports; researchers need detailed examples and informative analyses; and the ITS needs parameters it can use, rules it can interpret, or knowledge it can be modified to exploit.

Such informational goals are typically not clear at the outset, instead emerging from and in turn guiding successively refined instrumentation and analyses. Identifying such informational needs can be laborious. For example, it took a semester course project by five graduate students in human-computer interaction (Alpern et al., 2001) to design, user-test, and refine a prototype of a "teacher tool" to report children's usage and progress in Project LISTEN's Reading Tutor. Moreover, subsequent experience with this tool has led to additional refinement. In short, instrumentation is an iterative attempt to satisfy constraints on what data is feasible to input, what results are possible to compute, and what information is useful to and usable by each audience. This paper focuses on two aspects of this process: what data to collect, and what variables to compute.

2 What makes ITS data mineable?

Learning by doing is essential in education, and takes multiple forms and names, such as homework, practice, problems, exercises, labs, simulations, and explorations. The products of such student work can be useful to analyze. But the *process* of task performance can be even more useful to instrument. For example, model tracing (Anderson *et al.*, 1990) tracks a student's problem-solving steps relative to a model of correct performance, so as to detect mis-steps and provide corrective hints. Knowledge tracing (Corbett & Anderson, 1995) further analyzes such sequences to estimate the student's mastery of different skills, so as to guide tutorial decisions on which skills to teach and which problems to pose.

The types of data available to analyze depend on which activities are instrumented in machine-analyzable form. An ITS can record whatever student activities it involves, such as reading, writing, taking tests, performing various tasks in real or virtual environments, even communicating with peers. But what makes such data mineable? How can the ITS be modified to capture usefully mineable data?

Multiple grain size: To be mineable, the granularity of data should fit its intended analyses. For example, measuring the usage of an ITS requires data about the duration and frequency of student sessions. In contrast, data at the level of individual read words is useful for much finer-grained analyses. Logging data at multiple grain sizes supports multiple sorts of analyses (Mostow *et al.*, 2002c).

Reifying tasks: To be useful, data must be *machine-understandable*. Even a complete record of the student's brain activity would be useless without some way to extract useful information from it. Many tasks are important to student learning but hard for machines to observe or analyze, such as solving a math or physics problem on paper. Reification renders such processes machine-analyzable by reorganizing them to use operations that are easier for computers to instrument. For example, reification may replace handwritten input with typed input, freehand drawing with a limited palette of graphical objects and operations, and free-form responses with menu selections (Self, 1988) or at least canned "sentence openers" (Goodman *et al.*, 2003, pp. 364-365).

Timing: Some student processes are not observable. For example, reading is an important part of most ITSs, but the students' reading comprehension processes are mental, hence not directly observable by the ITS. Nonetheless, an ITS can still capture informative data about students' reading. It can record which text students see, at whose initiative (student or tutor), when, and how many times. It can log how long they take to read each document, page, sentence (by making them click for each new sentence), or even word (by using speech recognition to listen to them read aloud, as in Project LISTEN's Reading Tutor (Beck *et al.*, to appear; Mostow & Aist, 1997). Timing responses to multiple-choice questions can detect likely guessing behavior even when students' answers are correct (Mostow *et al.*, 2002d). Timing how long students spend in different activities can shed light on their motivation and help predict their gains (Mostow *et al.*, 2002a).

Writing: Data on students' writing may include not only their typed input itself, but also response time, duration, and even keystroke-level information. Although the general problem of understanding natural language is AI-complete, the problem of evaluating matches of student responses to expected answers is more constrained, with solutions ranging in depth from spell-checking, to using simple keyword analysis to grade short-answer questions, to using latent semantic analysis to grade students' essay questions (Foltz *et al.*, 1999), to using parsing and domain-specific knowledge to analyze and critique students' self-explanations of proof steps (Aleven *et al.*, 2002).

Peer communication: Communication with peers is important in some ITSs, but difficult to instrument. One possibility is to monitor computer-mediated channels such as email, newsgroups, and on-line chat, or channels built into the tutor itself (Goodman *et al.*, 2003). Analysis of computer-mediated peer communication could range from quantifying its frequency, volume, and patterns of who communicates with whom (Vassileva *et al.*, 1999) to tracking topic content using information retrieval methods.

Student data: Analysis of ITS interactions can benefit from student data that the ITS might not be able to observe, such as gender, age, IQ (Shute *et al.*, 1996), cognitive development (Arroyo *et al.*, 2000), prior declarative knowledge (Corbett *et al.*, 2000), or pretest scores (Mostow *et al.*, 2004). Such data can be supplied separately during the analysis phase (Mostow *et al.*, 2002c), input to the ITS by the student or teacher, or explicitly assessed by the ITS (Arroyo *et al.*, 2000). Even when the extra data would not be feasible for a production version of the ITS to collect, it can still be valuable in analyzing research versions of the ITS.

Manual labeling: Although an ITS can capture too much data to inspect by hand, manual analysis of a strategic sample can be very helpful. The tactic "hand-analyze 10 random examples" of a given case often reveals bugs or

insights. An example of more systematic manual labeling involves an experiment in the Reading Tutor to compare various ways to introduce a new word before a story where it occurs. The outcome variable was the student's ability to read the word in isolation after the end of the story. The Reading Tutor recorded the posttests and sent them to our lab, where they were transcribed by hand for accuracy (Mostow, to appear).

Adding probes: Student comprehension of reading material is essential but not directly observable. One way to gauge students' comprehension of what they read is to insert comprehension questions throughout the text. A simple method to generate such questions is to turn sentences into fill-in-the-blank questions by deleting words. This method is called "cloze" because it tests the reader's ability to deduce semantic closure. Various forms of the cloze method are used widely in elementary reading (McKenna & Layton, 1990; Vacca *et al.*, 1991, pp. 270-272) and language learning (Coniam, 1997). The cloze method has been applied to the Pascal programming language to measure students' comprehension of computer programs (Entin, 1984). In previous work (Mostow *et al.*, to appear; Mostow *et al.*, 2002d), we used multiple-choice cloze questions in the Reading Tutor to estimate children's performance on standard measures of reading comprehension (Woodcock, 1998), achieving correlations of over 0.8 between estimated and actual test scores. An attractive aspect of this simple method for assessing comprehension is its ability to generate, administer, and score comprehension questions automatically for any given text.

Randomizing tutorial decisions: Allocating credit and blame over the long sequence of ITS and student decisions leading to a given educational outcome is hard. One approach to making this problem somewhat more tractable is to embed randomized experiments in the ITS. For example, one such experiment (Aist, 2001) tested the effectiveness of briefly explaining new words just before students saw them in the Reading Tutor, compared to just seeing them in context without being interrupted. The Reading Tutor randomly chose to explain some words but not others. The outcome variable was performance on multiple-choice questions about explained and unexplained words, administered the next day the student used the Reading Tutor. Analysis of over 3,000 randomized trials showed that the inserted explanations helped – but not for all words and all students. They helped on rare, single-sense words, and they helped third graders more than second graders. Such experiments test whether a tutorial action is worth the time it takes to perform it – and for which students. By clarifying the stage at which a given type of assistance helps, such experiments can guide tutorial sequencing. Project LISTEN has published many such experiments (Aist & Mostow, 1998, 1999, 2000; Beck *et al.*, 2004a; Beck *et al.*, 2003; Beck *et al.*, 2004b; Heiner *et al.*, 2004b; Mostow, to appear; Mostow & Aist, 2001; Mostow *et al.*, 2001; Mostow *et al.*, 2003b; Mostow *et al.*, to appear; Mostow *et al.*, 2004; Mostow *et al.*, 2003c; Mostow *et al.*, 2002d).

3 Formulating ITS data as experimental trials

Instrumenting an ITS produces a rich stream of data, but it merely encodes a series of events. We need to translate such data into variables to visualize and analyze. To analyze the complex effects on eventual educational outcomes of extended, rich interaction with an ITS, it helps to decompose that interaction into a series of experimental trials defined by local decisions. However, some kinds of trials are more conducive than others to drawing well-supported causal inferences.

Decision: Each trial starts with a decision that occurs while (or before) a student uses the ITS and affects the ensuing tutorial interaction. A hardwired decision that always chooses the same option is hard to analyze because it provides no basis for comparison. Decisions made by the ITS are easier to analyze than decisions made by the student, because the effects of student-influenced decisions are hard to tease apart from other student effects. Randomized decisions are the easiest to analyze and provide the strongest evidence to support causal inferences.

Context: Each trial occurs in a context, such as a particular student in a particular class encountering a particular word in a particular story on a particular computer at a particular time on a particular date. The characterization of the context as a set of features serves as a basis for aggregating or disaggregating trials to relate context to outcome. For example, Heiner *et al.* (2004b) grouped trials by word difficulty and student proficiency. It is important to log (or be able to reconstruct) the context in which the trial occurs, and the set of options from which the random selection is made, which might depend on the context. For example, the Reading Tutor randomizes its choice of help to give on a word, but a rhyming hint is not an option for the word "orange."

Outcome: Each trial ends with an outcome. If the experimental outcome is formulated in advance, the ITS may be able to explicitly log each outcome as soon as it occurs. Logging each trial of an experiment as a single record in a table specific to that experiment, with fields encoding the experiment's context, decision, and outcome, simplifies analysis but is not always feasible. For example, often we do not define the outcome variable until after the fact.

Ideally trials are independent, but complications can occur. One such complication is "masking," in which one trial affects the outcome of another (Mostow & Aist, 2001). For example, suppose the randomized decision is what type of help to give on a word, and the outcome of the trial is the student's performance when the student encounters the word again in a later sentence. An instance of masking arises if the tutor gives help again just before the second encounter, thereby affecting the student's performance. Another complication is confounding the experimental manipulation with other influences on trial outcomes. For example, the student's performance may depend on how soon the trial ends, which is influenced in turn by how often the word occurs in English. If some types of word help (e.g. rhyming hints) tend to apply to commoner words than others (e.g. segmenting into syllables), then trial outcomes may confound help type with word frequency (Mostow *et al.*, 2004). Such complications are not necessarily insurmountable, but may need careful statistical treatment to control for them.

Several emerging ideas can be viewed as transforming ITS data into a set of trials. These ideas include segmenting tutorial interactions into episodes; abstracting data streams as the subset of interactions, or "slice," relevant to a particular target; formulating their outcomes in terms of available data; and aggregating in informative ways. We now discuss how these transformations can be applied and combined. Complementary ideas discussed above facilitate such transformations by modifying the instrumented process, for example by randomizing tutorial decisions, reifying student operations to make them machine-observable, or adding explicit probes to make more data available.

Segmentation: One way to simplify a complex stream of interaction is to parse it into shorter episodes that can be analyzed individually. For example, Beck *et al.* (2000) segmented tutorial dialogue at each student response to a tutorial action. As outcome variables, they measured the time it took the student to respond, and whether the response was correct. To characterize the context, they used 48 variables describing various features of the student, the problem, and recent instruction, such as the amount of help provided. Other analyses are also examples of segmentation (Aist & Mostow, 1999; Mayo & Mitrovic, 2001; Murray & Arroyo, 2002). Segmentation preserves the "width" of the interaction stream but cuts it into segments of shorter duration.

Slicing: A powerful way to abstract a stream of interactions is inspired by "program slicing" (Weiser, 1984), a method to simplify analysis of a computer program by considering only the parts that affect particular variables. In mining a stream of tutorial interactions, the idea of slicing is to focus only on interactions relevant to a given educational target, such as an individual rule or word. Unlike segmentation, slicing preserves the length of the interaction stream but factors it into narrow strands, one for each target. For example, knowledge tracing (Corbett & Anderson, 1995) "slices" students' problem-solving into opportunities to apply different rules, wich one slice for each rule. Plotting students' performance at successive opportunities to apply a given rule reveals a characteristic power-law learning curve. Aist & Mostow (1998) factored student-tutor dialogue on a given sentence into a separate slice for each word, consisting of the student's attempts to read the word, and the Reading Tutor's assistance on it. Jia (2002) plotted learning curves by tracking students' performance on a given word over successive encounters in different sentences. Performance measures included the probability of clicking on the word for help, and the amount of hesitation before speaking the word.

Aist & Mostow (1998) illustrates the combination of slicing and segmentation. This work segmented each word slice into episodes within an individual encounter of a sentence. Each episode started with one encounter of the word, and ended at the next encounter. Variables included the student's performance on the word at the start and end of the episode (accepted, omitted, or misread, according to the speech recognizer), and whether the Reading Tutor gave any assistance on the word during the episode. They characterized the local effects of such assistance by analyzing 227,693 episodes, of which 62,645 included assistance on the word, and 165,048 did not. "For example, what happened after a word was misread? The next attempt was correct 41% of the time with a word intervention – but only 15% without." The Reading Tutor's decisions on whether to intervene were not random, as they would need to be to attribute causality unambiguously. Later experiments (Beck *et al.*, 2004b; Mostow, to appear) remedied this limitation.

Outcome formulation: Another important step in mining tutor data is to define local outcomes. The embedded vocabulary experiment described earlier (Aist, 2001) added a next-day test to measure outcomes. However, tests take time and can interrupt learning. How can we define outcomes based on available data?

For example, one randomized experiment (Aist & Mostow, 1999) embedded in the Reading Tutor measured the local effects of "backchanneling" (making an encouraging sound such as "mm-hmm") when the student hesitated. The outcome was defined as whether the student responded within the next four seconds.

Another randomized intervention (Beck *et al.*, 2004a) tried to stimulate children's comprehension by asking occasional "wh-" questions such as "Who is the story talking about?" Rather than introduce additional probes to

measure the effects of this intervention, Beck *et al.* exploited the cloze questions already being asked. Specifically, they defined the outcome of each intervention as the reader's performance on the next cloze question. This formulation enabled them not only to show that the intervention was effective, but to analyze how performance varied with the recency of the intervention.

We already discussed how Aist & Mostow (1998) segmented word slices at successive encounters in order to measure the immediate *assistive* effects of Reading Tutor assistance. Heiner *et al.* (2004b) and Mostow *et al.* (2004) extended this approach to measure *learning* effects by defining the outcome of tutorial help on a word in terms of the student's performance on that word in a later sentence. This fine-grained (and noisy) measure of the effect of a single hint on a single word is a step toward assessing effects of tutorial help *policy* on students' overall *gains* in reading skills.

Aggregation: Although individual observations are sometimes informative, their value often comes from aggregating them in informative ways, especially when individual observations vary stochastically. Well-chosen variables can expose systematic structure in such variations so as to illuminate the diverse social, academic, technical, and motivational processes underlying them – or just to address the important educational research goal of "description – what is happening?" (Shavelson & Towne, 2002, p. 99).

To take a simple example, one important variable in educational (or any) software is its reliability. To track the reliability of the Reading Tutor in use at schools, we computed the percentage of student sessions that ended in crashes, and how this percentage changed when we installed updates. Characterizing where crashes occurred helped identify the most frequent bugs and the flakiest computers.

Another important variable is usage, which may vary dramatically from one day, classroom, or student to another. We found that factoring Reading Tutor usage into session frequency and session duration was informative (Mostow & Beck, to appear). The number of days per week that students used the Reading Tutor turned out to vary primarily *between* classrooms, reflecting each teacher's role as gatekeeper for software usage. In contrast, session duration turned out to vary primarily *within* classroom, reflecting students' individual decisions of when to get off. By factoring session frequency into how often *anyone* used the Reading Tutor in a classroom, and what fraction of the class used it on those days, we can presumably distinguish the relative frequency of teachers halting usage on days when problems arise, versus not putting students on in the first place.

Although overall usage is informative, so is more detailed information on where students spend their time. Mostow *et al.* (2002a) identified different aggregations to characterize students' time allocation among different activities on the Reading Tutor, such as stories chosen by the student vs. by the Reading Tutor, or reading vs. writing. Time allocation correlated significantly with gains in test scores, even controlling for total time on tutor.

Given a strong model of the student, knowledge tracing estimates the probability that the student has learned a given rule (Corbett & Anderson, 1995). Here aggregation consists of Bayesian updating based on the outcomes of successive opportunities to apply the rule. This variable guides the choice of what problem to present next. That is, the audience for the variable is the software itself.

Conclusion

We have discussed how to get mineable ITS data by making it machine-analyzable and randomizing decisions. We have described segmentation, slicing, formulation, and aggregation tactics for treating data as experimental trials consisting of decisions, contexts, and outcomes. This approach transforms year-long, fine-grained streams of daily, mixed-initiative tutorial interactions with several hundred students into hundreds of thousands of within-subject experimental trials. The resulting "big data" offers the statistical power needed to discover which tutorial actions help which students in which cases.

As the above examples illustrate, outcome variables are often deceptively simple once they are defined. The research challenge is how to formulate them from initially murky questions. Another problem is to compute the variables. This computation can be surprisingly hard to define and implement correctly. One cause is rich input data, which can make it difficult to identify and correctly log every possible case in which the variable is to be computed. Another cause is the need to define the variables after the data is collected, rendering them tricky to operationalize in terms of the particular data representation adopted.

The ability to define variables after the fact is critical to support exploratory and iterative data analysis. Replacing the Reading Tutor's sequential event log file representation with a carefully designed database (Mostow *et al.*, 2002c) has dramatically improved the ease, power, and quality of such after-the-fact analyses. We have focused here on how to make ITS data mineable. A companion paper (Heiner *et al.*, 2004a) discusses how we carry out the actual mining.

Acknowledgements

This work was supported in part by the National Science Foundation under ITR/IERI Grant No. REC-0326153. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. Thank you to the educators and students who generated our data, and current and past members of Project LISTEN who contributed to this work.

References (also see www.cs.cmu.edu/~listen)

Aist, G. (2001). Towards automatic glossarization: Automatically constructing and administering vocabulary assistance factoids and multiple-choice assessment. *International Journal of Artificial Intelligence in Education, 12*, 212-231.

Aist, G., & Mostow, J. (1998, March). Estimating the effectiveness of conversational behaviors in a reading tutor that listens. *Working Notes of the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, Stanford, CA.

---. (1999, September). Measuring the Effects of Backchanneling in Computerized Oral Reading Tutoring. *Proceedings of the ESCA Workshop on Prosody and Dialog*, Eindhoven, Netherlands.

---. (2000, June). Using Automated Within-Subject Invisible Experiments to Test the Effectiveness of Automated Vocabulary Assistance. *Proceedings of ITS'2000 Workshop on Applying Machine Learning to ITS Design/Construction*, Montreal, Canada, 4-8.

Aleven, V., Popescu, O., & Koedinger, K. (2002). Pilot-testing a tutorial dialogue system that supports self-explanation. *6th International Conference on Intelligent Tutoring Systems*, Biarritz, France, 344-354.

Alpern, M., Minardo, K., O'Toole, M., Quinn, A., & Ritzie, S. (2001). *Project LISTEN: Design Recommendations and Teacher Tool Prototype* (Unpublished Group Project for Masters' Lab in Human-Computer Interaction). Pittsburgh: Carnegie Mellon University.

Anderson, J., Boyle, C. F., Corbett, A., & Lewis, M. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence, 42*, 7-49.

Arroyo, I., Beck, J. E., Woolf, B. P., Beal, C. R., & Schultz, K. (2000). Macroadapting AnimalWatch to gender and cognitive differences with respect to hint interactivity and symbolism. *5th International Conference on Intelligent Tutoring Systems (ITS2000)*, Montreal, Canada, 574-583.

Beck, J. E., Jia, P., & Mostow, J. (to appear). Automatically assessing oral reading fluency in a computer tutor that listens. *Technology, Instruction, Cognition and Learning, 2*.

Beck, J. E., Mostow, J., & Bey, J. (2004a, September 1-3). Can automated questions scaffold children's reading comprehension? *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, Maceio, Brazil.

Beck, J. E., Mostow, J., Cuneo, A., & Bey, J. (2003, July 20-24). Can automated questioning help children's reading comprehension? *Proceedings of the Tenth International Conference on Artificial Intelligence in Education (AIED2003)*, Sydney, Australia, 380-382.

Beck, J. E., Sison, J., & Mostow, J. (2004b, June 27-30). Using automated speech recognition to measure scaffolding and learning effects of word identification interventions in a computer tutor that listens. *Eleventh Annual Meeting of the Society for the Scientific Study of Reading*, Amsterdam, The Netherlands.

Beck, J. E., Woolf, B. P., & Beal, C. R. (2000). ADVISOR: A machine learning architecture for intelligent tutor construction. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, Texas, 552-557.

Coniam, D. (1997). A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests. *CALICO Journal, 14*(2-4), 15-33.

Corbett, A., & Anderson, J. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction, 4*, 253-278.

Corbett, A. T., McLaughlin, M. S., & Scarpinatto, K. C. (2000). Modeling student knowledge: Cognitive tutors in high school and college. *User modeling and user-adapted interaction, 10*, 81-108.

Entin, E. B. (1984). Using the cloze procedure to assess program reading comprehension. *SIGCSE Bulletin, 16*(1), 448.

Foltz, P. W., Laham, D., & Landauer, T. K. (1999). Automated Essay Scoring: Applications to Educational Technology. *Proceedings of EdMedia '99*.

Goodman, B., Hitzeman, J., Linton, F., & Ross, H. (2003, June 22-26). Towards Intelligent Agents for Collaborative Learning: Recognizing the Roles of Dialogue Participants. *9th International Conference on User Modeling*, Johnstown, PA, USA, 363-367.

Heiner, C., Beck, J., & Mostow, J. (2004a, August 30). Lessons on using ITS data to answer educational research questions. *Proceedings of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, Maceio, Brazil.

Heiner, C., Beck, J. E., & Mostow, J. (2004b, June 17-19). Improving the Help Selection Policy in a Reading Tutor that Listens. *Proceedings of the InSTIL/ICALL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems*, Venice, Italy.

Jia, P. (2002). *Mining computer tutor-student interaction data to assess students' reading and predict future behavior.* Unpublished Master's Project for Knowledge Discovery in Databases, Center for Automated Learning & Discovery, Carnegie Mellon University, Pittsburgh, PA.

Mayo, M., & Mitrovic, A. (2001). Optimising ITS Behaviour with Bayesian Networks and Decision Theory. *International Journal of Artificial Intelligence in Education, 12*, 124-153.

McKenna, M. C., & Layton, K. (1990). Concurrent Validity of Cloze as a Measure of Intersentential Comprehension. *Journal of Educational Psychology, 82*(2), 372-377.

Mostow, J. (to appear). Evaluation purposes, excuses, and methods: Experience from a Reading Tutor that listens. In C. K. Kinzer & L. Verhoeven (Eds.), *Interactive Literacy Education: Facilitating Literacy Environments Through Technology*. Mahway, NJ: Erlbaum Publishers.

Mostow, J., & Aist, G. (1997, July). The Sounds of Silence: Towards Automated Evaluation of Student Learning in a Reading Tutor that Listens. *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, Providence, RI, 355-361.

---. (1999). Giving help and praise in a reading tutor with imperfect listening -- because automated speech recognition means never being able to say you're certain. *CALICO Journal, 16*(3), 407-424.

---. (2001). Evaluating tutors that listen: An overview of Project LISTEN. In K. Forbus & P. Feltovich (Eds.), *Smart Machines in Education* (pp. 169-234). Menlo Park, CA: MIT/AAAI Press.

Mostow, J., Aist, G., Beck, J., Chalasani, R., Cuneo, A., Jia, P., & Kadaru, K. (2002a, June 5-7). A La Recherche du Temps Perdu, or As Time Goes By: Where does the time go in a Reading Tutor that listens? *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems (ITS'2002)*, Biarritz, France, 320-329.

Mostow, J., Aist, G., Bey, J., Burkhead, P., Cuneo, A., Junker, B., Rossbach, S., Tobin, B., Valeri, J., & Wilson, S. (2002b, June 27-30). Independent practice versus computer-guided oral reading: Equal-time comparison of sustained silent reading to an automated reading tutor that listens. *Ninth Annual Meeting of the Society for the Scientific Study of Reading*, Chicago, Illinois.

Mostow, J., Aist, G., Bey, J., Burkhead, P., Cuneo, A., Rossbach, S., Tobin, B., Valeri, J., & Wilson, S. (2001). A hands-on demonstration of Project LISTEN's Reading Tutor and its embedded experiments (refereed demo). *Language Technologies 2001: The Second Meeting of the North American Chapter of the Association for Computational Linguistics.*, Pittsburgh, PA.

Mostow, J., Aist, G., Burkhead, P., Corbett, A., Cuneo, A., Eitelman, S., Huang, C., Junker, B., Sklar, M. B., & Tobin, B. (2003a). Evaluation of an automated Reading Tutor that listens: Comparison to human tutoring and classroom instruction. *Journal of Educational Computing Research, 29*(1), 61-117.

Mostow, J., Aist, G., Huang, C., Junker, B., Kennedy, R., Lan, H., Latimer, D., O'Connor, R., Tassone, R., Tobin, B., & Wierman, A. (2004, in press). 4-Month Evaluation of a Learner-controlled Reading Tutor

that Listens. In V. M. Holland & F. N. Fisher (Eds.), *Speech Technology for Language Learning*. Lisse, The Netherlands: Swets & Zeitlinger Publishers.

Mostow, J., & Beck, J. (to appear). When the Rubber Meets the Road: Lessons from the In-School Adventures of an Automated Reading Tutor that Listens. In B. Schneider (Ed.), *Conceptualizing Scale-Up: Multidisciplinary Perspectives*.

Mostow, J., Beck, J., Bey, J., Cuneo, A., Sison, J., & Tobin, B. (2003b, June 12-15). An Embedded Experiment to Evaluate the Effectiveness of Vocabulary Previews in an Automated Reading Tutor. *Tenth Annual Meeting of the Society for Scientific Studies of Reading*, Boulder, CO.

Mostow, J., Beck, J., Bey, J., Cuneo, A., Sison, J., Tobin, B., & Valeri, J. (to appear). Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions. *Technology, Instruction, Cognition and Learning, 2*.

Mostow, J., Beck, J., Chalasani, R., Cuneo, A., & Jia, P. (2002c, October 14-16). Viewing and Analyzing Multimodal Human-computer Tutorial Dialogue: A Database Approach. *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI 2002)*, Pittsburgh, PA, 129-134.

Mostow, J., Beck, J., & others. (under revision). Computer-Guided Oral Reading versus Independent Practice: Comparison of Sustained Silent Reading to an Automated Reading Tutor that Listens. *Journal of Educational Psychology*.

Mostow, J., Beck, J. E., & Heiner, C. (2004, June 27-30). Which Help Helps? Effects of Various Types of Help on Word Learning in an Automated Reading Tutor that Listens. *Eleventh Annual Meeting of the Society for the Scientific Study of Reading*, Amsterdam, The Netherlands.

Mostow, J., Beck, J. E., & Valeri, J. (2003c, June 22). Can Automated Emotional Scaffolding Affect Student Persistence? A Baseline Experiment. *Proceedings of the Workshop on "Assessing and Adapting to User Attitudes and Affect: Why, When and How?" at the 9th International Conference on User Modeling (UM'03)*, Johnstown, PA, 61-64.

Mostow, J., Roth, S. F., Hauptmann, A. G., & Kane, M. (1994, August). A prototype reading coach that listens [AAAI-94 Outstanding Paper Award]. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, 785-792.

Mostow, J., Tobin, B., & Cuneo, A. (2002d, June 3). Automated comprehension assessment in a reading tutor. *Proceedings of the ITS 2002 Workshop on Creating Valid Diagnostic Assessments*, San Sebastian, Spain, 52-63.

Murray, T., & Arroyo, I. (2002). Towards Measuring and Maintaining the Zone of Proximal Development in Adaptive Instructional Systems. *Sixth International Conference on Intelligent Tutoring Systems*.

Poulsen, R. (2004). *Tutoring Bilingual Students With an Automated Reading Tutor That Listens: Results of a Two-Month Pilot Study*. Unpublished Masters Thesis, DePaul University, Chicago, IL.

Self, J. (1988). Bypassing the intractable problem of student modelling. *Intelligent Tutoring Systems*, 18-24.

Shavelson, R. J., & Towne, L. (Eds.). (2002). *Scientific Research in Education*. National Research Council, Washington, D.C.: National Academy Press.

Shute, V., Gawlick-Grendell, L. A., Young, R. K., & Burnham, C. A. (1996). An Experiential System for Learning Probability: Stat Lady Description and Evaluation. *Instructional Science, 24*(1), 25-46.

Vacca, J. A. L., Vacca, R. T., & Gove, M. K. (1991). *Reading and Learning to Read* (2nd ed.). New York: Harper Collins.

Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C., & Grant, S. (1999). A Multi-agent Approach to the Design of Peer-help Environments. *International Conference on Artificial Intelligence in Education*, Le Mans, France, 38-45.

Weiser, M. (1984). Program slicing. *IEEE Transactions on Software Engineering, SE-10*(4), 352-357.

Woodcock, R. W. (1998). *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU)*. Circle Pines, Minnesota: American Guidance Service.