International Journal of Computer Applications (0975 - 8887)
Volume 70 - No. 26, May 2013

A Personalized Learning System with Adaptive Content Presentation and Affective Evaluation Facilities

Rejaul Karim Barbhuiya
Department of Computer Science
Jamia Millia Islamia
New Delhi, India

Khurram Mustafa
Department of Computer Science
Jamia Millia Islamia
New Delhi, India

Suraiya Jabin
Department of Computer Science
Jamia Millia Islamia
New Delhi, India

ABSTRACT

An Intelligent Tutoring System (ITS) should be able to select appropriate chunks of learning materials as well as evaluate learning outcomes while keeping in mind learner's various meta-cognitive and meta-affective factors. But literature review suggests that such systems are rare as they are complex and time consuming to develop. We have designed an adaptive intelligent tutoring system which is being implemented as a rules-based-expert-system for the dual purpose of - i) adaptive content selection and ii) evaluation of learning gain along with remedial actions. The system is in implementation stage and through this work, we inform in details about the developmental strategies adopted, e.g., use of Java Expert System Shell (JESS) for rules and fact base, Apache-tomcat-server for Java implementation. This work also highlights the rule based implementation of `domain` and `affective` planner along with details about the rules in textual formats. Our student model is able to recognize learner's guessing (gaming) behavior, interest, independence, and confidence level. It can also differentiate - a learner's incorrect answer due to a guess from that due to lack of sufficient domain knowledge. This framework can be used as a guiding principle to build a more robust tutoring system by incorporating other student modeling attributes.

General Terms:

Intelligent Tutoring System, Educational Technology, Rule Based Expert System, User-Modeling and User-Adapted Interaction

Keywords:

Student Model, Affect, Gaming, Guess, Learning Performance, JESS, Physical Sensor, Learning Objectifx

1. INTRODUCTION

A stand-alone adaptive intelligent tutoring system (AITS), if designed successfully, will be able to facilitate anytime, anyplace learning for all kinds of learners irrespective of their age, background, skill/knowledge level, and strengths/weaknesses. Intelligent student modeling is believed to be the key to achieve this goal of personalized adaptation. A good progress has been made in student modeling, but a lot more needs to be achieved to see an intelligent tutoring system (henceforth, ITS) being able to successfully model an experienced human teacher. We studied some of the most notable works in the field of ITS keeping in mind the following two criteria:

(1) The Student Modeling Procedure Employed, and

(2) The Area of Application

1.1 Student Modeling Techniques

A popular approach to student modeling is based on learning style. Identifying and adapting to student's learning style improves performance greatly [20]. Commonly used learning style identification schemes includes Honey-Mumford's learning style [21], Felder-Silverman's learning style [16] and Mayers-Briggs's personality test [29]. Some of the works based on learning style are [8, 11, 20, 37]. However, it is shown that only learning style based instructional process has limited usefulness [6] as they represent only one aspect of learner's characteristics.

Another widely practiced approach is called `affective` student modeling. These systems attempts to identify learner's cognitive as well as various states of mind like emotion, motivation, engagement, frustration, boredom, anger, confidence, gaming tendency, flow, delight,eureka and many other traits to carry out personalized adaptation based on current affective states. Affect has received lots of interest in student modeling [12, 13] and various mechanisms have been proposed to detect student's affective states both statically and dynamically. This includes use of various physical sensors to measure heart rate [24], skin conductance [9, 10], detection of postures [28], conversational cues [14], audio data [17] and combination of different physical sensors [15]. For a detailed review on affective student modeling, refer [7].

However, affective student modeling techniques in general and hardware or physical sensor based student modeling in particular have limitations which makes them less suitable for use in large scale real world systems [4, 31]. Many of them require use of image-processing, pattern recognition, and sometimes their accuracy is also being questioned [38]. These hardware devices could be annoying to the learner [35](p.9) as one has to wear a particular sensor or sit under constant monitoring of a camera. Besides, there are privacy issues also as they collect one's very personal information [35](p.8).

On the other hand, a simpler yet effective student model can be built by minutely observing learner's behavior during an instructional session. We have taken this approach to build the student model for a generic and adaptive tutoring system. We call it a software based student modeling which is less intrusive compared to hardware based student modeling methods. It is discussed in detail at section 2.

Student modeling based on behavioral patterns has been reported in some works. Arroyo and Woolf [3] inferred learner's hidden attitude towards learning by analyzing log files of data collected along four dimensions: 1) problem solving behavior, 2) help activity, 3) help timing and 4) other time related parameters. Aleven et. al., [2] proposed a taxonomy of help seeking behaviors and the kinds of hints to be given by the tutoring system to encourage positive behaviors. Del Soldato and Du Boulay [33] extended the traditional ITS architecture and introduced the concept motivational state modeling and motivational planning. De Vicente and

Pain [36] considered student confidence, satisfaction, effort and interest in their motivation diagnosis rules in MOODS.

1.2 Application of ITS

Our second review criteria was the application domain or the intended purpose of use of an ITS. Traditional e-learning systems couldn't succeed much because they presented static web pages to the learner without proper guidance in terms of navigation and knowledge mastery level. ITS as a complete instructional package should facilitate a learner with suitable and relevant learning materials (both textual and pictorial) as well as evaluation and remedial suggestions. But, none of the existing ITSs have both of these abilities.

For example, SQL tutor [27, 26], which is being used for a long time by a large number of students evaluates learner's query language skills. Similarly, PUMP algebra tutor [22] evaluates algebra knowledge and infers about misconceptions by observing learner's stepwise problem solving process. CTAT [25] as an authoring tool gives the facility of recording a predetermined correct problem solving steps either in JAVA of Flash. ANDES [34] takes a similar approach for physics. They are either designed to be used alongside classroom learning process or expect the learner to know required conceptual knowledge on his/her own effort. However, learner's interest can be greatly enhanced if an ITS can provide1 the required learning material in small chunks in a timely and effective manner.

Keeping these two points in mind, we have developed a framework for an adaptive and generic intelligent tutoring system. It is a complete learning package in the sense that it incorporates both the above mentioned requirements. It is currently being implemented as a production rules based expert system. However, our aim through this work is not to present the framework in greater details as it is being published elsewhere. This work covers the overall working principles of the system in an algorithmic form and highlights a few implemented production rules. The section 2 of the article contains a brief introduction to our student modeling procedure followed by a quick revisit of the hierarchical domain knowledge structure in section 3. Section 4 presents the system architecture from its working point of view and the last section concludes our work.

2. SOFT STUDENT MODELING APPROACH

The student model records six different parameters during a learner's interaction with the ITS, particularly during the problem solving activities. Table 2 gives a brief introduction to these variables. For each action performed by a learner while answering a question, the student model records and updates the values of these six variables stored in the expert systems's fact base. `If-then` type production rules then matches these fact values with their left hand side patterns and the matching rules get fired to carry out appropriate instructional tasks.

3. LEARNING OBJECTS AND THE DOMAIN GRAPH TRAVERSAL

The system represents domain knowledge in a hierarchical top-down structure as given at figure 1. The broad area of study (subject) is divided into modules, each module has multiple topics and each topic is further divided into multiple concepts. For each concept, the system recognizes four different types of primitive instructional ingredients - conceptual knowledge, examples, questions and summarized conceptual knowledge as learning objects (hereafter, it will be referred as `LO` or simply `object`).

Two standard specifications for learning objects based content modeling are SCORM [30] and IEEE-LOM [1]. Many studies have used these guidelines for content modeling [5, 23, 32]. But we have taken a different approach to the concept of learning objects. Here, each micro level instructional ingredients like feedback, problem, example and textual information is represented as an independent learning object. This gives the flexibility to re-use these LOs while teaching other concepts also. For example, while explaining about a concept, if a teacher wants to give an example of another related concept, he can simply refer to the example LO of that concept instead of revisiting the whole concept.

Before describing how the domain an affective planning module works, it is necessary to define certain terms recognized by these two planners. The following definitions explains some of these important terms.

DEFINITION 1 (THEORY). *This learning object, LO_{th}, contains only those necessary domain knowledge required to be studied to understand a particular concept. It can be either in the form of text, picture or a combination of both.*

DEFINITION 2 (EXAMPLE). *Cognitive science have established the fact that while learning (from a book or web-page), a concept is better understood if it contains one or more examples. The example learning object, LO_{ex}, has been designed for this purpose. It contains an example of an event or activity described through a particular concept.*

Table 1. Student Modeling Attributes

Variable Names	Permissible values	Description
Total time spent (TTS)	Less/More	How much time the learner has spent till now on that problem
Help utilized	Yes/No	Whether learner asked for any hint or not while answering the question
Time before first help (TBH)	Less/More	Total time spent by the learner before asking hint/help for the first time.
Number of attempts (NA)	Less/More	Total attempts made for this problem till now.
Task Progress (TP)	Succeeded/Failed	Learners progress on the current problem
Performance history (PH)	Low/Medium/High	Learner's overall performance level

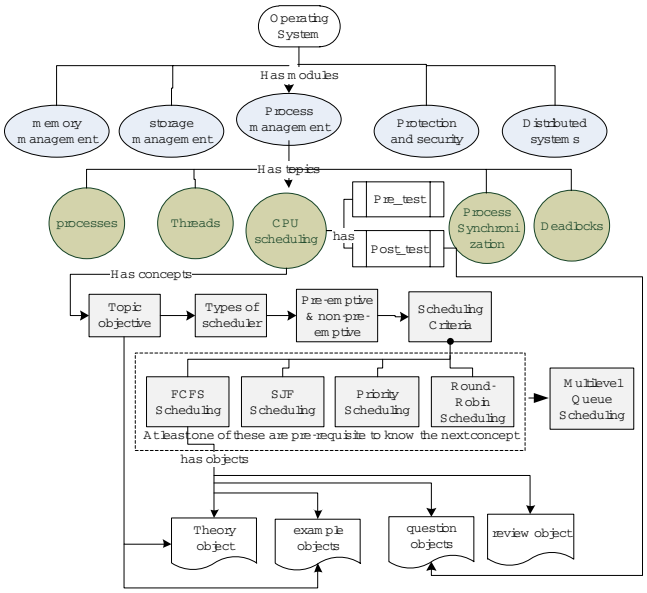

Fig. 1. A Partial Domain Knowledge Structure for the subject - Operating System Concepts

DEFINITION 3 (QUESTION). *LO_{pr}, is the most important learning object from pedagogical module's point of view. Majority of our student modeling attributes are measured during a learner's problem solving activities.*

DEFINITION 4 (REVIEW). *LO_{rv}, contains summarized domain knowledge. This is particularly useful in situations where learner has previously read the LO_{th}, but unable to recall. So, instead of revisiting the LO_{th}, LO_{rv} can be useful in terms of saving time and retaining learner's interest.*

DEFINITION 5 (CONCEPT). *C_i $\leftarrow$[LO_{th},LO_{ex},LO_{pr},LO_{rv}], consists of a theory LO, one or more example LO, one review LO and one or more question LO. A concept C_i has two types of links to other concepts: pre-requisite and followed-by. First link points to those concepts C_j,C_k,.. for which C_i is a pre-requisite. Second link points to the concept C_p, which comes next after completing C_i as per the default domain plan.*

DEFINITION 6 (TOPIC). *T_i $\leftarrow$[C_i,pre-test, post-test], consists of a pre-test, post-test and all those concepts required to be mastered to get a complete understanding about that topic.*

DEFINITION 7 (PRE-TEST). *$Test_{pre}$ $\leftarrow$[C_1,C_2,...,C_n], includes one question from each concept covered under a particular topic. On successfully answering a question in pre-test, the corresponding concept mastery level is marked as* `medium`*, else mastery level is marked as* `not-mastered`*.*

DEFINITION 8 (POST-TEST). *$Test_{post}$ $\leftarrow$[C_1,C_2,...,C_n], also includes one question from each concept belonging to a same topic. The ITS evaluates a learner's overall topic level knowledge through this post-test. The tutoring loop of algorithm 1 starts for each unsuccessful answer at post-test.*

Algorithm 1 explains how the domain knowledge graph of figure 1 is traversed. By default, these paths are required to be traversed by the domain planner. However, sometimes, these paths are not followed. The affective planner, which observes learner's problem solving behavior to infer about certain affective states, takes appropriate steps to keep the learner in a state of mind conducive for learning. As a result, the affective planner often takes over the control from domain planner and in such scenario, the paths mentioned at algorithm 1 may not be followed. We are not covering in details the affective planner here as it is not the intended purpose of this work.

Algorithm 1: Rules for Navigating the domain knowledge graph

Input: Learner's pre-test results; A graph of LOs connected through links
Output: A set of personalized navigational path within the graph structure

1 **Initialization** *Load student model at login and set current topic = previous session current topic. For new student, set current topic = first topic of domain knowledge graph.;*
2 **for** *each selected concept C_i,* **do**
3 present LO_{th};
4 **if** *LO_{th} visited/completed* **then**
5 **if** *has LO_{ex}* **then**
6 present LO_{ex};
7 suggest more LO_{ex} to Low achievers;
8 **end**
9 select a LO_{pr} based on PH;
10 **for** *each LO_{pr} at line 9* **do**
11 **if** *completed and learner is Low achiever* **then**
12 set next LO_{pr} diff_level++;
13 select next LO_{pr};
14 **if** *no LO_{pr} unvisited* **then**
15 select next concept C_i
16 **end**
17 **end**
18 **else if** *completed and a Medium/High achiever* **then**
19 set next LO_{pr} diff_level = Tough;
20 **if** *no LO_{pr} unvisited* **then**
21 select next concept C_i
22 **end**
23 **end**
24 **else if** *Failed and a Low achiever* **then**
25 set next LO_{pr} diff_level–;
26 **end**
27 **else if** *Failed and a Medium/High achiever* **then**
28 select next LO_{pr} of same diff_level;
29 **end**
30 **end**
31 **end**
32 **else if** *learner is a medium/high achiever* **then**
33 goto line 5;
34 **end**
35 **else**
36 goto line 3;
37 **end**
38 **end**

4. WORKING PRINCIPLE

We are currently implementing the proposed ITS as a rules based expert system using the popular Java Expert System Shell (JESS) [19]. We have choosen JESS [19] over other languages like CLIPS, LISP or PROLOG because of the following advantages. JESS implements the popular RETE algorithm [18] in its inference engine, which makes it very efficient and optimize the conflict resolution policy. Besides, JESS has a complete Java API which allows its easy integration with Java. It has the flexibility of being embedded in a Java application and we can also embed Java codes inside a JESS application.

An online implementation of the proposed system is being developed, wherein Tomcat's Apache server (CITE) handles server side java implementation. We have used the model-view-controller (MVC) paradigm for programming the user interfaces. The controller logic will be maintained in servlets, and the view would be built at runtime using Java Server Pages (JSP). The data is stored using Rete objects, thus making the model encapsulated. Figure 2 shows the working modality of our system.

The working memory (fact base) of the expert system contains learner's profile info and his/her previous visit info as well as completion status for each module, topic, and concept. The fact base also maintains learner's performance record at pre-test and post-test, values of affective states like confidence, independence, and interest. Seven of the total fourteen facts implemented using JESS's `deftemplate` construct are given at appendix to give readers an idea.

As mentioned earlier, various conditions laid out for the domain and affective planners have been implemented as JESS rules. On certain mouse events or after a pre-defined time intervals, the `jess engine` gets initialized and updates the fact base reflecting learner's current state. This becomes the active working memory for the rule base. The JESS rules match their left hand side patterns with these working memory values and those matching rules after conflict resolution updates the fact base. The rule base of the present system consists of:

(1) Eleven domain planner rules for selecting appropriate LO as per the plan outlaid in algorithm 1.
(2) Sixteen rules deals with unsuccessful problem solving events and identifies the reason for the failure as well as infers about learner's affective states
(3) Four rules to check against guessing (gaming or help misuse) by the learner.

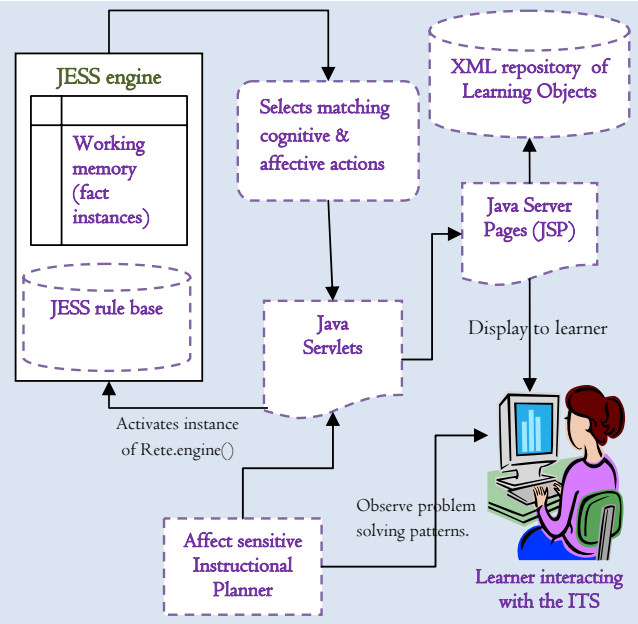

Fig. 2. Working structure of the proposed ITS

(4) Seven rules to select an appropriate feedback message.

Thus, we have twenty eight rules as of now and the numbers will surely increase as we plan to add more functionalities to the system. Below is a listing of the eleven domain planner rules in textual form. A few rules written in JESS has been given at appendix for readers to get a better insight about the implementation methodology.

4.1 Rules for selection of next question

Rule 1 If (learner succeeds in a problem of difficulty-level `easy`) then (select a problem of difficulty-level `medium`)

Rule 2 If (learner succeeds in a problem of difficulty-level `medium`) then (select a problem of difficulty-level `tough`)

Rule 3 If (learner fails in a problem of difficulty-level `medium`) then (select a problem of difficulty-level `easy`)

Rule 4 If (learner fails in a problem of difficulty-level `tough`) then (select a problem of difficulty-level `medium`)

Rule 5 If (this is first question for this `concept` & learner is a medium/high achiever) then (select a problem of difficulty-level `medium`)

Rule 6 If (this is first question in this `concept` & learner is a low achiever) then (select a problem of difficulty-level `easy`.)

4.2 Rules for Traversing the domain

Rule 7 If ('lo-readimg-material' attempted first time, is completed or skipped, and is a Medium/High achiever) then (select an 'example-lo')

Rule 8 If ('lo-readimg-material' attempted first time, is skipped, and is a Low achiever) then (recommend repeat 'lo-readimg-material')

Rule 9 If ('lo-example' attempted first time, is completed or skipped,and is a Medium/High achiever) then (select a 'question-lo')

Rule 10 If ('lo-example' attempted first time, is completed, and is a Low achiever) then (recommend another 'example-lo')

Rule 11 If ('lo-example' attempted first time, is skipped, and is a Low achiever) hen (recommend repeat 'example-lo')

5. CONCLUSION AND FUTURE WORKS

Designing an Intelligent Tutoring System to support both adaptive teaching and affective guidance is a challenging task. Adaptive teaching includes selection and presentation of the most suitable learning material in an effective and timely manner. Affective guidance means selection of relevant feedback as well as deciding when to present the "review LO", checking for incomplete pre-requisite concepts, and navigation to fill learner's knowledge gap. The present implementation targets only a small domain of the computer science subject "operating system concepts". There will be different types of domain knowledge construction templates which will assist the content expert or teacher to either use an existing LO, or edit a learning object based on requirements. The expert may also create a new LO if he/she finds existing learning objects to be insufficient. Each of the various instructional components like descriptive domain knowledge, examples, problems, hints, feedback are being developed as independent learning objects. The present system can be used as a guiding principle to develop a more robust tutoring systems. More functionality can be added for a detailed student modeling covering other affective aspects like boredom, anxiety, fear, disguise, etc. as well as learning style, and various ergonomic aspects effecting learning performances.

6. ACKNOWLEDGEMENTS

The first author is a Senior Research Fellow under UGC-BSR fellowship for meritorious students and this work was supported by the grant. However, all the opinions and findings mentioned in this paper are those solely of authors and not those of the granting agency.

7. REFERENCES

[1] Ieee learning technologies standards committee, learning object metadata (lom) working group 12. http://www.ieeeltsc.org:8080/Plone/working-group/learning-object-metadata-working-group-12/learningobject-metadata-lom-working-group-12, 2007.

[2] Vincent Aleven, Bruce Mclaren, Ido Roll, and Kenneth Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2):101–128, April 2006.

[3] Ivon Arroyo and Beverly Park Woolf. Inferring learning and attitudes from a bayesian network of log file data. In *Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pages 33–40, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.

[4] Ryan S. J. D. Baker, Sujith M. Gowda, Michael Wixon, Jessica Kalka, Angela Z. Wagner, Aatish Salvi, Vincent Aleven, Gail Kusbit, Jaclyn Ocumpaugh, and Lisa Rossi. Sensor-free automated detection of affect in a cognitive tutor for algebra. In *EDM*, pages 126–133, 2012.

[5] Amal Battou, Ali El Mezouary, Chihab Cherkaoui, and Driss Mammass. An adaptive learning system architecture based on a granular learning object framework. *International Journal of Computer Applications*, 32(5):18–27, 2011.

[6] Elizabeth J. Brown, Timothy J. Brailsford, Tony Fisher, and Adam Moore. Evaluating learning style personalization in adaptive systems: Quantitative methods and approaches. *IEEE Trans. Learn. Technol.*, 2:10–22, January 2009.

[7] Rafael A. Calvo and Sidney DMello. Affect detection: An interdisciplinary review of models, methods, and their

applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010.

[8] Hyun Jin Cha, Yong Se Kim, Seon Hee Park, Tae Bok Yoon, Young Mo Jung, and Jee-Hyong Lee. Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In *Intelligent Tutoring Systems'06*, pages 513–524, 2006.

[9] C. Conati, R. Chabbal, and H. Maclaren. A Study on Using Biometric Sensors for Detecting User Emotions in Educational Games. In *Proceedings of the Workshop Assessing and Adapting to User Attitude and Affects: Why, When and How? In conjunction with UM '03, 9th International Conference on User Modeling*, 2003.

[10] Cristina Conati and Heather Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Model. User-Adapt. Interact.*, 19(3):267–303, 2009.

[11] Keeley A. Crockett, Annabel Latham, David McLean, Zuhair Bandar, and James O'Shea. On predicting learning styles in conversational intelligent tutoring systems using fuzzy classification trees. In *FUZZ-IEEE*, pages 2481–2488. IEEE, 2011.

[12] Michel Desmarais and Ryan S. J. D. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, pages 1–30, 2011.

[13] Sidney DMello and Art Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145 – 157, 2012.

[14] Sidney K. D'Mello, Scotty D. Craig, Amy Witherspoon, Bethany Mcdaniel, and Arthur Graesser. Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, 18(1-2):45–80, February 2008.

[15] S.K. DMello, R.S. Taylor, and A. Graesser. Monitoring affective trajectories during complex learning. In *Proceedings of 29th Annual Meeting of the Cognitive Science Society*, pages 203–208, Austin, TX, 2007.

[16] Richard M. Felder and Rebecca Brent. Understanding student differences. *Journal of Engineering Education*, 94:57–72, 2005.

[17] K. Forbes-Riley and D. J. Litman. Predicting emotion in spoken dialogue from multiple knowledge sources. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 201–208, 2004.

[18] Charles L. Forgy. Rete: a fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, page 1737, 1982.

[19] Ernest Friedman-Hill. *Jess in Action: Java Rule-Based Systems*. Manning Publications Co., Greenwich, CT, USA, 2003.

[20] Patricio García, Analía Amandi, Silvia Schiaffino, and Marcelo Campo. Evaluating bayesian networks' precision for detecting students' learning styles. *Comput. Educ.*, 49:794–808, November 2007.

[21] P. Honey and A Mumford. *The manual of learning styles*. Peter Honey Publications, 1992.

[22] Kenneth R. Koedinger, John R. Anderson, William H. Hadley, and Mary A. Mark. Intelligent tutoring goes to school in the big city. *Int. Journal of Artificial Intelligence in Education*, 3:30–43, 1997.

[23] Panayiotis Kyriakou, Ioannis Hatzilygeroudis, and John Garofalakis. A tool for managing domain knowledge and helping tutors in intelligent tutoring systems. *Journal of Universal Computer Science*, 16(19):2841–2861, oct 2010.

[24] Scott W. McQuiggan, Bradford W. Mott, and James C. Lester. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction*, 18:81–123, 2008.

[25] A. Mitrovic, K. R. Koedinger, and B. Martin. A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modelling. In P. Brusilovsky, A. Corbett, and F. d. Rosis, editors, *UM 2003*, volume 2702 of *LNAI*, pages 313–322, Berlin, 2003. Springer-Verlag.

[26] Antonija Mitrovic, Michael Mayo, Pramuditha Suraweera, and Brent Martin. Constraint-based tutors: A success story. In Lszl Monostori, Jzsef Vncza, and Moonis Ali, editors, *Engineering of Intelligent Systems*, volume 2070 of *Lecture Notes in Computer Science*, pages 931–940. Springer Berlin Heidelberg, 2001.

[27] Antonija Mitrovic and Stellan Ohlsson. Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, 10(3):238–256, 1999.

[28] Selene Mota and Rosalind W. Picard. Automated posture analysis for detecting learners interest level. In *Conference on Computer Vision and Pattern Recognition. Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction*, volume 5, page 49. IEEE Computer Society Press, Madison, 2003.

[29] I. B. Myers and M. H. McCaulley. *Manual: A guide to the development and use of the Myers-Briggs type indicator*. Consulting Psychologists Press, Palo Alto, CA, 1985.

[30] Dodds Philip and E. Thropp Schawn. *Sharable Content Object Reference Model (SCORM) 2004 3rd Edition Conformance Requirements Version 1.0*. Advanced Distributed Learning (ADL), 3rd edition.

[31] Jennifer Sabourin, Bradford Mott, and James C. Lester. Generalizing models of student affect in game-based learning environments. In Sidney DMello, Arthur Graesser, Bjrn Schuller, and Jean-Claude Martin, editors, *Affective Computing and Intelligent Interaction*, volume 6975 of *Lecture Notes in Computer Science*, pages 588–597. Springer Berlin Heidelberg, 2011.

[32] G. Santos and A. Figueira. Web-based intelligent tutoring systems using the scorm 2004 specification - a conceptual framework for implementing scorm compliant intelligent web-based learning environments. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 676–678, 2010.

[33] Teresa Del Soldato and Benedict du Boulay. Implementation of motivational tactics in tutoring systems. *J. Artif. Intell. Educ.*, 6:337–378, January 1995.

[34] Kurt VanLehn, Collin Lynch, Kay Schulze, Joel A. Shapiro, Robert Shelby, Linwood Taylor, Don Treacy, Anders Weinstein, and Mary Wintersgill. The andes physics tutoring system: Five years of evaluations. In *Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pages 678–685, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.

[35] Angel De Vicente and Helen Pain. Motivation diagnosis in intelligent tutoring systems. In *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, ITS '98, pages 86–95, London, UK, UK, 1998. Springer-Verlag.

[36] Angel De Vicente and Helen Pain. Informing the detection of the students' motivational state: An empirical study. In *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, ITS '02, pages 933–943, London, UK, 2002. Springer-Verlag.

[37] J. E. Villaverde, D. Godoy, and A. Amandi. Learning styles' recognition in e-learning environments with feed-forward neural networks. *Journal of Computer Assisted Learning*, 22(3):197–206, 2006.

[38] Qiao Xiangjie, Wang Zhiliang, Yu Jun, and Meng Xiuyan. An affective intelligent tutoring system based on artificial psychology. In *Proceedings of the First International Conference on Innovative Computing, Information and Control - Volume 3*, ICICIC '06, pages 402–405, Washington, DC, USA, 2006. IEEE Computer Society.

APPENDIX

Fact base constructed with JESS `deftemplate` construct

```
(deftemplate student
       (slot stud-id)
       (multislot stud-name)
       (multislot stud-clas))

(deftemplate concept
       (multislot lo-names)
       (multislot lo-ids)
       (multislot concept-name)
       (slot concept-id)
       (slot status)
       (slot stud-id)
       (multislot pre-req-concept)
       (slot next-concept))

(deftemplate learning-object
       (multislot lo-name)
       (slot lo-type)
       (slot lo-id)
       (slot reqd-time)
       (slot elapsed-time)
       (slot diff-level)
       (slot status)
       (slot attempt-count (default 0))
       (slot stud-id))

(deftemplate pre-test
       (slot test-id)
       (slot status)
       (multislot question-ids)
       (slot test-score)
       (multislot missing-concept-ids))

(deftemplate prob-solving
       (slot stud-id)
       (slot session-number)
       (slot lo-id)
       (slot question-type)
       (slot attempt-number)
       (slot problem-status)
       (slot mark-scored)
       (slot correct-answer)
       (slot help-taken))

(deftemplate stud-motivation
       (slot stud-id)
       (slot session-number)
       (slot guessing)
       (slot confidence)
       (slot independence)
       (slot interest)
       (multislot because))

(deftemplate recommendation
       (multislot recommended-activity)
       (multislot recommended-lo)
       (multislot feedback-message)
       (multislot because))
```

Rule1 and Rule2 implemented in JESS

```
Rule 1: "For all L-M-H, if 'LO readimg
    material' is attempted first time &
    completed, it has some examples, selct an
    'example-lo'"

(defrule nxt-exmpl-mdm-high
;lo-readimg-material' attempted first time
(learning-object (lo-id ?thloid)
       (stud-id ?stud)
       (lo-name ?reading-material)
       (lo-type theory)
       (attempt-count ?x&:(eq ?x 1)))
(concept (concept-id ?cid))

;the LO has been completed
(learning-object (reqd-time ?time))
(learning-object (elapsed-time ?actual-time&:
       (>= ?actual-time ?time))
       (status ?s&completed))

;the LO has one or more incomplete examples
(concept (concept-id ?cid) (lo-ids ?exloid))
(learning-object (lo-id ?exloid)
       (lo-type example)
       (status ?s&~completed))
=>
(assert
(recommendation (recommended-activity example)
                     (recommended-lo ?exloid)
                     (because completed))))

Rule 2: "if 'lo-readimg-material' is attempted
     first time & completed, but it has NO
    example, then selct a 'question-lo'"

(defrule no-exmpl-aftr-th-mdm-high
; lo-readimg-material' being attempted first
    time
(learning-object (lo-id ?thloid)
                     (stud-id ?stud) (lo-name
                         ?reading-material)
                     (lo-type theory) (attempt
                         -count ?x&:(eq ?x 1))
                         )
(concept (concept-id ?cid))

; and the lo-readimg-material has been
    completed
(learning-object (reqd-time ?time))
(learning-object (elapsed-time ?actual-time&:
                     (>= ?actual-time ?time))
                     (status ?s&completed))

; and this LO has no examples
(not(concept (concept-id ?cid)
                 (lo-names example)))
=>
(assert (recommendation
              (recommended-activity question)
              (recommended-lo ?exloid)
              (because completed))))
```