Towards a More Transparent Tutor: Opening up Assessment and Control Processes to Learners

Tyler Robison[1], Steven Tanimoto[1]

[1] Computer Science and Engineering Department,
Seattle, WA, USA
{trobison,tanimoto}@cs.washington.edu

Abstract. With the substantial complexity of some assessment and decision-making techniques used in computer tutorials (use of Bayes nets, new modes of interaction, complex production rule systems, etc.) there is a need for mechanisms that allow both authors and learners to inspect, question and otherwise investigate the tutorial's structure and processes. This paper proposes a tutorial system with transparency built-in to most of its components, so that learners can explore what the system is doing, including such things as how their inputs are used, how outputs from the system are calculated, what the modes or 'thoughts' of the system are, and how pedagogical decisions are made. We argue the increasing importance of these capabilities, their potential for fostering metacognition, and offer an analysis of the opportunities involved.

Keywords: transparency, assessment, metacognition, authoring

1 Introduction

Some computer tutoring systems seek a more productive tutoring experience by emulating aspects of human tutors, such as engaging learners in a dialogue about their knowledge [1] or inferring various characteristics of learners [2,3,4] that may be used in making pedagogical decisions. These forms of interaction have great potential for helping learners, yet they also have potential for complicating an already complex learning process. If new forms of interaction are available to learners, such as questioning or challenging the tutor, the learners must understand them to use them effectively. If sophisticated techniques are used to assess learners, such as Bayesian networks [5,4], how are the learners to understand the implications of their actions and reason about the assessment process? In Dimitrova's work [1], the flow of the tutorial was found frustrating at times because of lack of explanation of why it acted as it did; some perceived it to be jumping around without any obvious reason. In this work we explore the possibility of making the tutorial process transparent, so that learners are better able to understand the tutorial and their interactions with it, and perhaps develop metacognitive skills as a result.

Many researchers are interested in the concepts of openness and transparency in order to help improve and refine educational technology systems [1,5,6,7,8, 9,10,11,12,13]. Open systems allow users to 'see inside' the system's inner workings

and observe such things as what the system believes about the learner [5,6,9,11,12,14], representations of the domain [8,12], or the learner's standing when compared to expectations or peers [6,7]. Open systems can also allow more than merely viewing the inside of the system, they may also allow the learner to interact with it in new ways, such as negotiating with the system over the learner model [6,14,15], or entering into a dialogue about the domain knowledge [1]. The system Crystal has allowed users to ask 'why' of actions in a word-processor user-interface and receive text answers, allowing them to figure out why the system behaved as it did [16].

Researchers have argued several benefits that are to be had by making systems open. First, by making the inner-workings visible, designers are encouraged to improve the quality of their models and techniques [6,8,13], as they may now be scrutinized. If the learner can see into their own learner model and interact with it (directly or indirectly), there is potential for that model to be corrected, expanded and refined; as noted by John Self [13], instantiating a learner model can be very difficult, but can be made easier if the learner is included in the loop. Another possibility of openness is that of improving the learner's metacognition; by seeing into their own models, learners may reflect on that information, and improve their understanding of their own knowledge [1,6,7,14]. Some researchers have also argued the importance of developing an understanding of a system in order to better make use of it, take advantage of its features, and be aware of its limitations [9,10], and open and transparent systems may aid a learner in that development.

In this work we explore this issue by applying concepts of transparency to the structure and processes of a computer tutorial teaching propositional logic; as learners work through the tutorial, they are given a view into the tutor's 'thoughts'. In section 2 we describe the system from a learner's perspective, and the ways they can interact with the transparency, and potential metacognitive benefits that may be gained. In section 3 we discuss the interface for authors, so that they can design a tutor and automatically gain transparency without any extra effort on their part. For purposes of this work, we consider the authors of tutorials and the instructors that run them with students as two different groups, and here we are concerned primarily with the authors.

2 Learner's Interface

The core philosophy is that the learner should, via a simple and direct interface, be able to examine the tutorial in a new way, determine why events occurred as they did, and determine what the consequences of future actions may be. In this section we begin with two hypothetical scenarios showing the assessment process transparency; these will help illustrate its features and interactions.

2.1 Usage Scenario 1: Average Student Bill

To begin with, we'll follow a hypothetical student Bill as he goes through a tutorial system augmented with transparent assessment processes. Bill is enrolled in a college Discrete Structures course in the computer science department, which is currently covering propositional logic. As a study aid, all students in the class are given access to an online program intended to give them experience with propositional logic. It is a mixed initiative system in which the tutorial presents a series of questions and feedback based on responses, but the learner is able to select the category of questions. The system decides the difficulty of the questions in general, but the learner can request easier or harder questions, and the system will adapt accordingly.

Bill is a fairly typical student overall, but has performed lower on a recent homework than he would like and so wants to get some more practice. Bill begins by logging onto a library computer and accesses the program via the course website in order to get some more practice with propositional logic. After browsing through a list to select his topic, 'Propositional Logic', Bill is shown the screen in figure 1a; a question is shown with an input field in which to type in the answer. Below is a meter showing Bill's inferred mastery of the material in general, which starts at 0% as the system has no information yet, and some controls to modify the problem category (currently AND) and difficulty.

Bill clicks on 'Help' to get some basic information on how to interact, then answers the question by typing 'False' into the text field and clicking 'Submit'; he sees a message that he was correct, and that his mastery value increased slightly. He continues answering questions for a few minutes, getting most correct, and notices that the General Mastery value changes a bit. After awhile, he notices that his most recent answer, though correct, did not raise his mastery value, which is at 33%. He remembers his instructor talking about some built-in features to help students understand the tutorial, which his instructor showed briefly in class, and he decides to try and use those to figure out what's happening.

Bill clicks right on the 'General Mastery' meter, and a menu with the following options appears:

- Ask "What does General Mastery mean?"
- Ask "Why is the General Mastery value at 33%?"
- Ask "How can I improve the General Mastery value?"

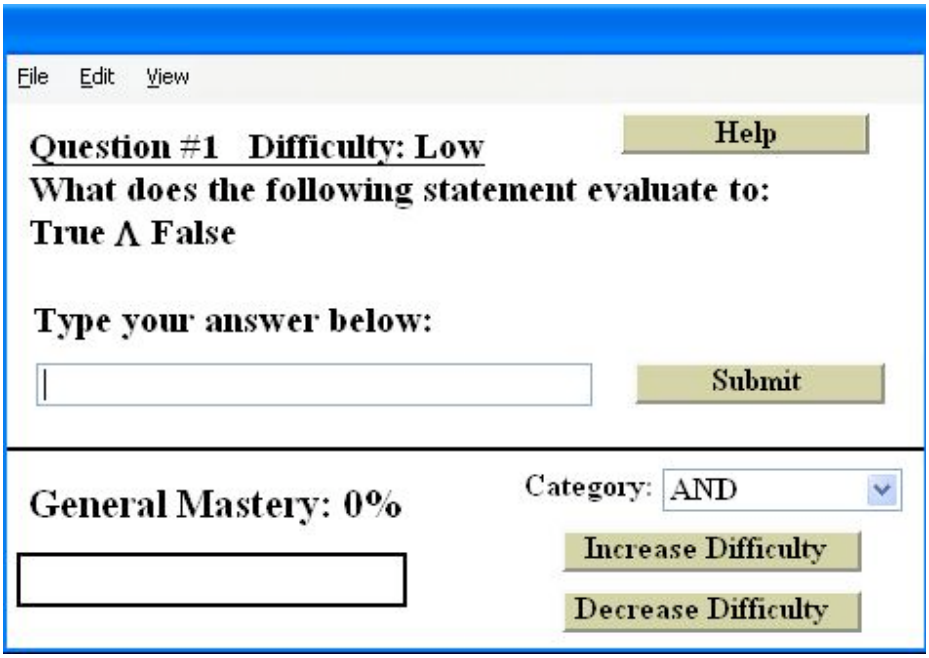
Bill clicks on the second option, which seems to be the most relevant, and sees the window in figure 1b. General Mastery is calculated via a Bayes net and while Bill isn't aware of this, he does get a general impression of what affects General Mastery; it is dependent on AND Mastery, OR Mastery, DeMorgan's Law Mastery and Gaming the System. Note that only a simplified version of a portion of the Bayes net is shown, similar to the Bayes net visualizations of Zapata-Rivera [5]; showing the entire network, complete with probability tables, is likely to be confusing to the learner.

Bill clicks the 'Replay Last Input' button in the new window and sees a brief animation showing the correct answer he gave as it tries to update the AND Mastery, but cannot as it is already at its maximum value; the tutor has seen enough evidence that Bill knows AND. Therefore no change is propagated to the General Mastery value. Bill then closes this new window, right clicks on the General Mastery again,

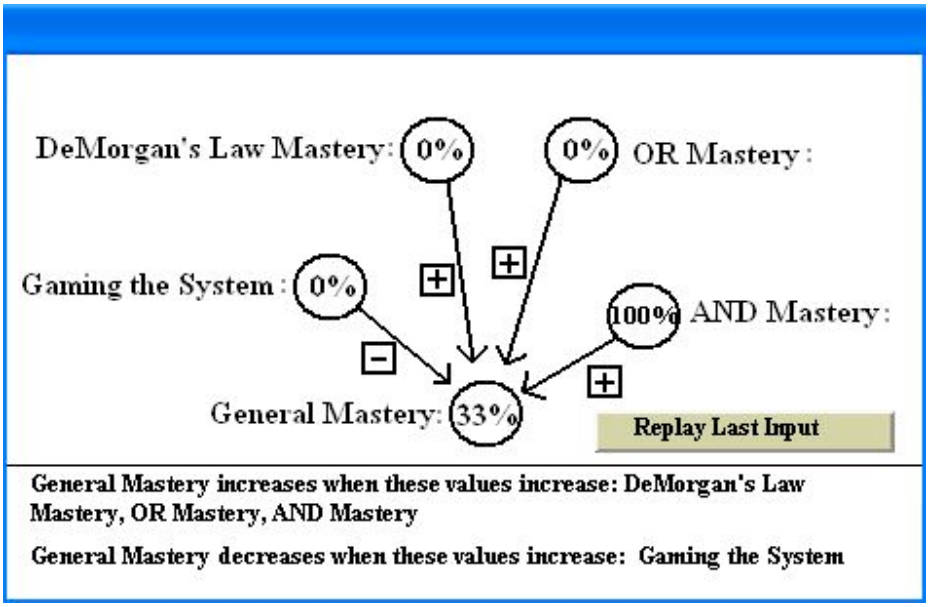
and selects the 'Ask "How can I improve the General Mastery value?"' option. A new window opens up; it is primarily text stating that General Mastery can be increased by increasing AND Mastery, OR Mastery or DeMorgan's Law Mastery, with a note that AND Mastery is already at full. It also states that it can be increased by decreasing Gaming the System, with a note that Gaming the System is at the minimum value. At the bottom of the window is advice to the learner switch categories to DeMorgan's Law and answer some of those questions.

Bill changes the question category to 'DeMorgan's Law' and, after consulting the help pages and attempting several questions, begins to get the hang of DeMorgan's Law and answers a series of the questions correctly. At this point Bill notices that his General Mastery is only slowly increasing, and right clicks to ask the question "Why is the General Mastery value at 40%?", and sees the Bayes net shown before, but updated with new values. He notices that the DeMorgan's Law Mastery is not at its maximum score; confused that it still isn't changing, he right clicks on the 'DeMorgan's Law Mastery' item. A menu with 3 questions of form similar to those above opens up, and Bill selects the question "How can I improve DeMorgan's Law Mastery?". The system responds by opening a window showing and describing the equation used to calculate this mastery value, based on the last 5 questions in that category and their difficulty. At the bottom is the suggestion to increase difficulty.

Bill then increases the difficulty, answers several more questions, noting that the General Mastery and DeMorgan's Law Mastery are now both increasing again when a correct answer is given, and proceeds on with the tutorial.



(a)



(b)

Fig. 1. Mock-up screenshot following the scenario in section 2.1. a) Tutorial as Bill begins. b) Window opened in response to the question "Why is the General Mastery value at 33%?" showing a simplified version of the relevant portion of the Bayes net used to calculate General Mastery.

In this scenario, Bill's primary concern is getting practice with these types of problems and developing a high mastery value, and he uses the transparency as a means to achieve these. Bill used the transparency here because he was uncertain what the tutorial was doing and because he wanted to figure out how to accomplish a certain goal. We expect this type of usage to be common, using the transparency views to help explain unexpected transitions in the tutorial or make a decision regarding goals or strategies. Now let's consider another student, Mary.

2.2 Usage Scenario 2: High Performing Student Mary

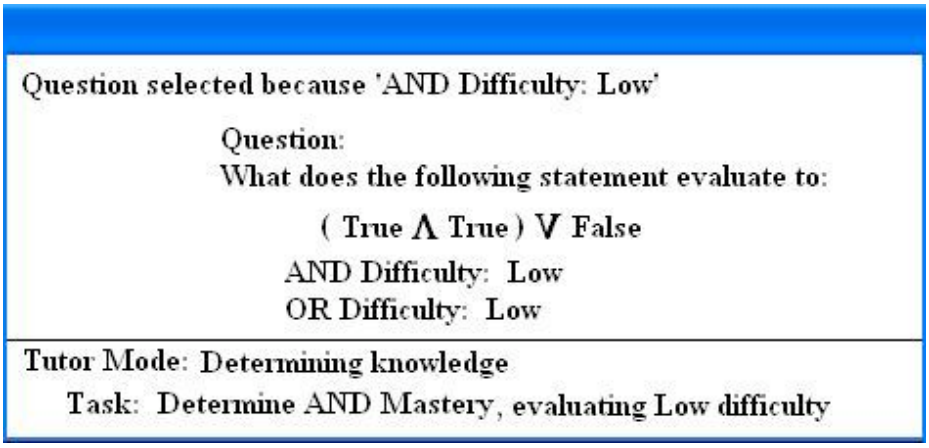
Mary is student in the same class as Bill. Mary has been doing quite well in the course and has one of the top grades in the class. She decides to use the tutorial program in part out of curiosity, and in part to test her knowledge. Mary logs onto a computer, accesses the program, and is about to begin the questions when she remembers the instructor mentioning how you can investigate the inner-workings of the tutorial via a menu you can get by right clicking on different areas. She clicks on the question presented to her, and a menu appears with the following options:

- Ask "Why was this question selected?"

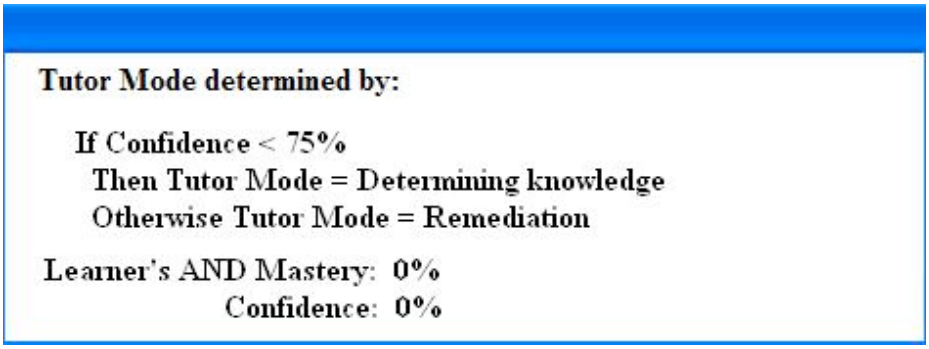
- Ask “How will answering this question affect me?”

Out of curiosity Mary selects the first question, and she is shown the window in figure 2a; the problem was selected because the tutor is inspecting Mary’s knowledge of Low-difficulty AND questions.

Mary looks over the information shown and, interested in this ‘Tutor Mode’, she right clicks on it and asks “Why is Tutor Mode ‘Determining Knowledge’?” to show the window in figure 2b. Mary reads through the information, mentally notes how the tutor functions, and proceeds on with the questions.



(a)



(b)

Fig. 2. Mock-up screenshots following the scenario in 2.2. a) Window opened in response to question “Why was this question selected?” b) Window opened in response to question “Why is Tutor Mode ‘Determining Knowledge’?”

Mary's use of the transparency differs from Bill's, in part because the use began out of curiosity, and not in an attempt to resolve a particular issue. We expect this to be another use of the transparency; exploration out of curiosity and a desire to see what the tutorial has to offer.

2.3 Features Accessible to the Learner

We plan to open up the transparency of the tutorial to learners via a questioning interface similar to Crystal [16], but with increased possibility for exploration. We conceive of our system allowing the learner to ask four kinds of questions of assessment components of the tutorial (and not domain knowledge):

1. 'What is it?': The system responds to this question by giving the learner a brief description of what the component represents; it is akin to information gained from a simple Help menu.
   Example: The system may respond to "What is 'OR Mastery'?" with the text "'OR Mastery' represents your level of knowledge about the OR operator, as inferred by the tutorial"
2. 'Why is this the way it is?': The system responds to this with a justification for a value or decision made.
   Example: The system responds to "Why is 'OR Mastery' at 50%?" by showing the computation made to determine that value (perhaps a Bayes net or equation). Values this computation depended on could be questioned themselves.
3. 'How can I achieve …?': Asking how a certain goal can be obtained gives the learner a description of how that goal can be reached (what values it is dependent on, for instance), and advice about a possible next step.
   Example: The system may respond to "How can 'OR Mastery' be increased?" by showing the other variables on which it depends, and give a suggested first step.
4. 'What would the consequences be if …?': The system responds to hypothetical 'What if?' questions by letting the learner know what the state of the system would be under different circumstances.
   Example: The system may respond to "What if 'OR Mastery' were 100%?" by showing how that change would affect all dependent variables (General Mastery, as it is calculated using OR Mastery).

These components that could be inquired about would range from parts of the learner model to variables used in making pedagogical decisions in the tutorial, such as whether the tutor is trying to assess the learner's knowledge level, or attempting remediation. Other questions could be allowed, such as determining the author's intent on including some aspect by asking 'Why is this structure the way it is?' but for this work we want to focus on the above four. These questions should help learners better understand the tutorial, but they may also help the learner develop metacognitive skills in the process; this is the focus of the next section.

2.4 Metacognitive Potential

Researchers have looked into offering metacognitive feedback to learners [17,18], with the potential for learners to adopt these metacognitive principles and strategies for themselves. We seek a similar result by opening the tutorial up to the learner, letting them see how they are being assessed and how pedagogical decisions are made, with the potential for learners to investigate and perhaps adopt these techniques. Some possibilities are listed below:

- By seeing the mechanisms behind the assessment and pedagogical decisions of the tutorial, learners may appropriate some aspects of the tutor; reflection on these may assist them in self-guided learning, such as the setting of goals and assessing their own progress. For instance, a response the question "Why is 'OR Mastery' at 50%?" may help learners understand how they can assess themselves.
- When faced with a confusing or unexpected event in the tutorial process, and having the tools to investigate on their own, learners may improve their self-reliance and ability to 'debug' a process.
- Seeing inside the tutorial process, and their places in it, may encourage self-reflection, increased knowledge-awareness, and potentially improve confidence.

To what degree and under what circumstances these benefits may be had is a subject we hope to investigate in the future; one of the key purposes of this system is as a versatile platform with which to test different aspects of transparent assessment. As the area of transparent assessment processes has not been well explored, we want to use this system to shed light on possible uses. There are, however, likely to be downsides as well as benefits; the next section discusses some of these.

2.5 Potential Downsides

As with any additional information given to learners, there may be downsides: will it be distracting? Will they understand it? Will they misunderstand it? In addition to these, assessment transparency carries another risk. Researchers have explored the issue of learners gaming the system, its detection and consequences [a], and it is possible that revealing the internals of the system to a learner would only make this easier. As discussed in the next section, the full transparency of the system need not be exposed to all learners in all situations; it would be up to the author or instructor using the system to determine what is too much, and what is the right amount. The flip-side of this issue is the notion that the increased possibility of gaming may encourage authors to put more thought and care into the system than they may when placing the tutorial in a black box. In any case, transparency of assessment processes may well bring both positive and negative effects, and so delving into the effects, strengths, weaknesses, and appropriate context will be an important part of future research.

3 Author's Interface

In the previous section we have presented what the learners would see and how they would interact with the system. In this section we briefly discuss the topic of an authoring system that allows the creation of transparent processes. We believe that, while transparent assessment has much to offer, it is more likely to be used in practice if it is easier to author. Thus, we want to create an authoring system for tutorials that does most of the transparent assessment work automatically, giving the author those views 'for free'. At the same time, it is important that authors be able to expand and customize the transparency, and adapt it to their own purposes. This could extend to reducing the transparency in some cases or eliminating it altogether; the transparency features are meant to be options for authors, not forced upon them.

We plan to give authors a number of tools to make decisions and calculate values in their tutorials; some examples of these tools would be Bayesian networks and if-then rules. Each of these tools would have built-in transparency, and could be combined together by the author to create arbitrarily complex systems. For instance, an if-then rule could be used to determine the mode of the tutorial, where the condition refers to a value, which is in turn calculated by a Bayesian network, which in turn is a function of the learner's inputs. All the author specified values in the tutorial would be defined in such a way, and so if a learner asks a question about how a particular value's dependencies or effects, the system would be able to search through, find all relevant information, and answer the learner's question.

In section 2.3, questions 2-4 would be in part answered automatically by the system, as they primarily involve looking to see what determined a given value, or what a given value determines. One challenge of the design of this automation will be putting this information in a format useful to the learner; for this we intend to build off of similar work, such as work on making the learner model more accessible to learners and Zapata-Rivera's work on helping learners make sense of Bayesian networks shown to them [5].

4 Conclusions

In this work we seek to improve learners' interactions with tutorial systems by making the underlying processes of these systems visible and open to exploration. Not only would this make these systems more useful to learners, but they also have the potential to help foster metacognition. Here we have described how learners would interact with the transparency, and some aspects of the author's interface; our next step in the research will be to create and evaluate such a system.

References

1. Dimitrova, V.: STyLE-OLM: Interactive Open Learner Modelling. In: International Journal of Artificial Intelligence in Education, pp. 35-78. IOS Press (2003)

2. Baker, S.B., 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 CHI, pp. 383-390. ACM (2004)
3. Walonoski, J.A., Heffernan, N.T.: Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In: 8th International Conference on Intelligent Tutoring Systems, pp. 382-391. Springer, Berlin, Heidelberg (2006)
4. Arroyo, I., and Woolf, B.P.: Inferring learning and attitudes from a Bayesian Network of log file data. In Artificial Intelligence in Education, pp. 33-40. IOS Press (2005).
5. Zapata-Rivera, J.D., Greer, J.E.: Inspecting and Visualizing Distributed Bayesian Student Models. In: 5th International Conference on Intelligent Tutoring Systems, pp. 544-553. Springer, Berlin, Heidelberg (2000)
6. Bull, S., Kay, J.: Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework. In: International Journal of Artifical Intelligence in Education, pp. 89-120. 17:2 (2007)
7. Bull, S., Mabbott, A.: 20000 Inspections of a Domain-Independent Open Learner Model with Individual and Comparison Views. In: 8th International Conference on Intelligent Tutoring Systems, pp. 422-432. Springer, Berlin, Heidelberg (2006)
8. Cook, R., Kay, J.: The justified user model: a viewable, explained user model. In: Proceedings of User Modeling, pp. 145-150. Hyannis, MA, (1994)
9. Kay, J.: Learner Know Thyself: Student Models to Give Learner Control and Responsibility. In: International Conference on Computers in Education, pp. 17-24. AACE (1997)
10. Maass, S.: Why Systems Transparency? In: The Psychology of Computer Use, pp. 19-28. Academic Press (1983)
11. Mitrovic, A., Martin, B.: Evaluating the Effects of Open Student Models on Learning. In: 2nd International Conference on Adaptive Hypermedia & Adaptive Web-based Systems, pp. 296-305. Springer, Berlin, Heidelberg (2002)
12. Morales, R. Pain, H., Conlon, T.: Effects of Inspecting Learner Models on Learners' Abilities. In: Proceedings of Artificial Intelligence in Education, pp. 434-445. IOS Press (2001)
13. Self, J.: Bypassing the Intractable Problem of Student Modelling. In: C. Frasson & G. Gauthier (eds.), Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education, pp. 107-123, (1990)
14. Bull, S., Pain, H.: 'Did I say what I think I said, and do you agree with me?': Inspecting and Questioning the Student Model. In: Proceedings of World Conference on Artificial Intelligence in Education, pp. 501-508. AACE (1995)
15. Mabbott, A., Bull, S.: Student Preferences for Editing, Persuading and Negotiating the Open Learner Model. In: 8th International Conference on Intelligent Tutoring Systems, pp. 481–490. Springer, Berlin, Heidelberg (2006)
16. Myers, B., Weitzman, D.A., Ko, A.J., Chau, D.H.: Answering Why and Why Not Questions in User Interfaces. In: Proceedings CHI, pp. 397-406. ACM (2006)
17. Gama, C.: Metacognition in Interactive Learning Environments: The Reflection Assistant Model. In: Proceedings Intelligent Tutoring Systems, pp. 668-677. Springer-Verlag, Berlin Heidelberg (2004)
18. Roll, I., Aleven, V., McLaren, B.M., Koedinger, K.R.: Can Help Seeking Be Tutored? Searching for the Secret Sauce of Metacognitive Tutoring. In: Proceedings of Artificial Intelligence in Education, pp. 203-210. IOS (2007)