Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems

Ido Roll, Ryan S. Baker, Vincent Aleven, Bruce M. McLaren,
Kenneth R. Koedinger

Human Compute Interaction Institute
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213
idoroll@cmu.edu , rsbaker@andrew.cmu.edu , aleven@cs.cmu.edu ,
bmclaren@cs.cmu.edu , koedinger@cmu.edu

Abstract. Intelligent tutoring systems help students acquire cognitive skills by tracing students' knowledge and providing relevant feedback. However, feedback that focuses only on the cognitive level might not be optimal - errors are often the result of inappropriate metacognitive decisions. We have developed two models which detect aspects of student faulty metacognitive behavior: A prescriptive rational model aimed at improving help-seeking behavior, and a descriptive machine-learned model aimed at eliminating attempts to "game" the tutor. In a comparison between the two models we found that while both successfully identify gaming behavior, one is better at characterizing the types of problems students game in, and the other captures a larger variety of faulty behaviors. An analysis of students' actions in two different tutors suggests that the help-seeking model is domain independent, and that students' behavior is fairly consistent across classrooms, age groups, domains, and task elements.

1 Metacognition in Intelligent Tutoring Systems

Intelligent tutoring systems offer support and guidance to learners attempting to master a cognitive skill [7,11]. When students ask for help or make a mistake in such a tutor, they receive feedback on their problem-solving actions, that is, they receive feedback at the cognitive level. However, mistakes can also be made at the higher metacognitive level, which coordinates the learning process. Such metacognitive skills include self-assessment and help-seeking strategies, among many others. When cognitive errors originate from an incorrect metacognitive decision, feedback on student metacognition would be more appropriate. A tutoring system should try to improve students' metacognitive skills, by, for example, guiding a student who avoids using help to seek help at the right moment.

Several studies have shown that students often make unproductive metacognitive decisions, which affect their learning process [9,13]. Some types of poor metacognitive decisions include avoiding or misusing help [1,16] and attempting to obtain cor-

rect answers without thinking through the material (termed "gaming the system" [4,5]). There is evidence that these types of unproductive metacognitive decisions negatively affect learning. For instance, students with a tendency to game the system on steps they find difficult (e.g., by systematic guessing) tend to learn less from their interaction with the tutor. Similarly, the ability to seek the right help at the right time is correlated with learning, whether with tutoring systems or in the classroom environment [3,16]. Nevertheless, students do not use available help resources effectively enough (for an extensive review, see [2]). For example, Aleven et al. [1] found that when asking for hints, students spent less than 1 second on 68% of the hints before reaching the most detailed hint level, which provides an answer rather than an explanation.

Fortunately, classroom studies have shown that metacognitive skills can be improved through appropriate guidance [15]. Recently, there has also been increasing interest in improving the metacognition of students using intelligent tutors [8,12,14]. While this work focused on improving students' metacognition, such interventions are likely to be more effective if the tutor can detect which students are having metacognitive difficulties, and what the nature of those difficulties is for each student. Work towards detecting such difficulties in the domain of self-explanation is done by Conati et al. [6].

In this paper we present work towards modeling students' metacognition in order to detect errors related to misuse of the system's facilities and help resources. In particular, we compare two models that attempt to capture different aspects of students' metacognition, and show how different populations of students working with different tutors behave similarly when analyzed by one of these models. We also discuss how this work will be used to help improve student metacognition.

1.1 Cognitive Tutors

We explore these issues within the context of Cognitive Tutors, a type of intelligent tutoring system, based on ACT-R theory, that are now used in approximately 5% of US high schools, as part of complete one-year curricula in various math courses. Cognitive Tutor curricula produce learning gains that are around one standard deviation higher than those produced by traditional instruction [10].

Cognitive Tutors are based on cognitive models that detail the skills to be learned and the typical errors students make. Each student's knowledge level is continuously evaluated using a Bayesian knowledge-tracing model [7], enabling the tutor to choose the most appropriate exercises for each student.

Cognitive Tutors provide on-demand help with several levels of contextualized hints. The more hints a student asks to see on a specific problem step, the more explicit the hints become, until the final "bottom out" hint typically gives the student the answer. Some Cognitive Tutors provide additional forms of information resource. For instance, the Geometry Cognitive Tutor has a decontextualized online glossary that lists and illustrates relevant problem-solving principles, theorems, and definitions.

1.2 The Help-Seeking Model

One software agent that we currently develop is the Help-Seeking Tutor. The Help-Seeking Tutor can be added to any Cognitive Tutor (with minor adaptations [1]), and is based on a prescriptive rational Help-Seeking Model. The Help-Seeking Model describes the student's ideal help-seeking behavior; it determines what type of action the student should ideally perform by taking into account the student's estimated mastery level for the skill involved in the step, the number and types of actions so far on the step, and the time the student spends answering.

According to the model (fig. 1), when a student encounters a new step, she is first expected to think about the step (1). Following that, the student judges whether the step is familiar at all (this is approximated by using the tutor's estimation of the probability the student knows the skill). If the student is not familiar with the skill, then she asks for a hint that scaffolds the solution process (2) (note, a hint is not presented automatically, to give the student opportunity to self-assess her lack of sufficient knowledge, and thus practice another important metacognitive skill). After spending some time reading the hint, the student evaluates the hint's helpfulness. If, in the student's own estimation, the hint provides sufficient information, she attempts to solve the step (4). Alternatively, if after reading the hint, the student still does not see how to solve the step, she asks for the next hint.

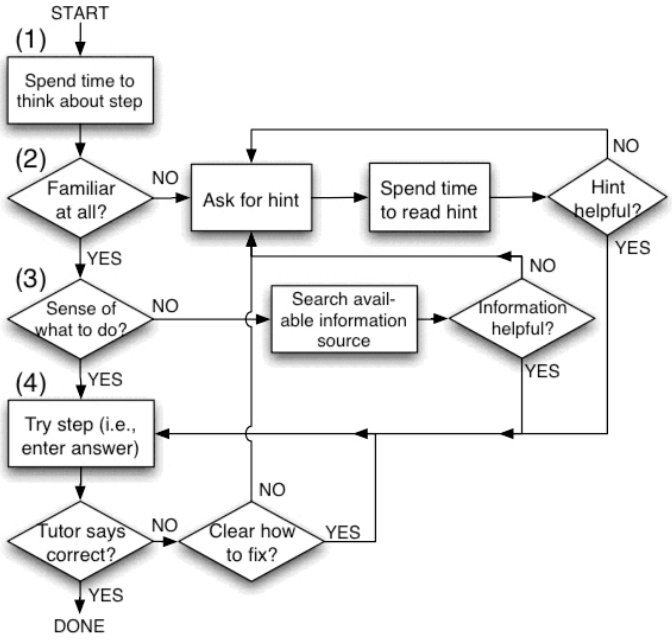

Fig. 1. The help-seeking metacognitive model

A student with a higher skill-level, who is familiar with the step (2), evaluates whether she has a sense of what to do. If the student does not know what to do, the student searches the available information resource (such as the Glossary in the Angles unit or the dictionary in foreign language tutors. Searching decontextualized information resource is an important skill that resembles searching a source such as the WWW) (3). If the information resource is not helpful, the student asks for a hint. Finally, when there is no information resource available, or when the student believes

she can solve the step, she tries to solve it (4). If the answer turns out to be wrong, the student either tries to fix it (if she thinks she knows how), or asks for a hint. Upon successfully completing a step, the student proceeds to the next step and the process repeats itself.

We have implemented this model and the deviations from it as a set of 61 production rules [1]. 29 of these rules capture ideal help-seeking behavior, while the rest capture the various ways that students' help-seeking behavior diverges from the described model. These "metacognitive bug rules" enable the Help-Seeking Tutor to display a metacognitive hint, suggesting to the student what she should do in order to learn most effectively.

The Help-Seeking Model contains 11 categories of such metacognitive errors (e.g., "try step too fast" or "Clicking through hints"), clustered into 4 main bug families (fig. 2): Help abuse, Try-step abuse, Help avoidance, and General bugs [1]. We hypothesize that displaying messages on these metacognitive bugs will be more effective in some instances than the messages at the cognitive level which Cognitive Tutors and intelligent tutoring systems typically provide. For example, a student who is identified as guessing too quickly will receive the following message: "Slow down, slow down. No need to rush. Perhaps you should ask for a hint, as this step might be a bit difficult for you".

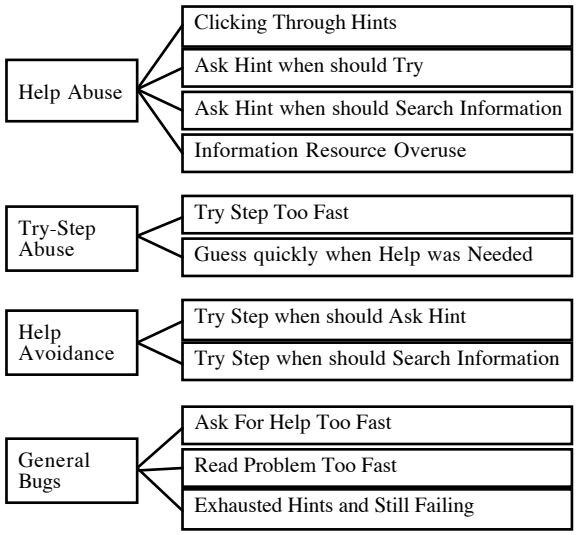

Fig. 2. Taxonomy of Help-Seeking bugs

1.3 The Gaming Detector

A second model that we have developed is the Gaming Detector. Unlike the rational Help-Seeking Model, we used machine-learning techniques to develop the Gaming Detector. Another difference form the Help-Seeking Model is that the Gaming-Detector does not attempt to model ideal behavior. Instead, it is a descriptive model, attempts to capture the behaviors characteristic of those students who use the tutor least appropriately. Specifically, this detector identifies whether a student is attempting to game the system – attempting to succeed on the assigned problems by systematically exploiting properties and regularities in the system, rather than by thinking about the material. Gaming the system has been found to have a greater negative impact on learning than several types of off task behaviors [5]. Once the system de-

tects that a student is gaming, the system can then adapt to encourage the student to use the software more appropriately.

Through classroom observations, we determined what percentage of time each student spent attempting to game the system [5]. Two behaviors were defined to constitute gaming: systematic guessing and clicking through straight to the bottom-out hints.

The challenge that faces the Gaming Detector is to identify Gaming automatically during the interaction with the student. We used machine-learning techniques to develop a Latent Response Model that could identify gaming students by their actions, and used Leave-One-Out-Cross-Validation to be certain that the model was not overfit to individual students through cross-validation [4]. This model first made predictions about whether each individual student action was an instance of gaming, and then aggregated these predictions to make a prediction about the frequency of time each student spent gaming. The model uses a combination of four features to predict that an action is gaming:

1. An action is more likely to be gaming if the student has already made at least one error on this problem step within this problem, and has also made a large number of errors on this problem step in previous problems.
2. Fast actions following an error are likely to be instances of gaming, unlike slow responses to an error.
3. Errors on popup menus are especially indicative of gaming.
4. Slips, where a student makes a number of errors on a well-known skill, are not counted as gaming.

Both the Help-Seeking Model and the Gaming Detector have been found to be effective at predicting how much different students learned [1,4]. In this paper, we use the two models to analyze student behavior across different units and compare the models to each other, in terms of the student behaviors they capture.

2 Comparing Students' Actions Across Tutors

Since the Help-Seeking Model captures a large variety of metacogintive errors that students make, it is of interest to compare students' metacognitive behavior across different tutor units and classrooms using this model. In addition, although the Help-Seeking model was designed to be domain-indepenedent, this premise was not validated until now.

We analyzed two datasets from different groups of students working with different tutors, and calculated the relative frequencies of each bug category in the two datasets. The first of these, collected during 1999, is of 40 students working with the Angles unit of the Geometry Cognitive Tutor for 7 hours. This data includes almost 60,000 actions. The other dataset, collected in 2003, consists of around 20,000 actions performed by 70 students over the course of an hour and a half of use of a tutor unit about Scatterplots.

There are fairly substantial differences between the units and the groups of students in these two datasets, including domain differences (one is for geometry, the other is

in the domain of data analysis), age differences (one group of students was high-school age, while the other was in middle school), differences in available help resources (the Geometry Tutor had a glossary and hints, while the Scatterplot unit only had hints), and finally, differences in task elements (the Geometry Tutor had only numeric fields for the students to fill in, while the Scatterplot unit had numeric and text fields, multiple choice questions, and point plotting on a graph).

Using the help-seeking model described above, we classified each action in the datasets as either metacognitively correct or as in a bug category. Not all bugs could occur in both units. While the Angles unit has an available information resource in the form of a Glossary, the Scatterplot unit has none. In that case, the help-seeking model predicts that students that do not need hints will try to solve the step immediately. This distinction between the units creates a minor difference between the applications of the model; while a student with an intermediate skill level on a certain step is expected to consult the Glossary in the Angles unit, she is expected to try to solve the step in the Scatterplot unit.

Due to differences in the logging mechanisms, successive hint requests in the Scatterplot dataset were captured as single hint requests. In order to draw valid comparisons between the two units, we recoded successive hints in the Angles dataset as single hint requests as well, thus eliminating the "clicking through hints" category.

Results. We compare the patterns of students' metacognitive behavior by comparing the frequencies of the metacognitive bug categories (table 1). Although the domains, age groups, help opportunities and task elements were all different, the frequencies of the bug categories were remarkably similar: $r=0.89$ ($F(1,6)=18$, $p<0.01$). In other words, students tended to make similar metacognitive bugs such as overusing help across the two units.

As seen in table 1, all bug categories have similar frequencies across the units except the "Try step too fast" category. While 17% of the actions in the Angles unit are categorized as "Try step too fast", as many as 36% of the actions on the Scatterplot unit fall under the same definition. Actions categorized as "Try step too fast" are appropriate solution attempt that are done too fast (in less than 5 seconds).

The explanation for this difference possibly lies in the different interface elements between the two units. The Angles unit has only numeric fields, while The Scatterplot unit has also multiple-choice questions (implemented as radio-buttons and popup menus) that are faster to answer, thus reducing the overall duration of the action. In addition, we hypothesize that students in the Scatterplot unit reach mastery in procedural skills faster than student in the Angles unit, thus become experts more quickly, and work with the tutor more rapidly.

The Help-Seeking Model was considerably more successful in predicting learning in one data set than in the other. There was a significant correlation between students' frequencies of Metacognitively Correct Steps and their posttest scores (when controlling for pre-test) in the Angles unit, but not in the Scatterplot unit (Angles unit: partial $r=0.74$, $t(37)=4.7$, $p<0.001$; Scatterplot unit: partial $r=0.08$, $t(67)=0.7$, $p<0.5$).

A probable explanation to the lack of correlation between metacognitive bugs and leaning in the Scatterplot unit resembles the explanation for the inflation of "Try step

Table 1. the properties of the databases and the frequencies of the metacognitive bugs

Unit		**Angles unit**		**Scatterplot unit**
Domain		Geometry		Data analysis
Age group		High school		Middle school
Task elements		Numeric fields		Numeric, text, plot multiple choice
Available help resources		Hints and a glossary		Hints
Bug categories		All actions	Actions comparable to Scatterplot unit	
Help Abuse	Clicking through hints	33%	0%	Not logged
	Ask hint when should Try	1%	2%	1%
	Ask hint when should Search Information	1%	2%	N/A
	Information Resource Overuse	1%	2%	N/A
Try Step abuse	Try step too fast	11%	17%	36%
	Guess quickly when help was needed	7%	12%	11%
Help Avoidance	Try step when should Ask Hint	8%	13%	11%
	Try step when should Search Information	3%	5%	N/A
General Bugs	Ask for help too fast	3%	6%	3%
	Read problems too fast	1%	3%	1%
	Exhausted hints and still failing	1%	2%	0%
	Metacognitively Correct	28%	36%	38%

too fast", as noted before: the Help-Seeking Model might not capture appropriately skilled students' behavior on skills they know well. Fast actions of skilled students are interpreted as being too fast, while they actually may be a result of their high knowledge level. In that case, we would expect that students who reach mastery level will perform more "Try step too fast" bugs, and indeed, there is marginally significant **positive** correlation between "Try step too fast" and posttest scores (when controlling for pretest, $r=0.2$, $t(67)=2$, $p=0.075$). Moreover, when eliminating the "Try step too fast" category, we do find a negative, marginally significant correlation between posttest scores and metacognitive bugs (when controlling for pretest, $r=-0.2$, $t(67)=-2$, $p<0.09$). In other words, besides the "Try step too fast" bug that has a positive correlation with learning, we observe a negative correlation between metacognitive bugs, as captured by the help-seeking model, and successful learning also in the Scatterplot unit.

3 Comparing the Two Models of Metacognition

In addition to comparing the units, we tried to learn more about the students' metacognitive errors and about the way they are captured by the models, by applying both

models to data from the same unit – in this case, the Scatterplot unit described above. The gaming behavior that the Gaming Detector captures consists of help abuse and systematic guessing. These behaviors are related to errors captured by the Help-Seeking Model, so a comparison between the two approaches is of interest. As mentioned before, the Help-Seeking Model is a prescriptive rational model, while the Gaming Detector is a descriptive machine-learned model. Since the two models focus on metacognition differently, there may be benefit to using them in complementary fashion.

The two models are significantly correlated; students who perform more correct metacognitive actions according to the Help-Seeking Model, engage less frequently in gaming behavior as predicted by the Gaming Detector ($r = -0.42$, $F(1,69)=15$, $p<0.001$). The correlation between the sum of the Help-Abuse and Try-step Abuse, (the two bugs in the Help Seeking Model which correspond to the observations used to train the gaming detector), and the output of the Gaming-Detector, is lower and only significant ($r=0.20$, $F(1,69)=3$, $p<0.1$).

The observed frequencies of gaming give us a standard for how successful each model detects attempts to game the system. The Gaming Detector reveals significant behavioral differences between students who game the system but still learn (these "gamed-not-hurt" students were found to game on the easiest steps) and students who game the system and fail to learn (these "gamed-hurt" students were found to game on the hardest steps [4]). To be maximally useful, a metacognitive model should accurately identify the gamed-hurt students —it is not as important to identify the gamed-not-hurt students.

Table 2. distinguishing game-hurt from game-not-hurt students

	Gamed-Hurt	Gamed-Not-Hurt
Gaming Detector	r = 0.77 (F(1,68)=100, p<0.001)	r = 0.06 (F(1,69)=0.3, p>0.6)
Help Seeking Model	r=0.20 (F(1,68)=3, p<0.1)	r=0.03 (F(1,69)=0.1, p>0.7)
Both models in conjunction	r=0.79 (F(2,67)=54, p<0.001)	r=0.09 (F(2,67)=0.3 p>0.7)

As seen in Table 2, the Gaming-Detector is effective at detecting Gamed-Hurt students, while its correlation with the degree of gaming in gamed-not-hurt students is not different from random. Interestingly, the Help-Seeking Model also distinguishes between the two groups, without having been designed to. Its correlation to gamed-hurt is marginally significant, and it is not correlated to gamed-not hurt. Moreover, using the models together is more effective than either alone ($t(67)=1.9$, $p<0.07$). Though the improvement is modest, it suggests that the Help-Seeking Model captures gaming-related behavior that is not explained by the Gaming-Detector alone.

While the Help-Seeking Model identifies the Gamed-Hurt students, it does not find that they game mainly on the hardest steps (as found by the Gaming-Detector). The reason for that may be, as mentioned before, that the model might capture fast answers on mastered (and thus easy) skills as being inappropriate, and therefore has a high rate of metacognitive bugs on easy steps.

4 Discussion and Conclusions

Though tutoring systems are capable of effectively tracing students' cognition and giving relevant feedback on that level, in order to maximize learning, they should also respond to students' poor metacognitive decisions. The first step in doing so is detecting metacognitive errors. Since metacognitive behavior is not tied to a specific subject matter, improving it may improve learning across tutors and domains.

We have developed two metacognitive models: The Help-Seeking Model is a prescriptive rational model aimed at modeling appropriate help-seeking behavior and detecting ineffective help-seeking behavior. The Gaming Detector is a machine-learned model that identifies students who are trying to make progress in the curriculum by systematically exploiting the tutor's properties without thinking about the problems.

We investigated two different datasets using the Help-Seeking Model. Though the datasets differ in age group, subject matter, task elements, and help resources, we found a consistent pattern in students' metacognitive errors. This finding suggests that the help-seeking model is not tied to a specific domain, and can be generalized to tutors that include different tools. While the rates of most bug categories were remarkably similar across units, this was not the case for the "Try step too fast" category. In addition, while in the Angles unit the Help-Seeking Model was a good predictor of learning, in the Scatterplot unit we did not observe such a correlation. A probable explanation to both findings is that the Help Seeking Model falsely classifies appropriate fast actions as buggy ones, which is more relevant to the generally faster Scatterplot unit.

Beyond this, we compared the two approaches to model metacognition, within the context of one dataset. While targeted at different goals, both models successfully distinguish gaming students from other students; combining the models yields better results than either of the models can obtain alone. Though the two models are correlated, they seem to capture different properties of students' behavior. The Gaming-Detector is much more successful at identifying gaming students who do not learn, which is its main goal; the Help-Seeking Model can identify a larger span of metacognitively faulty behaviors. Future development of metacognitive models should combine the advantages of the different approaches.

These results show us that a tutoring system can be aware of a user's metacognition. Once refined, the models will be used to extend the Cognitive Tutors so that they provide feedback to students on their metacognitive behavior. We will examine whether students improve their metacognitive skills following targeted appropriate feedback, and ultimately whether they become better learners as a result.

Acknowledgments: We would like to thank Matthew W. Easterday and Eun-Jeong Ryu for their assistance. This research is sponsored by NSF Award IIS-0308200. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

References

1. Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R.: Toward tutoring help seeking - Applying cognitive modeling to meta-cognitive skills. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* Springer Verlag, Berlin (2004) 227-239.
2. Aleven, V., Stahl, E., Schworm, S., Fischer, F., Wallace, R.M.: Help Seeking in Interactive Learning Environments. *Review of Educational Research,* 73(2), (2003) 277-320
3. Arbreton, A.: Student Goal Orientation and Help-Seeking Strategy Use. In S. A. Karabenick (Ed.), *Strategic help seeking. Implications for learning and teaching,* Erlbaum, Mahwah (1998) 95-116
4. Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting Student Misuse of Intelligent Tutoring Systems. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* Springer Verlag, Berlin (2004) 531-540
5. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z.: Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System". In *Proceedings of ACM CHI 2004: Computer-Human Interaction (2004)* 383-390
6. Bunt A., Conati C., Muldner K.: Scaffolding Self-explanation to Improve Learning in Exploratory Learning Environments. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* Springer Verlag, Berlin (2004)
7. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction,* 4, (1995) 253-278
8. Gama, C.: Metacognition in Interactive Learning Environments: The Reflection Assistant Model. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* Springer Verlag, Berlin (2004) 668-677
9. Gräsel, C., Fischer, F., Mandl, H.: The Use of Additional Information in Problem-Oriented Learning Environments. *Learning Environments Research,* 3 (2001) 287-305
10. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education,* 8 (1997) 30–43
11. Linton, F., Bell, B., Bloom, C.: The Student Model of the LEAP Intelligent Tutoring System. In *Proceedings of the Fifth International Conference on User Modeling,* UM96 (1996) 83–90
12. Luckin, R., Hammerton, L.: Getting to Know me: Helping Learners Understand Their Own Learning Needs through Meta-cognitive Scaffolding. In *Proceedings of Sixth International Conference on Intelligent Tutoring Systems,* ITS 2002, Springer Verlag, Berlin (2002) 759-771
13. Renkl, A.: Learning from worked-out examples: Instructional explanations supplement self-explanations. *Learning & Instruction,* 12, (2002) 529-556
14. Roll, I., Aleven, V., Koedinger, K.R.: Promoting Effective Help-Seeking Behavior through Declarative Instruction. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems,* ITS 2004. Springer Verlag, Berlin (2004) 857-859
15. White, B., Frederiksen, J.: Inquiry, Modeling, and Metacognition: Making Science Accessible to all Students. *Cognition and Instruction,* 16(1) (1998) 3-117.
16. Wood, H., Wood, D.: Help Seeking, Learning and Contingent Tutoring. *Computers and Education,* 33,(1999) 153-169.