Shikshak: An Architecture for an Intelligent Tutoring System

Sunandan Chakraborty[a], Devshri Roy[a], Anupam Basu[a]
[a]*Department of Computer Science & Engineering,*
Indian Institute of Technology, Khargapur, India
{sunandanchakraborty,droy.iit,anupambas}@gmail.com

Abstract: Low literacy rate, high dropout rates and some other lacking infrastructure in the field of education in India demanded an alternative learning environment. In this context Intelligent Tutoring System (ITS) can be a plausible solution. In this paper, we present an ITS called *Shikshak*. We describe the different modules of a system and their functionalities and discuss the adaptive features of the system. Finally, we demonstrate through some experiments the efficacy of the system in teaching children basic programming in C. The system showed some improvement in performance compared to the performance of students learning from standard classrooms.

Keywords: Intelligent Tutoring System, Curriculum Sequencing, Student Modeling

Introduction

One of the most important challenges faced by most of the developing and the underdeveloped countries is the spread of education among all. As an example, in India the literacy rate was about 65% and in the state of West Bengal it was 69.22% in 2001 [1]. A plausible reason behind this low literacy rate is attributed to the want of proper schools, proper infrastructure and poor teacher to student ratio as well as the resultant high drop out rate [2][3]. The said perspective vindicates the necessity of developing an alternate attractive, affordable and effective teaching platform that will capture the attention of children. Moreover, in traditional classroom, teaching follows a fixed method and pace. However, a heterogeneous group in such classrooms will have different requirements. Hence, such a method of teaching might not be optimized for the individual students.

Computer-based education tools such as Intelligent Tutoring Systems (ITS) can provide a fruitful solution to the above problem. An ITS can maintain a one-to-one relationship with the students and provides personalized and adaptive teaching methods. It can also to a large extent address the issue of unavailability of skilled teachers.

However, the existing ITSs are all meant for a particular domain [4] [5]. But in the present context we need an ITS which can be used for teaching in any domain to different categories of students. Furthermore, reviewing the state-of-the-art, we found some general drawbacks in the existing systems. Some of the drawbacks are enumerated below.

- Different courses from different domains cannot be represented using the same representation scheme [4][8][12].
- The process of adding new materials in the system is often complicated, to be used by not so computer savvy teachers.
- In most of the systems the student model considered all students as a homogenous entity [6][7][8].

- In some other cases student model is not transparent to the students. Comprehensive knowledge of one's performance is very vital for her progress.

In this paper, we present an ITS which is capable of handling the above tasks. In the following section, we discuss some related works. In section 3 we present an elaborate description of the system and discuss about some experiments performed on this system and their results in section 4. Finally, we conclude the paper in section 5. We have christened the present ITS as *Shikshak*, which in Sanskrit means 'The teacher'.

2. Related Work

In 1982, Sleeman and Brown reviewed the state of the art of computer aided instruction systems and first coined the term **Intelligent Tutoring Systems (ITS)** to describe a new generation of computer-based education tools. They defined ITS as being computer-based (1) problem-solving monitors, (2) coaches, (3) laboratory instructors, and (4) consultants. [9]. They used this term to distinguish the new systems from the existing Computer-Aided Instruction (CAI) tools. In these ITSs for the first time the use of Artificial Intelligence was observed. This made the systems "intelligent".

Thereafter many ITSs have been built to teach various subjects and topics. For example, Andes [10] was built to teach Newtonian Physics. A similar system is SQL-Tutor [11]. Often, more specific topics were targeted in ITSs. Like, CIRCSIM [12] taught medical students, Physiology.

Various methods were used in implementing these ITSs. Some of the most popular techniques were, Bayesian Networks [13] [7], Artificial Neural Networks [14], Fuzzy Logic [15], and machine learning techniques such as Reinforced Learning [16], Expectation Maximization [17] etc. Some of the systems used novel methods like dialogue management [18], using sensors to track the emotion of the learner [19].

3. Architecture of *Shikshak*

The overall structure of Shikshak is divided into three major modules – *Domain Model, Student Model and Control Engine.*

3.1 Domain Model

Domain model is the knowledge base of the system, where the domain knowledge is organized for all the subjects to be taught by the system. The Domain Model consists of a Domain Organization Module and a Repository.

3.1.1 Domain Organization Module(DOM)

Domain organization module is a structural representation of the different courses and their components stored in the system. Two major data structures are used for this purpose.

Course Tree (CT): CT is a hierarchical structure of a course stored in the system. The root of the tree keeps the name of the course, and is called course node. Subsequently, in the lower parts of the tree different sections/subsections and topics of the course are kept. The leaf nodes of the tree are called topics and are the atomic teachable units. Topic nodes are also associated with some features – *Concepts (sub-topics), prerequisite topics, threshold score, difficulty,* and *importance*. A sample CT is shown in Figure 1.

Topic Dependency Graph (TDG): The nodes of TDG are constituted of the topics from the corresponding CT, whereas the edges in TDG depict ‘prerequisite’ relation between the nodes (topics). TDG drawn from the CT in Figure 1 is shown in Figure 2.

Every course stored in the system will have its own pair of CT and TDG.

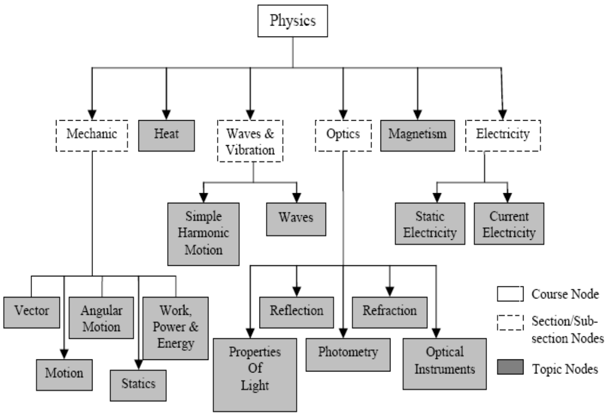

Figure 1. Course Tree for Physics

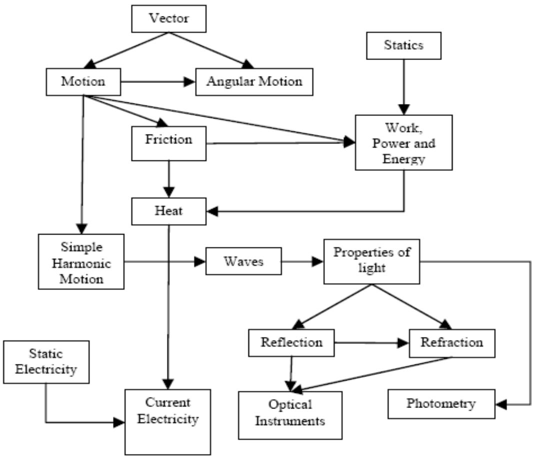

Figure 2. Sample TDG drawn from the CT in Figure 1

3.1.2 Repository

The Repository is a pool of learning and testing materials. For efficient access of materials from the repository the materials are tagged with various features. These features provide a graphic description of the documents, which helps the material selection agent (section 3.3) of the system in efficient and customized retrieval of the materials for the students. In order to tag the study materials a subset of the standard set of tags specified by IEEE LOM is used. In addition some extra features are added locally to incorporate some system specific information in the document description.

3.2 Student Model

The student model, that represents the students’ performance and learning pattern has been modeled in the present system as a Fuzzy State Transition Machine **<S, I, δ>**, where **S** is the set of states, **I** is the set of inputs and **δ** is the transition function. Each state is actually a composite state comprised of the **Coverage** state and the **Performance** state. The Coverage state is represented as the list of topics covered by the student in each course at any point of time. The Performance state consists of two components, **Comprehension-ability (C), Problem-solving skills (P).** Values of both the variables lie between **0** and **1**.

The performance of a student is computed by combining the individual values of the above parameters. The combined value is fuzzified using a set of fuzzy terms **{Excellent**, **Very Good**, **Good**, **Average, Poor}**. The fuzzified value represents the overall performance of the student. For a student, if after j^{th} topic the values of comprehension ability is C_j and problem-solving skill is P_j, then the fuzzy state will be calculated as [15][21], *Now, let* $v = \frac{(C_j + P_j)}{2}$

The membership functions of the fuzzy states are defined as:

$$\begin{aligned}
&\mu_{Excellent}(v) = 1 \ \ if \ \ 0.9 \le v \le 1\\
&\qquad\qquad\quad = 0 \ \ otherwise\\
&\mu_{VeryGood}(v) = v\\
&\mu_{Good}(v) = 2v \ \ if \ \ 0 \le v < 0.5\\
&\qquad\quad = (2\text{-}2v) \ \ if \ \ 0.5 \le v \le 1\\
&\mu_{Average}(v) = 1\text{-}v\\
&\mu_{Poor}(v) = 1 \ \ if \ \ 0 \le v \le 0.1\\
&\qquad\quad = 0 \ \ otherwise
\end{aligned} \tag{1}$$

The state with the highest membership will be taken as the performance state.

The results of the test taken by the students after each topic serve as the input to the state machine. Input value surpassing the threshold value of a topic indicates a successful completion of that topic. This leads to the augmentation of this topic in the *Coverage* part of the state. Again, the input value is combined with the previous performance state to compute the next performance state.

The input value is calculated from the test results in two phases. In the first phase the following two parameters are used,

- Correctness of the answer (c)
- Response Time (t) [15] [20]

Correctness lie between [0,1] and response time is the time taken to answer each question. For each question an intermediate value is computed using the following formula.

$$\begin{aligned}
\alpha &= c \text{ if } t \le t'\\
&= c \times \left(1 - \left(\frac{t - t'}{3t' - t'}\right)^2\right) \text{ if } t' < t \le 3t'\\
&= 0 \text{ if } t > 3t' \qquad \textit{where } t' \textit{ is the threshold time set for the question}
\end{aligned} \tag{2}$$

In the second phase α is further modified using two features of the question, namely,

- Relevance of the question with respect to the topic
- Difficulty of the question [15]

The final input value is calculated as,

$$i = \alpha^{\left(1 - \frac{diff}{2}\right)\left(1 - \frac{rel}{2}\right)} \qquad \textit{where } \boldsymbol{diff} \textit{ and } \boldsymbol{rel} \textit{ are real values lying in between } 0 \textit{ and } 1 \tag{3}$$

This grants the students with extra credits for answering a relatively difficult or relevant question. All the features used in this calculation, like, threshold time, relevance, difficulty are provided by the teacher. Other parameters (correctness, response time) are obtained from the student's response.

The state transitions are defined by the transition function δ:

δ*:(S 捏) → S*

Previous state and the input value define the next state. The current topic is added to the *Coverage* part. In order to calculate the *Performance* state, the updated values of comprehension ability and problem-solving skills are calculated first. Each question has a focus, which depicts whether the question focuses on C, P or both. The C and P values after the completion of topic *j* is computes as,

$$C_j = \frac{\sum_m i \times f_c}{m} \quad \textit{where } f_c = 1 \textit{ if focus of the question includes C },\ 0 \textit{ otherwise} \tag{4}$$

where, *i* is the input value and *m* is the no. of questions in the test. Similarly, $\mathbf{P_j}$ is calculated using $\boldsymbol{f_p}$. The C_j and P_j reflect the performance of the student in the current topic. Let, C_{j-1} and P_{j-1} be the <C, P> pair in the previous state. The modified <C, P> pair is computed as

$$\hat{C} = \frac{(C_{j-1} + C_j)}{2}, and \ \hat{P} = \frac{(P_{j-1} + P_j)}{2} \qquad (5)$$

The $\hat{C}$ and $\hat{P}$ is the <C, P> pair for the next state, depicting an average of the student's previous and current performances. The fuzzy state is calculated according to Eq. (1) presented earlier.

The Student Model also stores some additional information about the student. They are, *performance history, detailed test results, types of the documents preferred, languages preferred*.

3.3 Control Engine

Following a single teaching plan may not be successful for all the students. Hence, this system devises separate plans for each student. Moreover, based on the student's performance it is capable of re-planning the teaching method. Such planning and adaptation is performed by the Control Engine of the system. Depending on the data about students' state the Control Engine i) traverses the Topic Dependency Graph to select the most appropriate topic sequence, ii) chooses the most suitable set of contents of the chosen topic, from the Repository for a particular student, iii) collects and analyzes the feedbacks (test results) from the students and plans how to improve upon the drawbacks found from the feedbacks and maximizes their performance.

These tasks are performed by the three sub-modules of the Control Engine, they are – **Topic Planner (TP), Material Selection Module (MSM)** and **Result Analyzer (RA)** respectively.

The TP builds a path along the TDG starting from the initial topic to the final topic of a subject for each student separately. For example, from the TDG in Figure 3 four sequences can be drawn (without violating the prerequisite relation), which are shown in Figure 4. The topic planner has been modeled using a fuzzy-rule based system. At any point of time the system has to choose from a fixed set of topics whose prerequisites have been covered. For example, if we consider the TDG from Figure 3, after 'Properties of Light' has been covered the next topic can be either 'Reflection' or 'Photometry'. The selection among the possible alternatives is based on each student's cognitive ability and the selection might differ for different students. The selection is done by firing fuzzy rules [21] which compare the topics with the student's ability.

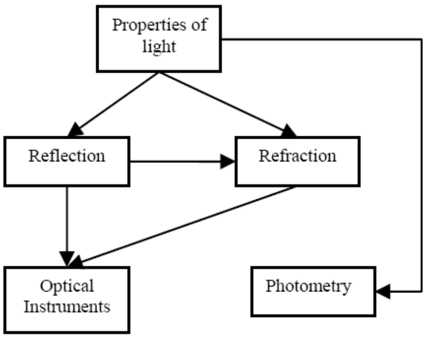

Figure 3. TDG for Optics
(a part of the TDG shown in Figure 2)

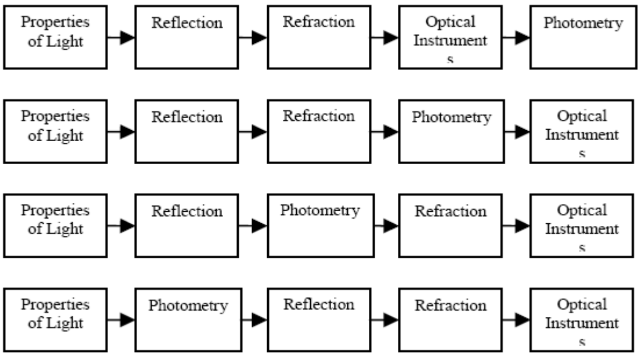

Figure 4. All possible sequences drawn from TDG in Figure 3

After a topic has been selected by the TP, the Material Selection Module (MSM) selects the proper materials to teach that topic. The MSM selects the type of material, such as explanation based, visual based, instructional etc. as is suited to the learning style of a particular student. In order select materials to teach topic **t**, MSM searches the repository and selects all the materials, which contains relevant information about **t**. In the second phase, previously constructed set (containing materials on **t**) is refined according to the student. Each document in the set is matched with the student and a score (in the range [0, 1]) is computed, where, 1 represents the most suitable document. Finally, all the materials, with a score higher than a threshold value are selected. Computation of the score is also done using fuzzy rules. The details of the fuzzy rules for TP and MSM the reader may refer to [22].

Result Analyzer (RA) collects the feedbacks (test results) from the students and analyzes them, to obtain vital information about a student's performance and scopes for further refinement in a student's teaching plan by re-planning the content selection using the MSM module or re-sequencing of the topics using TP, if she is unable to score a topic's threshold score. In addition, RA tries to find scopes for improvement in a student's performance, even if she had crossed the threshold score. Tests are so designed that from the results various data about a student's performance can be obtained. They can be analyzed to extract the student's drawbacks and means to improve upon them.

4 Results and observation

In order to evaluate the system a workshop was organized where school students used the system. The students' performances were mainly analyzed to estimate the efficacy of the system. Apart from that, some other points were also observed to evaluate the different inherent features of the system.

Thirty three students participated in the workshop and belonged to the age group of 14 to 16 years. Elementary C programming was selected as the topic to be taught using our ITS. It was observed that the students had no prior exposure to this subject and anything they learned about it was through the ITS. Thus, the students' performance depended solely on the ITS teaching and would reflect the efficiency or the drawbacks of learning from the ITS.

Foremost objective of evaluating an ITS is to compare the performance of teaching in traditional classrooms to the performance from teaching using an ITS. This will substantiate the success of the ITS if found to be producing improved performance. In order to check such results, the students were divided into two almost equal subgroups. Care was taken that the two groups contained students with similar cognitive ability to eliminate any bias. The first group used the ITS and followed an adaptive and personalized teaching, whereas the second group followed a fixed method of teaching in the classroom. Both the groups were taught the same topics. At the end of the course, both the groups were given the same set of problems to solve, from each topic they learnt. The problems were evaluated and the performances from both the groups were compared. The results obtained are shown in Figure 5. From the result it can be seen that apart from the first topic, students studied from the ITS outperformed the other group. In the first topic the relatively lower performance can be attributed to the fact that the students needed some time to get accustomed to a new learning environment.

In addition we analyzed the results obtained from the students studying from the ITS to get a more detailed evaluation of the ITS and its functionalities. First we checked the percentage of students whose performance improved along the course. And we found favorable results. The results are summarized below in Figure 6. Performance of majority of the students (85.19%) improved, after studying from the ITS. It is important to note that, in the first experiment (Figure 5) the results were based on externally provided problems and assignments. Here on (Figure 6 onwards), the results are based on the internal test kept within the system, which are evaluated implicitly by the system and stored in the student model.

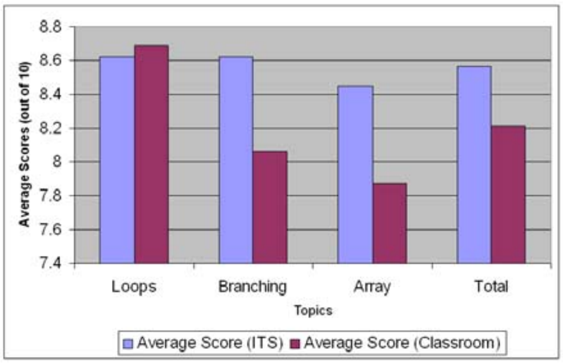

Figure 5. Comparison between ITS teaching and classroom teaching (left column shows ITS results and the right column classroom results)

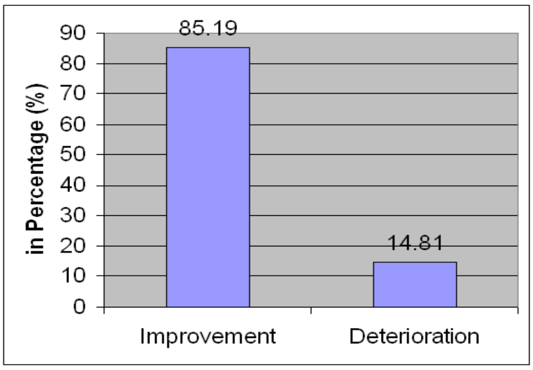

Figure 6. Net improvement and deterioration after studying from ITS (Performance compared between first and the last lesson of the course)

Finally, we demonstrate some inherent features of the system. Earlier, we claimed that Topic Planner devices individual topic sequences for the students. Here we present the all possible sequences drawn from the TDG in Figure 7 and the percentage of students following each the sequences. There were 3 possible sequences without violating the prerequisite relation. The "others" column depicts the various re-planned sequences for the students who needed repetition of the topics. Finally, we present the improvement in the individual comprehension-ability and problem-solving skills of the students. This results (Figure 8) show that the ITS has successfully worked on the improvement of these individual characteristics, leading to a general improvement in the performances.

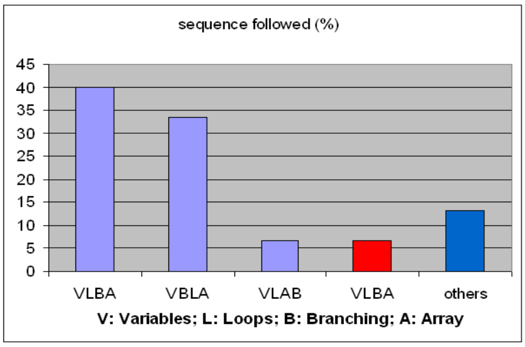

Figure 7. Percentage of student s following each Sequence

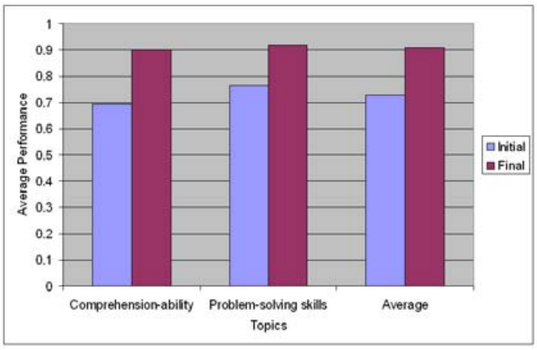

Figure 8. Improvement in individual student characteristics after studying from *Shikshak*

5 Conclusion

Shiskshak is an ITS that presents several novel features. These include the fuzzy student model and the fuzzy approach to Topic Planning and Material selection. Moreover, the Result Analyzer allows the system to adapt according to a student's performance.

The ITS has been tested using a field workshop and the efficacy of teaching the school children using the present ITS has been demonstrated. The results have demonstrated an overall 4% improvement in performance, compared to those who used classroom teaching alone. The system supports multilingual teaching, an aspect which has not been discussed in the present paper, due to paucity of space. However, due to this feature the present ITS is in the process of deployment in India, a multilingual country. Field testing has been carried out in urban schools. Presently, we are in the process of applying the ITS based approach in select rural schools, where both the number and quality of teachers are lagging behind those in the urban schools. Hence, it is expected that the performance would be further enhanced for the rural schools.

Acknowledgement

This project is funded by Media Lab Asia, New Delhi

References

[1] www.censusindia.net

[2] Majumdar, T. (2006) “Cost and financing of elementary education in West Bengal” in *The economics of elementary education in India.*. S. Mehrotra, Ed. Sage Publications, 2006, pp. 251-284.

[3] http://education.nic.in/stats/detail/18.pdf

[4] Gertner, A., VanLehn, K. (2000). “Andes: A coached problem solving environment for physics” in *Proc. 5th International Conference. Intelligent Tutoring Systems,* Berlin, 2000, pp. 133-142.

[5] Ramzan, K. A., Evens, M. W., Michael, J. A. Rovick, A. A. (2000) “Architecture of CIRCSIM-Tutor (v.3): A smart cardiovascular physiology tutor” in *Proceedings of the 7th Annual IEEE Computer-Based Medical Systems Symposium, Winston-Salem, NC, 1994*, pp. 158-163. IEEE Computer Society Press.

[6] Conati, C., Gertner, A., VanLehn, K., & Druzdzel, M. (2002). On-line student modeling for coached problem solving using Bayesian networks. In *Proc. Sixth International Conference on User Modeling,* Vienna , 1997, pp. 231-242. Springer

[7] Zapata-Rivera D. and Greer, J. (2004). “Interacting with inspectable Bayesian student models”. *International Journal of Artificial Intelligence in Education*, Vol 14, 2004, pp. 127 – 168.

[8] Baffes, P., and Mooney, R. (1996). A novel application of theory refinement to student modeling. In *Proceedings of American Association for Artificial Intelligence*, 403–408.

[9] Sleeman D. and Brown, J.S. (1982). “Introduction: Intelligent Tutoring Systems”. in *Intelligent Tutoring Systems*, D. Sleeman, J. S. Brown, Ed. Academic Press, 1982, pp. 1-11.

[10] Gertner, A., VanLehn, K. (2000). “Andes: A coached problem solving environment for physics” in *Proc. 5th International Conference. Intelligent Tutoring Systems,* Berlin, 2000, pp. 133-142.

[11] Mitrovic, A. (2000). “An intelligent SQL tutor on the web”. *International Journal of Artificial Intelligence in Education, 13*(2-4), 2003, pp. 171-195.

[12] Ramzan, K. A., Evens, M. W., Michael, J. A. and Rovick, A. A. (1994). “Architecture of CIRCSIM-Tutor (v.3): A smart cardiovascular physiology tutor” in *Proceedings of the 7th Annual IEEE Computer-Based Medical Systems Symposium, Winston-Salem, NC, 1994*, pp. 158-163. IEEE Computer Society Press.

[13] Conati, C., Gertner, A., VanLehn K., and Druzdzel, M. (1997). “On-line student modeling for coached problem solving using Bayesian networks” in *Proc. Sixth International Conference on User Modeling,* Vienna , 1997, pp. 231-242. Springer.

[14] Wang, T., Mitrovic, T. (2002). “Using neural networks to predict student's performance”. In *Proc. of International Conference on Computers in Education*, 2002, pp. 969-973.

[15] Weon, S., Kim J. (2001). “Learning achievement evaluation strategy using fuzzy membership function” i*n Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference,* Reno, NV, 2001, October pp. 10–13.

[16] Beck, J. E. (1997). “Modeling the student with reinforcement learning”. *Machine Learning for User Modeling Workshop at the Sixth International Conference on User Modeling*, 1997.

[17] Ferguson, K., Arroyo, I., Mahadevan, S., Woolf, B., and Barto, A. (2006). “Improving intelligent tutoring systems: Using expectation maximization to learn student skill levels” in Proc. *8th International Conference on Intelligent Tutoring Systems*, Taiwan, June, 2006, pp. 453-462.

[18] Evens, M. W., Brandle, S., Chang, R., Freedman, R., Glass, M., Lee, Y. H., Shim, L.S., Woo, C.W., Zhang, Y., Zhou, Y., Michael J.A., and Rovick A.A. (2001). “CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue” in Proc. *Twelfth Midwest AI and Cognitive Science Conference,* Oxford, 2001, pp. 16-23

[19] D’Mello, S. K., Craig, S. D., Gholson, B., Franklin, S., Picard, R. W., Graesser A. C. (2005), “Integrating affect sensors in an intelligent tutoring system” in *Affective Interactions: The Computer in the Affective Loop Workshop at International Conference on Intelligent User Interfaces 2005*, San Diego, 2005, pp. 7-13.

[20] Beck, J. E. (2004). “Using response times to model student disengagement” in *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, Maceio, Brazil, August, 2004, pp. 13-20.

[21] Ross, T. J. (1997). *Fuzzy Logic with Engineering Applications*. McGraw-Hill, 1997, pp. 134-146.

[22] Chakraborty, S., Bhattachary, T., Bhowmick, P., Basu, A. and Sarkar, S. (2007). Shikshak: An Intelligent Tutoring System Authoring Tool for Rural Education. Published in the Proceedings of IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD2007). Bangalore, India.