

Evaluating Dialogue Schemata with the Wizard of Oz Computer-Assisted Algebra Tutor

Jung Hee Kim¹ and Michael Glass²

¹Dept. Computer Science
North Carolina A&T State Univ.
Greensboro, NC 27411
jungkim@ncat.edu

²Dept. Math & CS
Valparaiso University
Valparaiso, IN 46383
michael.glass@valpo.edu

Abstract. The Wooz tutor of the North Carolina A&T algebra tutorial dialogue project is a computer program that mediates keyboard-to-keyboard tutoring of algebra problems, with the feature that it can suggest to the tutor canned structures of tutoring goals and canned sentences to insert into the tutoring dialogue. It is designed to facilitate and record a style of tutoring where the tutor and student collaboratively construct an answer in the form of an equation, a style often attested in natural tutoring of algebra. The algebra tutoring dialogue project collects and analyzes these dialogues with the aim of describing tutoring strategies and language with enough rigor that they may be evaluated and incorporated in machine tutoring. By plugging our analyzed dialogues into the computer-suggested tutoring component of the Wooz tutor we can evaluate the fitness of our dialogue analysis.

Introduction

Tutorial dialogues are often structurally analyzed for purposes of constructing tutoring systems and understanding the tutorial process. However there are not many ways for validating the analysis of a dialogue, either for verifying that the analysis matches the structure that a human would use, or for verifying that the analysis is efficacious. In the algebra tutorial dialogue project at North Carolina A&T State University we use a machine-assisted human tutor to evaluate our analysis of elementary college algebra tutoring dialogues. The project has collected transcripts of human tutoring using an interface that provides an enhanced chat-window environment for keyboard to keyboard tutoring of algebra problems [1]. These transcripts of tutorial dialogue are annotated based on the tutor's intentions and language. From these annotations we have created structured tutoring scenarios which we import into an enhanced computer-mediated tutoring interface: the Wooz tutor. In subsequent tutoring sessions, the tutor has the option of selecting language from the canned scenario, edited or ignored as the tutor sees fit, for tutoring some of the problems. The resulting transcripts are then analyzed to evaluate the fitness of our scenarios for tutoring, based on measures

such as pre- and post-test scores and the number of times that the tutor deviated from the script.

The algebra tutorial dialogue project captures tutoring of high school and college algebra problems with several goals in mind: 1) cataloging descriptions of tutoring behavior from both tutor and student, using where possible enough rigor that they might be useful for dialogue-based computerized tutoring, 2) evaluating the effectiveness of various tutoring behaviors as they are originally observed, and 3) describing these computer-mediated human educational dialogue interactions in general, as being of use to the educational dialogue and cognitive psychology communities. The Wooz tutor is a useful tool for partially evaluating our success in these endeavors.

Environment and Procedure

Computer-Mediated Tutoring Environment

The tutoring dialogues we captured consist of a tutor and a student working problems collaboratively. The dialogue model is of a tutor and student conversing, with both the problem statement and the equation being worked on being visible to both parties. We analyze typed communication because first, this is the mode most tractable for computerization and second, we can capture all the communication between student and tutor, there are no gaze, gesture, prosodic features, and so on to capture and annotate. Thus the computer-supported tutoring environment affords the following:

1. The statement of the problem currently being worked on is always on display in a dedicated window.
2. The equations being developed while solving the problem are displayed in a dedicated window, there is a tool bar for equation editing.
3. Typed tutorial dialogue appears, interleaved, in a chat-window.

Additionally there is some status information, e.g. which party has the current turn, and the tutor has some special controls, such as a menu of problem statements to pick from. One feature of this software environment is that the equation editor toolbar is customized for each problem, so extraneous controls not needed for solving the problem under discussion are not displayed.

A phenomenon annotated in other transcripts of algebra tutoring is deixis [2, 3], in particular pointing at equations or parts of equations. Although our interface has the capability to display and edit several equations at the same time in its equation area, it has no good referring mechanism for the participants to use. So far, we have not noticed this to be an issue in the dialogues we have collected.

Regarding our experience with the program, we have collected transcripts from 50+ students to date, each comprised of about one hour of tutoring, for a total of approximately 3000 turns and 300 problems. Students and tutors receive brief instruction before use, they have had little difficulty learning to use the application, including constructing equations.

Dialogue Collection

These problem-oriented tutoring dialogues are similar in form to those studied extensively by the ITS community, e.g. [3, 4, 5], whose salient features were summarized by [6]. An extract from a typical dialogue is illustrated in Figure 1.

Problems solved during these tutoring sessions include both symbolic manipulation problems and word problems, viz:

1. Please factor $3x^2 - 11x + 6$.
2. Bob drove "m" miles from Denver to Fargo. Normally this trip takes "n" hours, but on Tuesday there was good weather and he saved 2 hours. Write an equation for his driving speed "s".

Students solve an average of between 5 and 6 problems in an hour session.

One feature of our tutoring data collection protocol is that the student's performance on the pre-test determines which categories of problems will be tutored. The tutor gives priority to problems similar to the ones the student answered incorrectly on the pre-test, but did not leave totally blank. These are the areas where we judge that the student is likely most ready to benefit from tutoring. The post-test then covers only the problem areas that were tutored, so that any learning gains we measure are specifically measuring learning for the particular tutoring that occurred. For data analysis purposes the students are coded with an achievement level, on a scale of 1 (lowest) to 5. The achievement judgment is derived from the teacher of the student's algebra class, based on previous academic performance in the class.

The NC A&T dialogue project has accumulated 51 one-hour transcripts in this way. The students are all recruited from the first year basic algebra classes. About 24 of the transcripts were taught by an expert tutor, a professor of mathematics with extensive experience tutoring algebra, 16 are divided approximately evenly between experienced tutors, two people with extensive experience but no formal mathematics education background, and 11 were taught by a novice tutor, an upper-level mathematics student.

Students exhibit a learning gain of 0.35 across all tutoring sessions, calculated as:

$$(\text{posttest} - \text{pretest}) / (1 - \text{pretest})$$

where the test scores range from 0.0 to 1.0. The expert tutor's sessions exhibit a learning gain of 0.41, the experienced tutors' learning gain is 0.33, and the novice tutor's learning gain is 0.24. These data show that the dialogues do, in fact, record learning events. Furthermore it also indicates that even though novice tutors can be successful, additional tutoring experience seems to improve tutoring outcomes.

Dialogue Analysis

Figure 1 shows an extract from a relatively short dialogue where the student solved one multiplication problem. (In printed transcripts, the evolving equation in the equation window is interpolated into the dialogue every time the equation changes.) Even though the student performed perfectly in solving the problem, it illustrates the most prominent tutoring strategy used by our tutors: ensuring that the student can state the type of problem (multiplying polynomials in this case) and a technique to solve it (a mnemonic device in this case) before proceeding with a solution. Rarely do the tutors skip these steps. This tactic can also be seen in the transcripts of [2]. This tactic alone

is often enough to get the student to solve the problem, as illustrated, even when the student failed to solve similar problems on the pre-test. Getting the student to explicitly state the problem and method is consistent with the view that learning mathematics often invokes metacognitive processes [7].

	<i>Problem: $(3x-2)(x+5)$</i>
<i>Tu:</i>	<i>What operation do you need to apply here?</i>
<i>St:</i>	<i>Factoring.</i>
<i>Tu:</i>	<i>No, Not factoring.</i>
<i>Tu:</i>	<i>The expression is already factored.</i>
<i>Tu:</i>	<i>What is the opposite of factoring?</i>
<i>St:</i>	<i>Distributing.</i>
<i>Tu:</i>	<i>You can say distributing.</i>
<i>Tu:</i>	<i>But we are multiplying here.</i>
<i>Tu:</i>	<i>Now, what process do we use when we are multiplying two binomials.</i>
<i>St:</i>	<i>Foil.</i>
<i>Tu:</i>	<i>Excellent, You use FOIL.</i>
<i>Tu:</i>	<i>What does FOIL stand for?</i>
<i>St:</i>	<i>First Outer Inner Last.</i>
<i>Tu:</i>	<i>Good, Now go ahead and use FOIL to multiply the two binomials together.</i>
<i>St:</i>	$3x^2 + 15x - 3x^2 - 10$
<i>St:</i>	$3x^2 + 15x - 2x - 10$
<i>St:</i>	$3x^2 + 13x - 10$
<i>Tu:</i>	<i>Excellent.</i>

Fig. 1. Typical Tutorial Dialogue

We annotate our transcripts according to a hierarchy of the tutor's dialogue and tutorial goals. For purposes of constructing a mechanical tutor that models human dialogue behaviors, this style of rigorously annotated human dialogues has provided the data which inform several intelligent tutoring system projects, e.g. the CIRCSIM-Tutor baroreceptor reflex tutor [8, 9, 10], the Auto-Tutor computer literacy tutor [11], and the Ms. Lindquist algebra word problem tutor [12]. Our annotation scheme is similar to the CIRCSIM-Tutor scheme [10].

The model underlying this style of markup is that tutoring consists of a set of verbal gambits, whereby: 1) a gambit potentially spans multiple turns of dialogue, 2) each gambit addresses a particular tutorial goal, and 3) goals and subgoals are hierarchically organized, meaning there are gambits within gambits. We call a sequence of goals a schema, each subtree can also be a schema. This view of dialogue is motivated by current computer models of dialogue planning. Our schemata do not, in themselves, attempt to describe domain or pedagogical reasoning. For example we have a tutorial goal called obtain-factors which occurs as part of larger pedagogical gambits, but we do not record how the tutor finds factors. The result of this annotation process is that we identify tutoring schemata, common patterns of dialogue goals that the tu-

tors employ, without identifying the domain or pedagogical reasoning that may explain those schemata. In consequence, many of our schemata are quite problem-specific. The fact that this assemblage of goals and schemata is imputed from text by the researchers, and not derived in a principled way, makes evaluating them more important.

The Atlas-Andes tutor [13] guides the student through problem-solving tasks where the main tutorial mode consists of model tracing guided by physics reasoning. Our markup would be unable to capture and our Wooz tutor would be unable to evaluate such dialogues. However Atlas-Andes also includes, as an adjunct method of tutoring, dialogue schemata similar to our own called Knowledge Construction Dialogues. These dialogues would seem to be amenable to Wooz tutor evaluation.

A reason this style of analysis is possible is that our tutors do not teach much algebraic reasoning. Instead they emphasize applying problem-solving methods previously learned in class, along with teaching the metacognitive skills to know how to apply these methods.

Figure 2 shows the evolving trace of tutorial goals from one of our typical dialogues, as affected by student errors and retries. The three prominent goals discussed above are labeled identify-operation, identify-approach and solve-problem in this annotation scheme.

We abstract general schemata from many instances of tutoring such as Figure 2. The quite general-purpose schema of identify-problem, identify-approach, and solve-problem usually involves problem-specific sub-schemata. For example, to satisfy

<i>Goal Hierarchy</i>	<i>Comment</i>
IDENTIFY OPERATION	Operation is factoring
TUTOR OPERATION	Elicit the operation from the student, 1st attempt
TUTOR EXPRESSION	Student did not recognize it is an expression, this is a fixup
TUTOR OPERATION	2nd attempt
IDENTIFY APPROACH	Solution approach is trial-and-error factoring
TUTOR TRIAL-AND-ERROR	Elicit from or inform the student to factor by trial-and-error
SOLVE PROBLEM	
MAKE BINOMIALS	Ask the student to produce both binomial terms all at once.
OBTAIN FACTORS	Student failed, obtain terms one at a time, 1st attempt
OBTAIN FIRST FACTOR	
OBTAIN SECOND FACTOR	
CONFIRM FACTORING	
CHECK-VIA-FOIL	Confirm by multiplying back The student made an error.
OBTAIN FACTORS	2nd attempt
OBTAIN SECOND FACTOR	Correct the one term the student got wrong
(CONFIRM FACTORING)	Student did this without the tutor prompting

Fig. 2. Tutorial Goals in a Typical Dialogue

<p>IDENTIFY THE PROBLEM <i>What are we doing with this expression?</i> <i>We are factoring.</i></p> <p>INTRODUCE THE TRAIL-ERROR APPROACH. <i>What method can we use to factor this?</i> <i>Let's use the trial and error method.</i></p> <p>SOLVE PROBLEM SOLVE VIA MAKING BINOMIALS MAKE BINOMIALS <i>So you set up your 2 sets of parentheses and fill them in with factors of the first term and the last.</i> <i>This trinomial will factor into 2 binomials.</i></p> <p>CONFIRM FACTORING <i>Now how do we check our factors.</i> <i>Now use FOIL to check this</i></p> <p>MAKE FIRST TERM <i>What are the two factors of $2x$ squared?</i> <i>Ok, x squared factors into x and x, right?</i> etc.</p> <p>SOLVE VIA CORRECTING FACTORS etc.</p>

Fig. 3. Extract From Sentences For Each Goal as Presented to the Wooz Tutor

solve-problem in the trinomial factoring domain, we have a schema of make-binomials and confirm-factoring. If that fails, solve-problem might be satisfied by an alternate schema of obtain-factors, (which itself is composed of the goals obtain-first-factor and obtain-second-factor) followed by confirm-factoring.

Wooz Tutor

The tutorial schemata are then evaluated by using them in tutorial dialogues with students, via the Wooz Tutor¹. Running in Wooz Tutor mode, the computer-mediated communication software presents the human tutor with an additional menu of tutoring goals and a set of associated sentences for each goal. The tutor can optionally select and edit a sentence, then send it to the dialogue.

Note that since the Wooz tutor is a superset of our normal computer-mediated tutoring interface, it is possible to conduct tutoring dialogues where some of the problems are mechanically assisted and some are produced entirely from the human tutor.

¹ Wooz comes from Wizard of Oz. The public face of the tutor, including its language and goals, comes from the machine, while there is a human intelligence pulling the strings. The name is a bit of a misnomer, as we do not try to fool the students.

Following the identification of schemata, we collect examples of language used for each goal. The sets of goals and associated sentences are then collected together, one set for each problem, illustrated in Figure 3. Some of the sentences are simple templates where the variable slots can be filled in with the student's name or problem-specific information. On the Wooz tutor interface, the goals hierarchy appears as an expandable tree of nodes, where expanding a leaf node exposes the sentences that can be picked. Mouse-over of a goal node shows the first sentence that can be used for expressing that goal, enabling the tutor to peer inside the tree more readily. Figure 4 shows the Wooz tutor as the tutor sees it.

From the transcripts we can then evaluate how much of the dialogue came from the canned sentences, edited sentences, or entirely new sentences. We can also tell when the tutor left the goal script. This gives us an indication of the effectiveness and completeness of our isolated tutoring schemata and language.

The intelligence for understanding and evaluating student input, and deciding when and where to switch tutorial goals, still resides in the human tutor. The schemata we isolate and test with this method do not specify all that is needed for mechanizing the tutoring process with an ITS. However the tradeoff for leaving the decisions in the hands of a human tutor is that the simple evaluation of schemata is quite cheap.

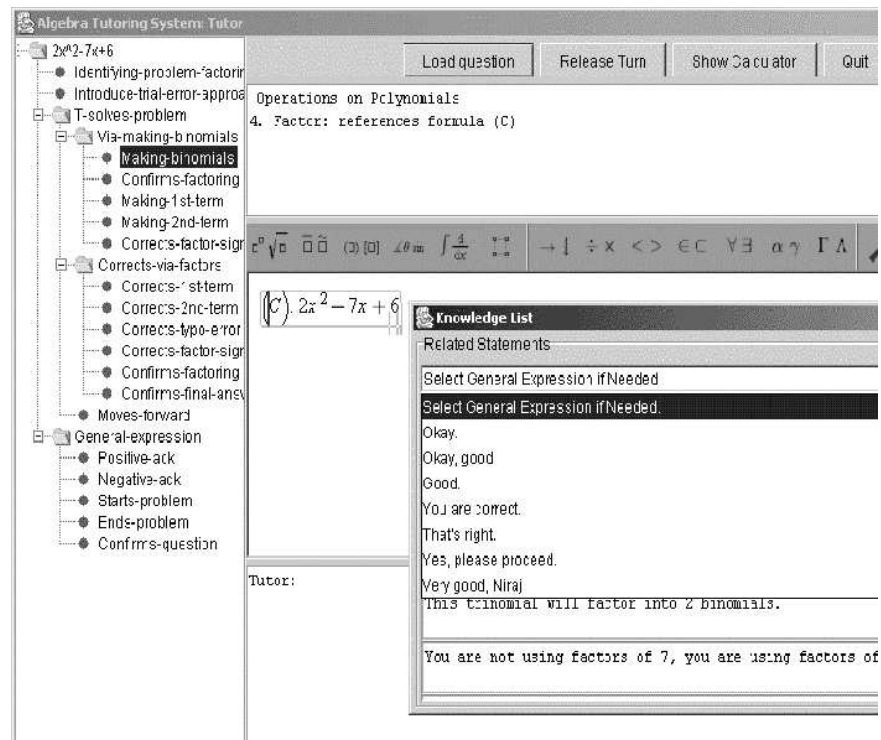


Fig. 4. Wooz Tutor Computer-Mediated Tutoring Interface, Tutor's Screen

Results and Discussion

We have 6 tutoring sessions where the expert tutor utilized the Wooz structured scenario for the trinomial factoring problem. Thus we have no estimates of statistical significance. The other problems in the same tutoring session were tutored by normal means. We have 15 examples of tutoring this problem without benefit of the structured scenario. The learning gains were 0.75 for the Wooz-assisted sessions and -0.14 (a loss) for the non-assisted sessions. The Wooz-assisted tutoring sessions had only lower achievement (levels 1 through 3) students, while the non-assisted sessions had a more mixed population. Considering only the students at the lower achievement levels gives a learning gain of 0.75 for Wooz and 0.0 for the unassisted tutors. Note also that the Wooz-assisted gains compare favorably to the 0.35 gain over all problems in all transcripts. These results point toward Wooz-assisted tutoring producing superior learning gains, but the numbers are so small that we do not have statistical significance.

Comparing the number of turns to tutor one problem (both tutor and student combined) and clock time to tutor one problem for Wooz vs. non-Wooz for the same problem, we see that Wooz is a trifle slower and less wordy in the achievement matched group, and a much slower and a trifle more wordy overall. Table 1 shows these results. We would have expected the Wooz assisted dialogue to be faster because of less typing, but this does not seem to be the case.

In the Wooz-assisted dialogues, the tutors almost always followed the suggested tutorial goal schemata. This suggests that we have the goal structure correct. We have not tried the computer-suggested goal structure and dialogue with novice tutors to see whether it affects their tutoring.

Table 1. Statistics for Wooz-Assisted vs. Manual Dialogues on One Problem

	Wooz-Assisted Dialogues	All Manual Dialogues	Achievement-Matched Manual Dialogues
Learning Gain	0.75	-0.14	0.00
Avg. Turns	21.83	20.13	25.30
Avg. Time	18:52	14:05	18:04
n	6	15	10

Of the tutor turns in the Wooz-assisted dialogue, 70% were extracted from the database of canned sentences with no change, 6% were edits of existing sentences, and 24% were new sentences. There is little difference between the edits and the new sentences, it seems that once the tutor started editing a sentence she changed almost the whole thing. The new and changed sentences almost always respond to specifics of student utterances that did not appear in the attested transcripts used in building the sentence database. Here is an example of a modified turn:

St: I'm going to use the quadratic formula.
Tu (original): Is this an equation?
Tu (edited): We use the quadratic formula for quadratic equations. Is this an equation?

This phenomenon, the human tutor responding to specific variations in the student responses, would seem to reduce the Wooz tutor's evaluative probity. When a tutor changes a sentence, we have no way to know whether the unchanged sentence would have worked just as well. Nevertheless, with experience we should build up knowledge of what rates of sentence modifications to reasonably expect. Forcing the tutor to follow Wooz tutor's suggestions would mean that discovering gaps in schemata would become more difficult, making it less useful as an evaluative tool.

Wooz bears a familial similarity to the snapshot analysis technique for evaluating intelligent tutoring systems, for example [14], whereby at various points in the tutorial session the choices of experienced tutors are compared with the choices of the machine tutor. In an ITS project, Wooz could function as a cheap way to partially evaluate the same schemata before they are incorporated into the machine tutor.

The Wooz tutor does not evaluate the completeness or the reliability of coding. It is thus not a substitute for traditional evaluation measures such as inter-rater reliability. But by evaluating whether schemata imputed from transcripts are complete and efficacious it could provide an additional measure of evaluation to a dialogue annotation project. In particular a high inter-rater reliability shows that the analysis is reproducible, not that it is useful. This technique can help fill that gap.

Conclusions

The technique of providing canned tutoring goals structure and sentences to the human tutor in keyboard-to-keyboard tutoring seems to work well for our purpose of evaluating whether we have analyzed dialogue in a useful manner. We can evaluate whether the tutoring language and goal structure are actually complete enough for real dialogues and actually provide effective tutoring.

The input understanding and decision making structures that would be necessary for building an ITS are not evaluated here. The positive result is that Wooz tutor evaluation is cheap and easy, since you do not have to do all the work to commit to working tutoring software. Furthermore you can evaluate only a few small dialogues by mixing them in with ordinary un-assisted tutoring. Compared to techniques for evaluating transcript annotation such as inter-rater reliability measurement, Wooz tutoring provides the advantage that it tests the final transcript analysis in real dialogues.

We have no evidence, partly because of a small number of test cases and partly because we do not force the tutor to follow the machine's suggestions, that the artificial assist to the tutor speeds up the tutoring process or improves learning outcomes.

Acknowledgements

The Wooz tutor and ancillary applications were developed by a hard-working inspired group of NC A&T students that included Niraj Patel, Oliver Hinds, Kevin Purrington, and Jie Zhao. The idea for computer-assisted human tutoring was suggested by Kurt van Lehn, and the algebra tutorial dialogue project was suggested by Martha Evens.

This work was supported by the Cognitive Science Program, Office of Naval Research, under grant N00014-02-1-0164, to North Carolina A&T State University. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

References

- 1 Patel, Niraj, Michael Glass, and Jung Hee Kim. 2003. "Data Collection Applications for the NC A&T State University Algebra Tutoring Dialogue (Wooz Tutor) Project," Fourteenth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS-2003), Cincinnati, 2003.
- 2 Heffernan, Neil T. 2001. *Intelligent Tutoring Systems are Forgotten the Tutor: Adding a Cognitive Model of Human Tutors*. Ph.D. diss., Computer Science Department, School of Computer Science, Carnegie Mellon University. Technical Report CMU-CS-01-127.
- 3 McArthur, David, Cathleen Stasz, and Mary Zmuidzinas. 1990. "Tutoring Techniques in Algebra," *Cognition and Instruction*, vol. 7, pp. 197-244.
- 4 Fox, Barbara. 1993. *The Human Tutorial Dialogue Project*. Lawrence Erlbaum Associates.
- 5 Graesser, Arthur C., Natalie K. Person, and Joseph P. Magliano. 1995. "Collaborative Dialogue Patterns in Naturalistic One-to-One Tutoring," *Applied Cognitive Psychology*, vol. 9, pp. 495-522.
- 6 Person, Natalie and Arthur C. Graesser. 2003. "Fourteen Facts about Human Tutoring: Food for Thought for ITS Developers." In H.U. Hoppe, M.F. Verdejo, and J. Kay, *Artificial Intelligence in Education* (Eleventh International Conference, AIED-2003, Sidney, Australia), IOS Press.
- 7 Carr, Martha and Barry Biddlecomb 1998. "Metacognition in Mathematics from a Constructivist Perspective." In Hacker, Douglas, John Dunlosky, and Arthur C. Graesser, *Metacognition in Educational Theory and Practice*, Mahwah, NJ: Lawrence Erlbaum, pp. 69-91.
- 8 Kim, Jung Hee, Reva Freedman, Michael Glass, and Martha W. Evens. 2004. "Annotation of Tutorial Goals for Natural Language Generation," in preparation.
- 9 Freedman, Reva, Yujian Zhou, Michael Glass, Jung Hee Kim, and Martha W. Evens. 1998a. "Using Rule Induction to Assist in Rule Construction for a Natural Language Based Intelligent Tutoring System," Twentieth Annual Conference of the Cognitive Science Society, Madison, pp. 362-367.
- 10 Freedman, Reva, Yujian Zhou, Jung Hee Kim, Michael Glass, and Martha W. Evens. 1998b. "SGML-Based Markup as a Step toward Improving Knowledge Acquisition for Text Generation," *AAAI 1998 Spring Symposium: Applying Machine Learning to Discourse Processing*. Stanford: AAAI Press, pp. 114-117.
- 11 Person, Natalie K., Arthur C. Graesser, Roger J. Kreuz, Victoria Pomeroy, and the Tutoring Research Group. 2001. "Simulating Human Tutor Dialog Moves in AutoTutor," *International Journal of Artificial Intelligence in Education*, vol. 12, pp. 23-39.
- 12 Heffernan, Neil T. and Kenneth R. Koedinger, 2002. "An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor," *Intelligent Tutoring Systems, Sixth International Conference, ITS-2002, Biarritz*, Springer Verlag.
- 13 Rosé, Carolyn P., Pamela Jordan, Michael Ringenber, Stephanie Siler, Kurt VanLehn, and Anders Weinstein. 2001. "Interactive Conceptual Tutoring in Atlas-Andes." In J. Moore, C. L. Redfield, and W. L. Johnson, *Artificial Intelligence in Education* (Tenth International Conference, AIED-2001, San Antonio) IOS Press, pp. 256-266.
- 14 Mostow, Jack, Cathy Huang, and Brian Tobin. 2001. "Pause the Video: Quick but Quantitative Expert Evaluation of Tutorial Choices in a Reading Tutor that Listens." In J. Moore, C. L. Redfield, and W. L. Johnson, *Artificial Intelligence in Education* (Tenth International Conference, AIED-2001, San Antonio) IOS Press, pp. 243-253.