A Comparison of Decision-Theoretic, Fixed-Policy and Random Tutorial Action Selection

R. Charles Murray[1] and Kurt VanLehn[2]

[1]Carnegie Learning, Inc., Frick Building, 20th Floor, 437 Grant St., Pittsburgh, PA 15219
cmurray@carnegielearning.com
[2]Computer Science Department & Learning Research and Development Center
University of Pittsburgh, Pittsburgh, PA 15260
vanlehn@cs.pitt.edu

Abstract. *DT Tutor* (DT), an ITS that uses decision theory to select tutorial actions, was compared with both a Fixed-Policy Tutor (FT) and a Random Tutor (RT). The tutors were identical except for the method they used to select tutorial actions: FT employed a common fixed policy while RT selected randomly from relevant actions. This was the first comparison of a decision-theoretic tutor with a non-trivial competitor (FT). In a two-phase study, first DT's probabilities were learned from a training set of student interactions with RT. Then a panel of judges rated the actions that RT took along with the actions that DT and FT would have taken in identical situations. DT was rated higher than RT and also higher than FT both overall and for all subsets of scenarios except help requests, for which DT's and FT's ratings were equivalent.

1 Introduction

Intelligent tutoring systems (ITSs) that coach students as they attempt tasks often emulate the turn taking observed in human tutorial dialog [1; 2]. The tutor's main task can be seen as deciding what action to take on its turn, or *tutorial action selection*. Selecting tutorial actions involves inherent difficulties.

A significant source of difficulty is that the tutor is uncertain about the student's internal state because it is unobservable and changes over time (e.g., as the student learns). Furthermore, the tutor is uncertain about the effects of the tutor's actions on the student. Many ITSs [see, e.g., 3] have modeled the tutor's uncertainty in terms of probability using Bayesian techniques [4] for sound yet relatively efficient inference.

Another difficulty is that just what constitutes effective tutorial action depends upon the tutor's objectives and priorities. The tutor's objectives are likely to include increasing the student's knowledge, helping the student complete tasks, bolstering the student's affective state, and being a cooperative discourse partner. It may not be possible to maximize attainment of all objectives at once, in which case the effectiveness of the tutorial action alternatives depends upon the tutor's priorities. Tutors must often strike a "delicate balance" among multiple competing objectives [5, p.280; 6; 7].

Decision theory extends probability theory by considering, in addition to uncertainty, the decision-maker's objectives and priorities as a rational basis for making

decisions [8]. *DT Tutor* (DT) [9] uses decision-theory to decide the tutor's actions. For each alternative, DT computes (1) the probability of every possible outcome of that tutorial action, (2) the utility of each possible outcome relative to the tutor's objectives and priorities, and then (3) the alternative's *expected utility* by weighting the utility of each possible outcome by the probability that it will occur. DT then selects the tutorial action with maximum expected utility. This approach unifies considerations regarding (1) the tutor's uncertain beliefs about the changing tutorial state and the tutor's effects upon it, and (2) the tutor's objectives and priorities.

One advantage of a decision-theoretic approach is the capability to balance multiple tutorial objectives in a principled way. DT leverages this capability by simultaneously considering the student's knowledge, focus of attention, and affective state, along with joint task progress and the student-tutor discourse state. Another advantage is that by looking ahead to anticipate student difficulties and the influence of the tutor's actions, DT can provide *proactive* help to attempt to prevent student difficulties in addition to the *reactive* help that most ITSs provide.

While many ITSs have used Bayesian networks for reasoning under uncertainty [3], decision-theoretic approaches remain rare, and comparisons of decision-theoretic approaches with non-trivial competitors are rarer still. CAPIT [10] and iTutor [11] appear to be the only other decision-theoretic tutors that have been implemented and evaluated. However, these tutors were compared only with no tutoring at all [10], with consulting the teacher when required [11], and with randomized action selection [10]. Our work does not directly assess effectiveness with students, but it does compare decision-theoretic tutoring against a higher standard: a Fixed-Policy Tutor (FT) that selects tutorial actions by emulating the fixed policies employed by successful tutors such as Andes1 [12] and the Cognitive Tutors [13], which are theory-based [14], widely-used and highly effective [15]. DT and FT were also compared with a Random Tutor (RT), which selects randomly from among relevant tutorial actions.

2 The Three Tutors: DT, FT and RT

DT, FT and RT shared the same user interface and help messages. All of the tutors gave immediate flag feedback by highlighting correct responses in green and errors in red. The only difference between the tutors was the method they used to select from among the same tutorial action alternatives, which consisted of deciding whether to provide a help message, and if so, which message to provide. The differences in their performance were thus due solely to their action selection methods.

This study encompassed three different types of help situations. Most ITSs provide help (1) in response to help requests and (2) after some number of errors. Also included were (3) *step starts*, which are opportunities for the tutor to provide proactive help at the start of a step before the student has reached an impasse or made an error. Few ITSs provide help at step start, but human tutors sometimes do [e.g., 6; 16]. Help provided at step start or after an error is proactive since the student has not asked for help and may not even want it. Responses to help requests are reactive.

The tutors could choose either to provide no help message (a *null* help message) or to provide one of four different kinds of help messages: *prompt, hint, teach*, or *do*.

The *prompt* message pointed out pertinent information that was already available in the interface without providing any new information. The *hint* message provided partial information about the step – not enough to teach the student how to do the step but perhaps enough to either remind the student how to do the step or help the student figure it out. The *teach* message provided all the information that the student needed to understand the domain rule related to the step, including at least one example, and thus to help the student complete the step correctly by learning the rule. The *do* message told the student exactly what to do for the current step (e.g., what text to enter) without teaching anything about the related rule. These help messages are ranked in order of increasing explicitness about what to do for the current step, from *prompt* (the least explicit,), through *hint*, *teach*, and *do* (the most explicit).

RT randomly provided help relevant to the current step as follows: For proactive help opportunities, RT decided randomly whether to provide proactive help. For reactive help opportunities, RT always provided help. When RT decided to provide help, it decided in advance a random order for the four types of help messages and then returned help types in that order, repeating the order cyclically if necessary.

FT, consistent with Andes1 [12] and the Cognitive Tutors [13], always provided help after help requests, never provided help for step starts, and provided help after *n* errors (two in this study). When FT provided help, it followed a strong *successive explicitness* constraint: It always provided a help message that was minimally more explicit than any help already provided. In other words, first it provided a *prompt* message, then *hint*, then *teach*, before bottoming out with *do*. If the student continued to request help after that, the *do* message was repeated.

DT's help selection cannot be described in terms of a simple policy because it simultaneously considers multiple aspects of the tutorial state. However, one of these aspects, the student-tutor discourse state (modeled so that DT can be a cooperative discourse partner) did constrain DT's help selections. The two discourse constraints that DT followed were (1) to always provide help for help requests and (2) a weak successive explicitness constraint: It never provided less explicit help than had already been provided (so, e.g., if the student requested more help after receiving a *teach* message, the student wouldn't be disappointed with a *prompt* message). Constraint (1) is the same as FT and RT. Constraint (2), weak successive explicitness, is different than FT in that DT does not have to select the help message that is *minimally* more explicit than any help already provided. This means, for instance, that DT can provide a *teach* message as the first help provided, or progress directly from a *prompt* message to a *teach* message. The other difference between DT and FT's help selection is that DT always considers providing proactive help. It should be noted that DT can easily be configured to emulate FT (and therefore Andes1 and the Cognitive Tutors, among others) by considering only the discourse state (giving it a utility of 1 while giving all other aspects of the tutorial state a utility of 0) and increasing DT's discourse constraints to (1) never provide help for step starts, (2) provide help after errors only after the *n*th error, and (3) follow a strong successive explicitness constraint.

Because DT and FT both followed a successive explicitness constraint, a subset of the help opportunities were especially relevant for revealing differences between the help selection strategies of DT and FT. These were *first-message-opportunity* scenarios (FMOs), in which the tutor has the opportunity to select the first help message to be displayed for the current step. For FMOs, which occurred in over half the 350

scenarios in the study described below, DT had free reign over which help message to select (if any) while FT adhered to its fixed policy. First-message-opportunity scenarios were sometimes partitioned according to student performance on the pretest problem that corresponded to the rule required to complete the problem step: *pretest-wrong* and *pretest-right*. The idea behind this partitioning is that students who get a pretest problem wrong are more likely than those who get it right to need help during tutoring on steps that require knowledge of the rule tested by the pretest problem. This is by no means a perfect test – e.g., the student might have merely slipped on the pretest problem, or the student might have learned the rule since the pretest – but one advantage is that it does not require subjective judgment by the experimenter. It must be noted that DT was not given information about the pretest performance of students in the test set. However, DT could glean information about the likelihood that a particular student in the test set knew a rule in two ways: (1) by the percentage of the training set students who got the corresponding pretest problem correct (learned during phase I of the study as prior probabilities), and (2) by the student's performance during tutoring on steps related to the rule.

3 Study Design

A two-phase study design was employed. In the first phase, *data collection and tuning*, 60 students took a pretest, solved the same five multi-step calculus problems using RT, and then took a posttest. The students used RT so that we could collect data about the effects of *individual* tutorial actions while statistically controlling for the effects of *sequences* of tutorial actions by randomizing over the sequences in which the individual actions occurred. Students were allowed as much time as they needed to complete the problems and most took about an hour. The student data was partitioned into training and test sets of 30 students, which were matched according to pretest scores. Logged student-tutor interactions from the training set, along with pre- and posttest performance, were used to learn probabilities about student knowledge, student behavior, and the effects of tutorial actions. The data collection and tuning phase is described in detail elsewhere [17]. This paper focuses on the *assessment* phase.

During the assessment phase, we replayed logged student-tutor interactions from the test set while recording the responses that DT and FT would provide in the same tutorial situations. When the actions selected by RT and DT differed, the action selected by RT was replayed in order to preserve the fidelity of the replay, and DT updated its model of the tutorial state to include the action actually provided by RT. A similar process was undertaken to record the actions that FT would have taken for the same situations. A panel of judges then rated the actions selected by RT, FT and DT in a large sample of test set situations.

While this study design cannot provide conclusive information about the bottom line – which tutor is most effective with students – it has other advantages. First, it provided data for learning many of DT's key probabilities. Second, it allowed us to compare the action selections of different tutoring approaches in identical situations. Third, it can provide information that is much more detailed than the bottom line about what makes the tutors' actions effective or not in particular situations [18], in-

formation that can be used to improve not only DT but other tutors as well. Advantages two and three allowed us to decrease the grain size of the analysis from the student to the scenario – i.e., to help opportunities. With an estimated effect size for DT over FT of 0.2 standard deviations, we needed about 320 samples from each of three conditions (RT, FT and DT) using the conventional parameters of $\alpha = .05$ and $\beta = .20$ [19], and 960 students was more than we could afford. Reducing the grain size to the scenario allowed us to use many fewer students.

4 The Comparative Assessment

4.1 Subjects

Three paid judges were recruited from among graduate mathematics students who had extensive experience tutoring calculus as well as other mathematics to college and high school students. These judges were considered skilled, although not necessarily expert, because of their extensive mathematical knowledge and tutoring experience.

4.2 Materials

For each scenario to be assessed, judges were given a detailed printed description that included, among other things:

- A screen shot showing the student interface at the moment of the scenario.
- A description of the scenario, including whether the student had just requested help or made an error, along with the correct entry for the current step.
- The student's current number of correct entries, errors and help requests as a general indicator of the student's problem-solving performance and help usage.
- The student's performance on the pretest problem related to the current step.
- *Relevant Action History*: Previous student-tutor interactions on (1) any previous steps that use the rule related to the current step, and (2) the current step.

Each scenario listed the five possible tutorial responses in random order (the help messages corresponding to help types *prompt*, *hint*, *teach* and *do*, plus "no message" for the *null* response) and the judges rated each on a scale of 1 (worst) to 5 (best). The judges also chose their preferred response for each scenario.

Scenario Types and Stratified Sampling. For assessing the performance of the different tutorial action selection methods, 350 scenarios were to be selected from the test set. The intention was to select the scenarios randomly so as not to introduce any bias. However, a completely random selection would have produced a highly skewed sampling among help requests, errors and step starts. Of 5009 scenarios in the test set, 57% were step starts, 35% were errors, and 8% were help requests. This was just the opposite of what was desired for assessing help selection, for which help provided for help requests is arguably most important, help provided for errors is probably next most important, and it is debatable whether help should be provided for step starts at all. With a completely random distribution, the judges' ratings of the tutors would be

dominated by their ratings for step start scenarios and only weakly influenced by their ratings for help request scenarios. Therefore, a stratified sample was selected with the sample for each stratum randomly selected from among all the scenarios in that stratum: 175 help requests, 100 errors and 75 step starts.

One additional criteria was employed for selecting scenarios: Since RT selected relevant actions randomly, it might for a specific step select, say, *do* help (the most explicit), and then if the student was unsuccessful, select *prompt* (the least explicit help). Such sequences of help messages violated even the weaker successive explicitness constraint. It was unclear just how DT or FT should respond following sequences of help messages that violated their own constraints. A related concern was that it was unclear just how such seemingly odd (indeed, random) sequences of help messages would affect the judges' intuitions about what kind of help to provide next. Therefore, scenarios whose Relevant Action History included sequences of tutorial actions that violated the weaker successive explicitness constraint were excluded from the sample.

4.3 Procedure

The judges rated all possible responses for 350 scenarios. Note that with this design it is possible to use the same judges' ratings to assess the tutorial action selections of still more tutors, or updated versions of the same tutors, as long as they use the same student interface and select from the same pool of help messages. The judges were told that they were rating scenarios in order to provide information about what help messages would be best to provide for various situations. They had no idea which tutor provided which responses or that their ratings would be used to compare tutors.

For comparing the tutors, we constructed composite judges' ratings to better represent the population of skilled tutors. Our goal was to discount ratings that were outside the norm without excluding any ratings. To this end, we used the median rating for each response. The median discounts the effect of the magnitude of outlying ratings while still taking their existence into account. With outlying ratings for individual responses thus discounted, composite ratings for sets of responses were computed as the mean of the median rating for each response.

Ratings for All Three Tutors by Scenario Type. Table 1 displays results of paired-sample t-tests comparing RT vs. DT and FT vs. RT for all scenarios and for each scenario type, along with effect sizes and mean composite ratings. Effect sizes were calculated as the difference in means divided by the standard deviation of the control group: either RT or FT as applicable in their comparisons with DT.

As the table shows, the judges' ratings for DT were higher than their ratings for RT overall and for help requests, errors and first message opportunities, significant at level $p<.01$ with effect sizes ranging from .33 to .49. Only for step start scenarios was DT not rated significantly higher than RT after the Bonferroni correction for multiple comparisons. However, the significance before the Bonferroni correction was $p=.012$ and the Bonferroni correction is known to be very conservative to protect against Type I errors. The effect size for step starts was still a healthy .30.

Table 1. Tutor x Scenario Type, paired t-tests: RT vs. DT, FT vs. DT

Comparison			df	t	Sig.	Bonferroni Sig.*	Effect Size
RT vs. DT	RT Mean	DT Mean					
All	2.94	3.43	349	5.746	<.001	<.01	.35
Help Req	3.23	3.66	174	3.937	<.001	<.01	.33
Errors	2.31	2.95	99	3.324	.001	.010	.49
Step Starts	3.11	3.55	74	2.572	.012	.120	.30
FMOs	2.99	3.54	187	5.057	<.001	<.01	.40
FT vs. DT	FT Mean	DT Mean					
All	3.08	3.43	349	5.251	<.001	<.01	.24
Help Req	3.59	3.66	174	1.078	.282	1.0	.06
Errors	2.10	2.95	99	4.693	<.001	<.01	.61
Step Starts	3.19	3.55	74	3.222	.002	.020	.22
FMOs	3.12	3.54	187	4.351	<.001	<.01	.28

* Significance with Bonferroni cor rection for 10 t-tests (Sig. x 10)

The judges' ratings for DT were higher than their ratings for FT overall and for the scenario types of errors, step starts and first message opportunities, all with significance p=.02 or less and with effect sizes ranging from .22 to .61. For help requests, however, DT, with mean 3.66, and FT, with mean 3.59, were rated approximately equivalently with a .06 effect size and a significance level (with Bonferroni correction) of approximately p=1.0.

DT vs. FT for Help Requests. Since DT's and FT's ratings were approximately the same for help requests, one might expect that DT and FT selected mostly the same tutorial responses in the same situations. However, their patterns of responses were significantly different, with a Pearson's chi-square test of association of $\chi^2(3)=60.8$, p<.001. FT and DT also behaved significantly differently for FMO help requests, for which FT always selected the *prompt* response according to its fixed-policy. DT's response selections varied: For the pretest-wrong scenarios, DT selected *prompt* only 34% of the time and *teach* 66% of the time, receiving a mean composite rating of 4.00 while FT received a mean composite rating of 3.55. This difference was significant, t(28) = 2.218, p=.035. For pretest-right scenarios, DT selected *prompt* slightly more often, 44% of the time, and received a mean composite rating of 3.80 while FT's responses (always *prompt*) received a higher mean composite rating, 4.02, although this difference was not quite significant, t(44) = 1.634, p=.109. Apparently, the judges generally preferred the *teach* response when the student was more likely to need explicit help and the *prompt* response when the student was less likely to need explicit

Table 2. FMO scenarios, paired t-tests: FT vs. DT

Comparison	FT	DT	df	t	Sig.	Bonferroni Sig.*	Effect Size
Pretest wrong	2.51	3.24	74	4.606	p<.001	p<.002	.55
Pretest right	3.53	3.73	112	1.776	p=.079	p=.158	.14

* Significance with Bonferroni correction for 2 t-tests (Sig. x 2)

help. DT adjusted its response selections according to the same preference structure but did not adjust them enough when the student was less likely to need explicit help.

DT vs. FT for Errors. FT's ratings for errors were significantly lower than DT's because it always selects a *null* response the first time the student makes an error. All of the judges gave low ratings to *null* responses after errors. 68 out of the 100 error scenarios involved the student's first error, so FT received a low rating for most of the error scenarios. For first errors, DT's mean composite rating, 2.88, was significantly higher than FT's rating of 1.35, $t(67)=8.516$, $p<.001$, with a large effect size of 2.58. On the 32 error scenarios that did not involve the student's first error, FT, with a mean composite rating of 3.69, was rated higher than DT, which had a mean of 3.09, $t(31) = 2.094$, $p=.044$. This was in turn due to DT replying *null* on 13 of these 32 scenarios, for which it received a mean rating of only 1.23 compared to FT's mean of 3.10. The bottom line is that our judges did not like *null* responses to errors.

DT vs. FT for Step Starts. As with errors, FT received lower ratings than DT for step starts because of *null* responses. Per its fixed policy, FT always selected *null* responses for step starts. DT did not reply *null* on 21 of the 75 step start scenarios, and for these, DT's mean composite rating, 3.67, was significantly higher than FT's mean composite rating of 2.38, $t(20) = 3.959$, $p=.001$, effect size .92. DT's significant advantage in ratings when it did not reply *null* led to a significant advantage over FT in ratings for step scenarios overall, 3.55 versus 3.19, $p=.020$.

DT vs. FT for First-Message-Opportunity Scenarios (FMOs). FT always provided either the *null* or the *prompt* response for FMO scenarios, while DT also included the *teach* response and varied its responses according to the likelihood that the student needed explicit help. These differences paid off as DT was rated significantly higher than FT for FMOs, $p<.01$, effect size .28. A closer look shows that DT was nominally rated more highly than FT both for pretest-wrong and for pretest-right scenarios, as shown in Table 2. For pretest-wrong scenarios, DT's mean composite rating is significantly higher, $p<.01$, with effect size .55. For pretest-right scenarios, DT's mean composite rating is not significantly higher after the Bonferroni correction, $p=.158$.

5 Discussion

Fixed-policy tutors such as FT use a time-tested and proven, even theoretically-based [13] policy for selecting the response type for tutorial actions. However, this policy considers only (1) whether the student has just made a help request or the *n*th error, and (2) the most recent response type for the current step. The result is response selections that are all the same regardless of other attributes of the tutorial situation.

FT emulates the policies of Andes1 and the Cognitive Tutors, which follow a strong successive explicitness constraint and volunteer help only after *n* errors [13; 20]. This policy was based on psychological research showing that students remember material better when they generate it themselves. However, even the architects of the Cognitive Tutors and the theory behind them admit that "these may not be the best choices" since, for example, "[s]ome students stubbornly refuse to seek help even when they need it" and "students are often annoyed with the vague initial messages and decide there is no point in using the help facility at all" [13, p. 199]. Once students begin clicking past vague initial help messages, as many as 82-89% of students click all the way through to bottom-out help [20].

DT, like human tutors [1; 5; 6], considers multiple tutorial state attributes to decide when and how to provide help. These attributes include the student's knowledge, affective state and focus of attention, along with task progress and the discourse state. DT's resulting sensitivity to the tutorial state was demonstrated, for instance, in its responses to first-message-opportunities, for which not only did DT's responses vary significantly for pretest-right versus pretest-wrong scenarios, but its ratings were higher for both, significantly so for pretest-wrong scenarios. DT's greater sensitivity paid off in generally higher ratings from the judges.

A major reason why DT surpassed FT in the judges' ratings was DT's use of proactive help, which FT never provides for step start and first error scenarios. Proactive help when a student would otherwise flounder can save time, prevent confusion, provide valuable information at a time when the student is prepared and motivated to learn it, and avoid the negative affective consequences of frustration and failure.

DT's consideration of multiple tutorial state attributes and the variability of its responses is more like human tutors, for whom the timing of feedback appears "to depend critically on the consequences of the particular error or impasse encountered" [5, p.283]. When considering proactive help, human tutors "sometimes seek to forestall errors, sometimes intervene as soon as errors occur; at other times they may allow errors to occur" [6, p.85]. Indeed, the very effectiveness of human tutorial help "may arise because of the contingency of feedback style and content" [1, p. 346].

The bottom line in choosing a method for selecting tutorial actions is which technology delivers the desired capabilities for the least cost (in time and money). If the desired behavior of the tutor is unambiguously defined and only simple capabilities are required, fixed policy is best, no contest, because of its ease of implementation. However, as more sensitivity is required, as the need for flexibility increases, or as the desired behavior of the tutor becomes more ambiguous, decision-theoretic tutoring becomes more attractive. Decision theory can enable a tutor to respond in a principled way to an unlimited variety of situations – even unanticipated situations – without having to come up with a fixed-policy for every combination of uncertain beliefs.

References

1. Merrill, D.C., B.J. Reiser, S.K. Merrill, and S. Landes (1995). Tutoring: Guided learning by doing. *Cognition and Instruction*, 13(3): 315-372.
2. Graesser, A.C., N.K. Person, and J.P. Magliano (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9: 495-522.
3. Jameson, A. (1996). Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction*, 5(3-4): 193-251.
4. Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Mateo, CA: Morgan-Kaufmann.
5. Merrill, D.C., B.J. Reiser, M. Ranney, and J.G. Trafton (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2(3): 277-306.
6. Lepper, M.R., M. Woolverton, D.L. Mumme, and J.-L. Gurtner (1993). *Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors*, in Computers as Cognitive Tools, S.P. Lajoie and S.J. Derry, editors. Erlbaum. p. 75-105.
7. Reye, J. (1995). A goal-centred architecture for intelligent tutoring systems. In J. Greer (eds.), *7th World Conference on Artificial Intelligence in Education*, p. 307-314.
8. Russell, S. and P. Norvig (1995). Artificial Intelligence: A Modern Approach. Englewood Cliffs, NJ: Prentice Hall.
9. Murray, R.C., K. VanLehn, and J. Mostow (2004). Looking ahead to select tutorial actions: A decision-theoretic approach. *International Journal of Artificial Intelligence in Education*, 14(3-4): 235-278.
10. Mayo, M. and A. Mitrovic (2001). Optimising ITS behaviour with Bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, 12: 124-153.
11. Pek, P.-K. (2003). Decision-Theoretic Intelligent Tutoring System. PhD dissertation, National University of Singapore, Department of Industrial & Systems Engineering. ftp://ftp.medcomp.comp.nus.edu.sg/pub/pohkl/pekpk-thesis-2003.pdf
12. Conati, C., A. Gertner, and K. VanLehn (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4): 371-417.
13. Anderson, J.R., A.T. Corbett, K.R. Koedinger, and R. Pelletier (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences*, 4(2): 167-207.
14. Anderson, J.R. and C. Lebiere (1998). The atomic components of thought. NJ: Erlbaum.
15. Koedinger, K.R., J.R. Anderson, W.H. Hadley, and M.A. Mark (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8: 30-43.
16. Fox, B.A. (1993). The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems. Hillsdale, NJ: Lawrence Erlbaum Associates.
17. Murray, R.C. (2005). An evaluation of decision-theoretic tutorial action selection. PhD dissertation, University of Pittsburgh, Intelligent Systems Program. http://etd.library.pitt.edu/ETD/available/etd-08182005-131235/
18. Mostow, J., C. Huang, and B. Tobin (2001). Pause the Video: Quick but quantitative expert evaluation of tutorial choices in a Reading Tutor that listens. In J.D. Moore, C.L. Redfield, and W.L. Johnson (eds.), *10th International Conference on Artificial Intelligence in Education*, p. 343-353.
19. Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences. Erlbaum.
20. Aleven, V. and K.R. Koedinger (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson, and K. VanLehn (eds.), *Intelligent Tutoring Systems, 5th International Conference, ITS 2000*, p. 292-303.