



A Framework for User Modeling in QuizMAster

Sima Shabani, Fuhua Lin, Sabine Graf

Athabasca University, Canada

This paper presents an approach to user modeling in QuizMAster, a multi-user educational game shows that uses multi-agent systems to create a personalized learning environment. To keep the students motivated during the games, the system creates, maintains, and uses models of all attending contestants (i.e., learners). Taking advantage of prominent student modeling techniques with some novel ideas, we propose a structure and procedure to create student models and combined student models in QuizMAster. These models are used to create an adaptive environment in the quiz games, selecting and posing questions whose levels of difficulty closely match the knowledge levels of game contestants in certain areas of knowledge; meaning that the game questions for each game are adjusted to the knowledge levels of the learners and the game does not provide too easy or too difficult questions for the contestants. By selecting properly challenging questions for each game show, students are expected to stay motivated and continue

for citations:

Shabani S., Lin F., Graf S. (2012), *A Framework for User Modeling in QuizMAster*, Journal of e-Learning and Knowledge Society, v.8, n.3, 29-44. ISSN: 1826-6223, e-ISSN:1971-8829

benefiting from using the game show, increasing their knowledge through game-based learning.

1 Introduction

Computer games can stimulate users and encourage the development of social and cognitive skills (Mitchell & Savill-Smith, 2004). Quiz games are ideal educational tools that are not limited to specific subjects or user groups (Weng *et al.*, 2011). The element of competition enhances the learning environment by creating excitement among players. And finally, in virtual game environments, usage of interactive and intelligent multi agents increases the unpredictability and complexity of the games making them even more engaging (Maragos & Grigoriadou, 2007).

QuizMAster is an assessment tool that enables students to be tested within a virtual 3D multi-player game (Weng *et al.*, *op. cit.*). Two or more students simultaneously remotely log in to the system via a Web-based interface. Each student or learner is represented by an avatar in the virtual 3D world of QuizMAster. Students are able to view their own avatars as well as those of their opponents. Each game has a game show host who is also represented by an avatar visible to all contestants. The game show host asks each of the game questions from all the contestants. The students hear the voice of the host reading through each question; they also view the questions displayed on their screens. They individually answer each question by, for instance, selecting an answer from available choices in a multiple-choice format. Each correct answer would receive one mark. Figure 1 shows a screenshot from a prototype system of QuizMAster.

User modeling, which is a significant aspect of the QuizMAster architecture, creates and maintains models of learners who participate in the game show. Student models are created for each of the contestants. Each student model stores some data about the learners in order to help agents make the appropriate decisions while interacting with the learners.



Fig. 1 - QuizMAster.

The main objectives of the student models in QuizMAster include helping the pedagogical agents select the adequately challenging questions for each game as well as helping them to match students with similar knowledge level to compete against each other in one game, helping the pedagogical agents pick the informative feedback to be provided to the students at different stages of each game show, and creating a personalized environment for each contestant based on their characteristics and preferences.

Posing the right question is critical to the success of the system. If the questions are too hard, the students will be discouraged to continue playing. On the other hand if the questions are too easy, the students will lose interest in the game. Feedbacks are also key elements of the success of the system. If learners receive appropriate, continuous and personalized feedback during the game, they will stay motivated and engaged throughout the course of the game. A tailored environment adaptive to individual differences in learners can also be an additional motivating and engaging element of the system.

This paper presents a framework for user modeling in QuizMAster. Combination of QuizMAster's features and characteristics, including its multi-agent systems infrastructure for the purpose of learning through games, make it a unique platform for research in the field of user modeling. As a successful multi-agent systems project, with proper user modeling QuizMAster has the potential to achieve a much greater degree of success in providing an adaptive, fun, and encouraging learning platform for students.

This paper is organized as follows. Section one provides an introduction into this paper. Section two contains a review of relevant literature and similar

efforts in creating student models by several researchers in different projects. Section three contains detailed descriptions of the proposed framework for user modeling of single students in QuizMAster. Section four describes an approach for building combined student models for each game. Section five presents challenges and issues, and section six includes conclusions and recommendation for future work.

2 Literature Review

Disengaged students, who are fundamentally unmotivated, and who do not take the e-learning practice seriously, experience a much lower level of success in the learning processes (Baker, 2007). That is the reason why in recent years, student models have been extended to cover many more characteristics that influence motivational aspects of student models. The question that many researchers have recently tried to find the answer to is how to model key aspects of students' affect (D'Mello *et al.*, 2008; Cho & Kato, 2011), motivation (Conati & Maclaren, 2009), and meta-cognition (Metcalfe & Shimamura, 1994; Biswas *et al.*, 2010) and to present the students' individual differences to eventually create highly adaptive systems. The long-term goal of the field of Artificial Intelligence in Education is to support learning for students with a range of abilities, disabilities, backgrounds, and other characteristics (Shute, 2008). Woolf (2009) describes student models and indicates how knowledge is represented, updated, and used to improve tutor performance.

One major element of the success of an adaptive learning tool is how closely student models' aspects are to those of the real students. Researchers have been working on all known aspects of student models for the past few decades (Self, 1990). There are a few e-learning systems, with similar goals to that of QuizMAster which have successfully integrated student models in their architecture. Such examples of e-learning approaches with prominent student models include two e-learning systems, Cognitive Tutoring (CT) (Mitrovic *et al.*, 2003) and ALEKS (Canfield, 2001), and a modeling approach of Constraint-Based Modeling (CMB) (Mitrovic, 2012). Knowledge space theory or cognitive theory of knowledge space (Doignon & Falmagne, 1999) is the base for many student model developments since it provides a structure to describe the states of knowledge of a human learner. In adaptive hypermedia (Brusilovsky, 2001) as the crossroads of hypermedia and user modeling, the system creates a model of the user's characteristics such as goal, preferences, and knowledge.

Learners' skill model (Desmarais & Baker, 2012) is considered an important part of student modeling because by using skill models of learners, intelligent learning applications can define a meaningful learning path for each student to follow and achieve their educational goal. The issue with model of skills is

uncertainty. Two things may happen when assessing a student's skill that would mislead the skill level assigned to the student, slips and lucky guesses.

Multi-user multi-agent user modeling (Vassileva *et al.*, 2003) which is the approach in iHelp's user modeling is an alternative to user modeling server approach. In multi-user multi-agent approach unlike the user modeling server approach the models do not need to be stored in a centralized repository. Moreover, unlike in user modeling server approach where there exists only one user model per user, in multi-user multi-agent approach there is no global consistency when it comes to user information. Vassileva *et al.* (*Ibidem*) explain that in an environment as complex as iHelp, global consistency is an impossible practice due to the existence of many sources of user modeling information in the system.

3 User Modeling in QuizMAStEr

One of the goals of user modeling in QuizMAStEr is for the system to be able to select an appropriate set of students with similar knowledge in a certain domain and group them to compete in one game show, and also for the game questions to be selected appropriately to the level of the group's knowledge. Unlike in multi-user multi-agent approach where the task of grouping user models cannot be achieved easily, in the selected approach of user modeling server, information about user groups can be easily obtained and group user information can be straightforwardly added to the user models (*Ibidem*). This makes user modeling server approach a desirable method for user modeling in QuizMAStEr.

Furthermore, due to its flexibility and on-demand technique for creation of user models, multi-user multi-agent user modeling approach can carry out a wide variety of purposes and is a perfect solution for complex systems. In QuizMAStEr however we have limited number of purposes for user modeling, therefore a simpler approach such as user modeling server can provide sufficient means to achieve our goals of user modeling.

3.1 Student Model Structure

The student model of QuizMAStEr is an internal data structure purporting to describe a student (Self, *op. cit.*).

Five groups of fields comprise each student model; each group includes several attributes:

1. Identification information such as login ID, user password, student ID
2. Personal information such as name, age, gender, residency, first language, picture

3. Educational information: program enrolled in school, fields of study, knowledge level of each field of study
4. Affect and preferences information such as preferred style of learning, preferred type of questions in quizzes, level of enthusiasm for learning, level of confidence, level of motivation
5. QuizMAStEr specific information: including all past attendance in game shows such as number of games played, average score in games, average response times to the questions

A student model is initially created for each student when they interact with the system for the first time. At this time, students are asked to fill out a form to gather simple information such as identification, personal and educational information. For gathering information about affect and preferences, more complex questionnaires are given to students for optional usage.

Attributes of QuizMAStEr specific information is left empty until the student participate in at least one game. Pedagogical agents of QuizMAStEr collect certain data throughout each game show; those data will be added to the student models to make them more complete. There are also some attributes such as level of knowledge that are given an initial value through self assessment, but later on need to be updated according to the results from a knowledge test and quiz games.

3.2 Level of Knowledge

Knowledge level of students is one attribute that needs to be constantly and dynamically updated. As discussed, selecting the right question, aligned with the contestant's level of knowledge is crucial to the success of each QuizMAStEr game show. The knowledge level of each student is ranked on the scale of zero to four as shown in Table 1.

TABLE 1
Knowledge Level and the Corresponding Number in the System

| Knowledge Level | Number in the System |
|-----------------|----------------------|
| Novice | 0 |
| Beginner | 1 |
| Intermediate | 2 |
| Advanced | 3 |
| Expert | 4 |

Steps to determining the level of knowledge

Step 1: Through an initial questionnaire students are asked to assess their knowledge level for each of their domain of study.

Step 2: Self-assessment is important yet it is inherently error-prone due to misjudgment of one's own capabilities (Tsiriga & Virvou, 2003; Hothi & Hall, 1998), to obtain a more accurate evaluation of each learner's level of knowledge, in this next step each student is asked to take a test in every relevant domain of knowledge. Each test has four questions compiled by the course instructor of the relevant subjects. For every test, the instructor will have one question in every category of difficulty of easy, intermediate, difficult, and very difficult. For every correct answer the student obtains one mark, therefore the result of each test can be a score between 0 (corresponding to all questions answered wrongly) to 4 (for all question answered correctly).

The following equation can be used to obtain the level of knowledge in this step.

$$k = \text{average}(s \text{ and } t)$$

Where k is the knowledge level in one domain, s is the number corresponding to the initial self assessment, and t is the result of the initial evaluation test. Since at this stage it is unclear as what factor is the closest to the accurate knowledge level of learners, both s and t are considered factors of the same level of importance. Based on experiments, this assumption can be adjusted.

Step 3: Both steps 1 and 2 happen only once during users' first interaction with the system. Step 3 however takes place every single time the user plays in the game of QuizMAster. As will be explained in the next sub-section, the questions that are asked in each QuizMAster game show are to be picked based on the contestants' levels of knowledge calculated up to the point of that game. Once the result of each game is finalized the knowledge level again gets adjusted accordingly based on the equation below.

$$k = \frac{1}{2} \times \left(\frac{s}{n} * 4 + \hat{k} \right)$$

Where k is the knowledge level to be calculated, s is the score obtained in the last QuizMAster game, n is the number of questions of the last QuizMAster game, and \hat{k} is the current knowledge level from the student model.

For example let's say John Doe whose knowledge level in English Vocabulary is now calculated as 2.5 after step 2, plays for the first time in one

QuizMAster game show of English Vocabulary. This game show happens to have seven questions. John gives correct answers to only three of the questions therefore after this game his knowledge level gets adjusted to

$$\frac{\left(\frac{3}{7} \cdot 4 + 2.5\right)}{2} = 2.1$$

One benefit of the above equation is that the student models' knowledge levels are adjusted as the students improve their level of knowledge by obtaining higher scores in each QuizMAster game show. Let's say, student John Doe now improves his knowledge of Vocabulary and again participate in a Vocabulary QuizMAster game, this time he answers all eight asked questions correctly, therefore after this game his knowledge level in the domain of English Vocabulary is adjusted to 3.05.

3.3 Question Selection and Student's Knowledge Level

QuizMAster contains several databases where information about the users, agents, and the games are stored. One of the databases is the repository of game questions where all the possible game questions as well as metadata such as ontology, answer, question type, and difficulty level are stored (Weng *et al.*, *op. cit.*). Each question is classified by the course instructor giving the difficulty level values of 1 for extremely easy to 10 for extremely difficult. Based on the students' knowledge level, appropriate questions in the field can be selected.

Properly matching questions for each student in every domain of knowledge are the questions where their levels of difficulty match the knowledge level of the student. Table 2 presents the value for the questions' level of difficulty values and the corresponding value for the students' knowledge level.

TABLE 2
Difficulty Level of Questions and Matching Students' Knowledge Level

| Difficulty Level of Questions | Matching Students' Knowledge Level |
|-------------------------------|------------------------------------|
| 1 | 0 – 0.40 |
| 2 | 0.41 – 0.80 |
| 3 | 0.81 – 1.20 |
| 4 | 1.21 – 1.60 |
| 5 | 1.61 – 2.00 |
| 6 | 2.01 – 2.40 |
| 7 | 2.41 – 2.80 |

| Difficulty Level of Questions | Matching Students' Knowledge Level |
|-------------------------------|------------------------------------|
| 8 | 2.81 – 3.20 |
| 9 | 3.21 – 3.60 |
| 10 | 3.61 – 4.00 |

The Host agent in QuizMAster who is in charge of finding the most appropriate questions for a game has to select the questions whose level of difficulty correspond properly to the knowledge level of students according to values of Table 2.

4 Combined Student Model

The purpose of student modeling in most intelligent tutoring systems is to facilitate creation of personalized environment for each learner by identifying characteristics, needs and situations of students. What makes student modeling different in QuizMAster is the fact that there is more than one learner in each session of the QuizMAster game show. Student modeling in QuizMAster should reflect upon more than one user's knowledge level, preferences and characteristics at the same time. Combination of attributes of those students who simultaneously compete in one game show against each other should be considered for the process of student modeling in QuizMAster.

The focus of user modeling is limited to personalization of the game shows at the level where the proper questions are selected for each game. Since in each game there is more than one participating user, here a new concept is proposed named *Combined Student Model*.

A combined student model which is created and maintained by the Game Agent carries attributes for personalization of each game show based on attributes of all that game's contestants. One combined student model is created for each game show and not for each student. The Game Agent, who has access to all contestants' student model, calculates the required values to create the combined student model. The Instructor Agent and the Host Agent also use the values of combined student model, such as combined student knowledge level to select the adequately challenging questions for all playing students in a game show. Figure 2 shows the role of each pedagogical agents and their relationship with the student models, the combined student model as well as the game show and the databases entities.

To illustrate the presentation of each game and the proposed combined student model more clearly, let us assume that four contestants compete in each game of QuizMAster. As mentioned in the previous section, students'

knowledge level is presented by a value between 0 and 4 where 0 and 4 indicate beginner and advanced levels, respectively. To make sure that the contestants competing in a game with approximately the same level of knowledge in the subject, the Game Agent, who is in charge of selecting students for each game show, should comply with a rule that the subtraction of knowledge level of any two students competing in a game should not exceed 1.00. This means that four students with knowledge levels of 1, 2, 2, and 1 can compete in one game while students with knowledge levels of 1, 2, 3, and 2 cannot because the subtraction between the knowledge levels of the first and third students exceeds 1.00.

Once students are selected, the Game Agent creates the combined student model for the game. In this paper, we limit our focus on only one attribute of the combined student model which is the combined knowledge level.

Step 1: The Game Agent gives an initial value to the combined knowledge level, by simply calculating the average values of the students' knowledge levels. For example if our four contestants have knowledge levels of 1, 2, 2, and 1 the initial value of the combined knowledge level is set to 1.5.

Step 2: The combined knowledge level is updated by the Game Agent after every question of the game throughout the game show based on the responses received by the contestants. Since we assume four contestants compete in one game, each contestant's answer affects the value of combined knowledge level by the factor α of the number of contestants (in our case $\alpha = 1/4$).

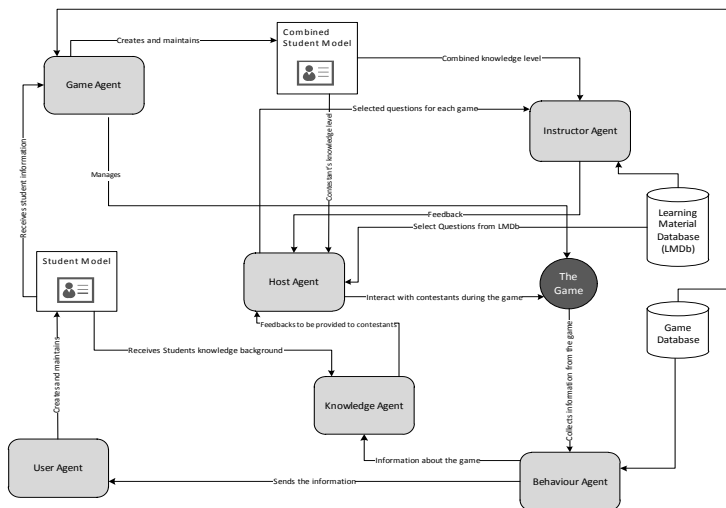


Fig. 2 - QuizMAster's Agents Relationships with Student Models

Every wrong answer of one contestant deducts 0.1 from the combined knowledge level and every correct answer adds 0.1 to the combined knowledge level. 0.1 has come from the fact that in Table 1 every 0.4 in knowledge level changes the corresponding difficulty level of the questions by one and we have four contestants in each game ($0.1 = 0.4/4$). If there would be, for example, eight contestants in each game then the 0.1 factor would need to change to 0.05.

Here is a scenario to clarify the idea: Contestants A, B, C, and D with knowledge levels of 1, 2, 2, and 1 respectively are selected to compete in one game. As mentioned, the initial combined knowledge level is set to 1.5, therefore based on values from Table 2, the questions' in knowledge domain should be of difficulty level of 4. After the first question is asked, one of the contestants gives a wrong answer and the other three give the correct answer. Once the answers are out, the Game Agent updates the combined knowledge level to $1.5 + (3 * .1) - (1 * .1) = 1.7$. Again going back to the values of Table 2, the next question has to have the difficulty level of 5. This process continues until all questions are asked and the game is over.

Throughout the game the combined knowledge level and therefore the difficulty level of the next question to be asked get updated constantly to keep the game at the level appropriate to all contestants' knowledge level.

After all the additions to and the deductions from the combined knowledge level, the combined knowledge level could go larger than 4 or smaller than 0. Both of these occurrences are indicative of the fact that the questions' difficulty levels identified by the instructors need adjustments. If the combined student knowledge level goes above 4, then this tells us that the questions are too easy; similarly it can be inferred that the questions are too difficult if the combined student knowledge level goes below 0.

5 Discussions

In an approach towards student modeling in e-Learning environment by McCalla (2004) called Ecological approach, for each learner interacting with the system, their characteristics and experiences are packaged as learner models. Here is how McCalla summarizes the Ecological approach as: "Each time a learner is interacting with a learning object, the learner model for that learner is attached to the learning object. Over time this means that each object collects many learner models and these can be mined for patterns of particular user to a given application." In the Ecological approach, the information gathered in student models over time, equips the developer with the data they need to evaluate, modify and expand the system. One example of such modification and expansion can be observed in QuizMAster. Proper student modeling in

QuizMAster can adjust the response time for questions to an accurate and precise number close to the average response time. The average needed response time is the average of time that takes students who are aware of the answer of a question to give their response.

Having an accurate number for response time is as important as having an accurate number for the difficulty level of the questions; both of these vectors are the factors that can guarantee the questions of a game to be not too difficult or not too easy and consequently cause the success of each game show.

When the instructors initially add questions to the system, they will be asked to assess the difficulty level of the questions as well as their estimated time needed for the questions to be answered in seconds. The system adds 30 seconds to that estimated time and mark this new time as the maximum time allowed to response to each question. Let us say one instructor adds a question such as “How many continents exist on planet earth?” in the knowledge domain of Geography. The instructor assesses the difficulty level of this question as 2 from the scale of 1 to 10, and the time needed to answer this questions as 40 seconds. Let us say this question is asked in one QuizMAster game where contestants are competing in knowledge domain of Geography. Since the instructor is marked the time for the question as 40 seconds, the contestants are given 70 seconds to respond. If they do not respond correctly within the 70 seconds time, they will not get a score for this question and if they do respond correctly within the 70 seconds, their time to respond correctly is captured. The student model containing the data representing whether or not a student gave the correct answer along with the time to respond, then this student model gets attached to the related question.

Once there are a number of student models attached to the learning object of a question, the estimated time for answering a question can be verified and adjusted if necessary. valuable information can be inferred from the data coming from the student models, one of which is the time needed to respond to the question correctly. Let us assume at one point in time there are 20 student models attached to the above question. Of the 20, 5 are student models of the contestants who did not answer the question correctly and 15 who answered the question correctly. If the average time to respond of all 15 correct answers is very close to 40 we can conclude that the original estimated time by the instructor is correct, and if not, the number 40 can be adjusted to a more accurate number of the average time of all correct answer.

Furthermore, the accuracy of the values of the questions’ difficulty level can be by analyzing through the data retrieved from the student models that are attached to the questions over time, much beneficial information can be attained. One example of such analysis could be for finding out the accuracy of the values of the questions’ difficulty level. This can be achieved by comparing

the difficulty levels of the questions with the knowledge level of students whose respond to them were incorrect.

User modeling for QuizMAster is even more challenging than user modeling in any other intelligent tutoring system for the reason of having multiple learners interacting with the system simultaneously in one game session. In single user e-learning systems user modeling facilitates creating a customized and personalized environment considering one user's skills and preferences. Identifying users' skills and preferences and applying them to create an adaptive system for one user at a time is a difficult task by itself. This task becomes much more difficult when a system like QuizMAster has to deal with multiple users' skills and preferences at the same time.

There are three major challenges along the way of creating and using this proposed framework for user modeling in QuizMAster:

- Issues with accuracy of self assessment: The initial student models rely heavily on values coming from self assessments. Although self assessment is a commonly used technique, it is certainly error-prone. Also, an important question with unknown answer is "Which one's value should be given more weight? Self assessment or results from the initial test." Here the assumption is that they both have the same level of importance so they are given the same weight. This assumption could be inaccurate and further investigations are planned to be conducted.
- Issues with accuracy of assumptions: Since there have been many unknown facts, there have been many assumptions applied to this proposal. For instance, there is an assumption that students whose knowledge levels are different only by the value of 1 are good matches to compete in one game show. Collecting data from a group of students can help with achieving more accurate assumptions.
- Issues with identifying the difficulty level of the games: The proposed technique in this paper looks for students with a knowledge level as close to each other as possible, yet in many cases the students' knowledge competing in the same game could not be identical. In every game there might be some students with higher and some with lower level of knowledge. Should we pick the questions according to the knowledge level of more knowledgeable or less knowledgeable students? The challenge here is to evaluate which one has more negative effects, the loss of interests by students who find the questions too easy or the ones who find the questions too difficult. By averaging out the knowledge level of all contestants, this proposal gives equal weights to loss of interest due to questions being too easy and too difficult.

Conclusions

This paper introduces a student modeling approach for a quiz game called QuizMAster. The proposed structure and procedure for user modeling in QuizMAster in this paper is mostly based on the results from similar studies along with some assumptions and estimations. To make sure that our proposed approach is effective and provides the desirable result, a prototype of student model is under development and will be put in place within the QuizMAster system. Once the system runs with the prototyped student models for a while, data associated with student models will be collected and analyzed. The result of such analysis can identify if the prototype is effective or needs further adjustments. Observing discrepancies, for instance between students' knowledge levels and the difficulty of questions that the students were able to answer correctly, should warn developers that modification is required.

As the first step and for the purpose of this paper the focus has been only on mainly one angle which is selecting adequately challenging questions for each game show according to the contestants' knowledge level. In the next steps additional elements of customization can be added to the system, for instance the type of questions to be posed in a game could be based on students' preferences or the feedback provided by the Host Agent could be based on students' behavior.

Acknowledgements

We thank NSERC of Canada for the financial support to the research.

REFERENCES

- Mitchell A., Savill-Smith C. (2004), *The Use of Computer and Video Games for Learning: A Review of the Literature*, Learning and Skills Development Agency, ISBN: 1853389048, 2004.
- Weng M.M., Fakinlede I., Lin F., Shih T. K., Chang M. (2011), *A Conceptual Design of Multi-Agent Based Personalized Quiz Game*, Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on, 19-21, 2011.
- Maragos K., and Grigoriadou M. (2007), *Designing an Educational Online Multiplayer Game for learning Programming*, In: Proceedings of the Informatics Education Europe II Conference, 2007.
- Baker R.S.J. (2007), *Modeling and understanding students' off-task behavior in intelligent tutoring systems*, In: Rosson, M. B., Gilmore, D. J. (eds.) Proceedings of the 2007 Conference on Human Factors in Computing Systems, CHI 2007, San Jose, CA, USA, 1059–1068, 2007.

- D’Mello S.K., Craig S. D., Witherspoon A.M., McDaniel B., Graesser A. C. (2008), *Automatic Detection of Learner’s Affect from Conversational Cues*, User Model. User-Adapt. Interact. 18(1–2), 45–80, 2008.
- Ortony A., Clore G.L., Collins A. (1988), *The Cognitive Structure of Emotions*, Cambridge University Press, Cambridge, 1988.
- Conati C., Maclaren H. (2009), *Modeling User Affect from Causes and Effects*, In Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: (UMAP ‘09), Geert-Jan Houben, Gord McCalla, Fabio Pianesi, and Massimo Zancanaro (Eds.). Springer-Verlag, Berlin, Heidelberg, 4-15.
- Cho J., Kato S. (2011), *Detecting emotion from voice using selective Bayesian pairwise classifiers*, Computers & Informatics (ISCI), 2011 IEEE Symposium on, 90-95, 20-23, 2011.
- Conati C., Maclaren H. (2009), *Empirically building and evaluating a probabilistic model of user affect*, User Model. User-Adapt. Interact. 19(3), 267–303, 2009.
- Metcalf J., Shimamura A. P. (1994), *Metacognition: knowing about knowing*, Cambridge, MA: MIT Press, 1994.
- Biswas G., Jeong H., Kinnebrew J., Sulcer B., Roscoe R. (2010), *Measuring Self-Regulated Learning Skills Through Social Interactions in a Teachable Agent Environment*, Res. Pract. Technology-Enhanced Learn. 5(2), 123–152, 2010.
- Shute V. J. (2008), *Focus on Formative Feedback*. *Review of Educational Research*, 78(1), 153 -189. doi:10.3102/0034654307313795.
- Woolf B. P. (2009), *Building Intelligent Interactive Tutors: Student-centered strategies for revolutionizing e-learning*, Morgan Kaufmann.
- Self J. (1990), *Bypassing the Intractable Problem of Student Modeling*, In: Proceedings of Intelligent Tutoring Systems, ITS’88, 18–24, Montreal, Canada. 1990.
- Mitrovic A., Koedinger K.R., Martin B. (2003), *A comparative analysis of cognitive tutoring and constraint-based modeling*, In: User Modeling 2003, 9th International, 2003.
- Canfield W. (2001), *ALEKS: A Web-based intelligent tutoring system*. *Mathematics and Computer Education*, 35(2), 152-158, 2001.
- Mitrovic A. (2012), *Fifteen Years of Constraint-Based Tutors: What We Have Achieved and Where We Are Going*, User Model. User-Adapt. Interact. 22, 2012.
- Doignon J. P. , Falmagne J. C. (1999), *Knowledge Spaces*, Springer Verlag, Heidelberg, 1999.
- Brusilovsky P. (2001), *Adaptive Hypermedia*, User Model. User-Adapt. Interact, 11(1–2), 87–110, 2001.
- Desmarais M., Baker R. (2012), *A Review of Recent Advances in Learner and Skill Modeling in Intelligent Learning Environments*, User Modeling and User-Adapted Interaction. 1(22), Springer Netherlands, 9-38, 2012.
- Vassileva J. , McCalla G., Greer J. (2003), *Multi-Agent Multi-User Modeling In I-Help,*” User Model. User-Adapt. Interact. 13, 179–210, 2003.
- Tsiriga V., Virvou M. (2003), *Initializing Student Models In Web-Based ITSs: A Generic Approach*, Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE

International Conference, 42- 46, 9-11, 2003.

Hothi J., Hall W. (1998), *An Evaluation Of Adaptive Hypermedia Techniques Using Static User Modeling*. In: P. Brusilovsky and P. De Bra (eds.): Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia, 45-55, 1998.

McCalla G. (2004), *The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners,*” Journal of Interactive Media in Education, Special Issue on the Educational Semantic Web, 2004.