

# Shikshak: An Intelligent Tutoring System Authoring Tool for Rural Education

Sunandan Chakraborty, Tamali Bhattacharya, Plaban K. Bhowmick, Anupam Basu and Sudeshna Sarkar

**Abstract**—Low literacy scenario in India and other developing nation demands an alternative learning environment to deal with the problem. Lack of trained teachers, high dropout rates are some of the major problems that need to be addressed. Intelligent Tutoring System (ITS) or ITS Authoring tools (ITSAT) can be thought of as a possible solution to these problems. In this paper we present *Shikshak*, an ITSAT developed by us and discuss its deployment in the district of Paschim Medinipur, West Bengal along with its sample effect on primary education.

**Index Terms**—Adaptive Tutoring, Educational Technology, Intelligent Tutoring Systems, Student Modeling

## I. INTRODUCTION

**E**LECTRONIC media create opportunities for reshaping education in important ways. The technology allows for more focused learning. Use of multimedia opens up possibilities to offer alternate modes for students to receive lessons from even off site expert teachers. The new technology transmits the material faster and in forms that can be customized to each student’s learning needs. Through teaching and learning, the individual acquires and develops knowledge and skills. The use of computers in schools has evolved dramatically over the past 25 years. Information and communication technology (ICT) in education provides a wide range of tools for teaching and learning. The adoption of ICT by teachers implies the development of meaningful contexts in the classroom. The success of ICT integration requires collaborative participation of the teachers towards careful selection, planning and integration of suitable contents along with regular updates and modification. Our main objective was to build a system, which would be authorable by the local teachers and para-teachers as well as adaptive. An authorable system should make the system independent of the domains to be taught and also of the medium of instruction. This would also make the system participatory as different teachers would be able to organize the system, build different course structures, include course materials etc., according to the need of her students. Incorporation of adaptation mechanism would make the system robust. The other important feature of our system is its modularity. There are separate modules, which work independently but can communicate with each other to keep consistency and correct execution of the system. We call our system, *Shikshak*. ‘Shikshak’ in Sanskrit means, the

Sunandan Chakraborty is with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India-721302, E-mail: sunandanchakraborty@gmail.com

Tamali Bhattacharya, Plaban K. Bhowmick, Anupam Basu, and Sudeshna Sarkar are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India-721302.

TABLE I  
LITERACY RATE IN SOUTH ASIA

Country	Literacy Rate %	
	Adult	Youth
India	61.3	73.3
Nepal	44.0	62.7
Pakistan	41.5	53.9
Sri Lanka	92.1	97.0
Bangladesh	41.1	49.7

teacher. In the subsequent sections we discuss the motivation behind this work, a short description of some related systems, our system in detail and an instance where this work has been deployed and some experiments performed.

## II. MOTIVATION

In India the literacy rate was about 65% [1] and in the state of West Bengal it was 69.22% in 2001 [2]. It implies that about 350 people out of 1000 people in India and 308 out of every thousand in West Bengal state cannot read or write. The scenario in the whole South Asia region is not encouraging either. The figures are shown in Table I [3]. The main reasons behind this low literacy rate may be want of proper schools, proper infrastructure and poor teacher to student ratio. In Table II [4] [5] [6], we show the student-to-teacher ratio in different countries of South Asia. The student-to-teacher ratio in West Bengal was 67 in 2001-02 [7].

The effect is more pronounced in the fringes of urban societies and rural India as these regions are severely deprived in terms of access and facilities in education. The lack of proper and attractive teaching interfaces makes the process of learning monotonous and this enhances the drop-out rate among the rural and semi urban school going children. The drop-out rate in West Bengal was 80.24% among the group of students of class I-X in 2003-2004 [8].

The above perspective vindicates the necessity of developing an alternate attractive, affordable and effective teaching platform that will capture the attention of children. In this scenario an Intelligent Tutoring System (ITS) can be quite relevant. An ITS can to a large extent address the issue of unavailability of skilled teachers. However, the existing ITSs are all meant for a particular domain [9] [10]. But in the present context we need an ITS which can be used for teaching in any domain to different categories of students from different socio-economic segments and with heterogeneous family educational background. But most of the available systems are domain specific, and tuning them for teaching in another domain or for a different teaching strategy is a complex task

TABLE II  
STUDENT-TO-TEACHER RATIO IN DIFFERENT COUNTRIES IN SOUTH ASIA

Country	Student-to-Teacher ratio	
	Primary	Secondary
India	40.2	37.2[2003]
Nepal	35.4	28.0
Sri Lanka	23.4	22.0[2003]
Bangladesh	30.0	40.0

and needs the participation of the system developers. This led us to develop a system which would be flexible and usable for different domains and that would allow participatory updates. Consequently the goal is to develop not only an ITS but also an ITS Authoring Tool (ITSAT). ITSAT is a tool through which ITS can be configured or edited. It is a shell above the ITS layer which provides some flexibility and proper interfaces.

Moreover, there are some students with learning disabilities, and communication disorders such as autism, for whom classroom teaching is not very suitable. They can not participate properly in the class, causing hindrance to their learning. For these students the proposed ITS, which provides one-to-one self study mechanism, can be a good alternative. The performance of these students can thus increase.

### III. RELATED WORKS

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. [11]. 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 [9] was built to teach Newtonian Physics. A similar system is SQL-Tutor [12]. Often, more specific topics were targeted in ITSs. Like, VisMod was used to teach cell structures in Biology [14], CIRCSIM [10] taught medical students, Physiology.

Various methods were used in implementing these ITSs. Some of the most popular techniques were, Bayesian Networks [15], Artificial Neural Networks [13], Fuzzy Logic [16], and machine learning techniques such as Reinforced Learning [17], Expectation Maximization [18] etc. Some of the systems used novel methods like dialogue management [19], using sensors to track the emotion of the learner [20]. Some other notable ITSs are FlexiTrainer [21], VersaTutor [22], Ontology based ITS by Suraweera [23] etc.

Some common drawbacks were observed in the existing ITSs. One of them was inflexibility, i.e. once an ITS is built it can rarely be changed or modified without the system developer's intervention. This proved to be inconvenient in some cases. Thus a shell above a normal ITS was necessary for configuring and editing the system. This led into the emergence of a new category called **Intelligent Tutoring**

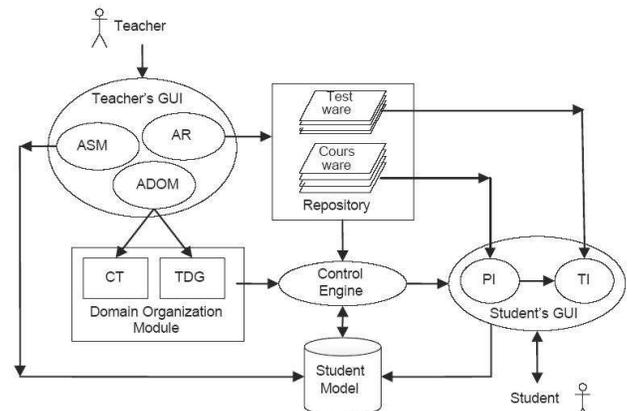
**System Authoring Tool (ITSAT)**. Some of the recently built ITSAT includes REDEEM, SIMQUEST, XAIDA [24] etc.

### IV. SYSTEM OVERVIEW

In this section, we discuss our system, Shikshak, in details. First we describe the system architecture. Next we illustrate the different modules, their purpose, functions, and goals etc.

#### A. System Architecture

A typical ITS has four basic modules, **domain model, student model, teaching model and user interfaces**. The domain model stores the course structure in an organized manner. A course can have various components, such as sections, sub-sections, topics etc. These components are kept in the domain model along with their dependencies. All the documents and materials required to teach a student are also stored in this module. The student model is the representation of the students the system is dealing with. The student model supplies the system with all the information it needs to adapt itself with the student. Thus, student model is an important tool for the adaptation procedure. The teaching module embodies all the decision-making process regarding course planning and adaptation. We also call this module the Control Engine, since this module controls the whole system, by taking inputs from the other components. Finally, the user interfaces have two parts - one for the teacher and the other for the student. Teacher's interface is used to configure and modify the system and its various components. So, the teacher's interface acts as the authoring tool. It is a shell above the main ITS. Through this interface, the teacher can enter new courses, modify the existing ones, edit teaching methods etc. The student's interface is used to deliver all the teaching instructions. The type and the nature of these instructions would vary with student's capability and performance level. The overall system architecture is shown in Fig. 1. In the remaining part of this section we discuss the various modules and their functionalities.



ASM: Authoring student model, AR: authoring repository, ADOM: Authoring Domain Organization Module, CT: Content Tree, TDG: Topic dependency graph

Fig. 1. Overall System Architecture of Shikshak

## B. Domain Model

The domain model deals with the course, its structure and various components. There are two major parts in this module. The first one is concerned with the structure and organization of the course and its topics. The second part handles the actual learning materials, contents etc. The former is the **Domain Organization Model** and the latter is called the **Repository**.

The **Domain Organization Model (DOM)** consists of two basic structures. **Course Tree (CT)** and **Topic Dependency Graph (TDG)**. Each course is stored in the system in a tree structure, called the Content Tree. In CT the root node is the name of the course. In the following levels, different sections/subsections of the course are kept. Finally at the leaf level, actual teaching units are stored. These nodes are called **Topics**. Each edge of the tree represents a 'part-of' relation between the children and their parent. In Fig. 2 a content tree for high school physics is shown. When all the leaf nodes or the topics are learnt, it is assumed that the course is also learnt by the student. A topic is considered to be learnt if the student crosses the **Threshold Score** in the test. The threshold score that is set varies with the student. Students with higher capabilities have higher threshold score set. This helps the students reach their own potentials.

The **Topic Dependency Graph** represents the dependencies among the different topics of a course. TDG is a directed graph. The nodes in the TDG are the topics from the corresponding CT, and the directed edges between the nodes in TDG depict a prerequisite-of relation between them. An edge from node A to node B implies that, topic A needs to be covered before topic B. The teacher constructs the CT directly during the course authoring process, where the prerequisites of all the topics are also mentioned. Using this information, the TDG is constructed by the system. Thus, both CT and TDG collectively represent a particular course, its components and interdependencies. Each and every course constructed in the system has its own pair of CT and TDG. TDG built from the CT of Fig. 2 is shown in Fig. 3.

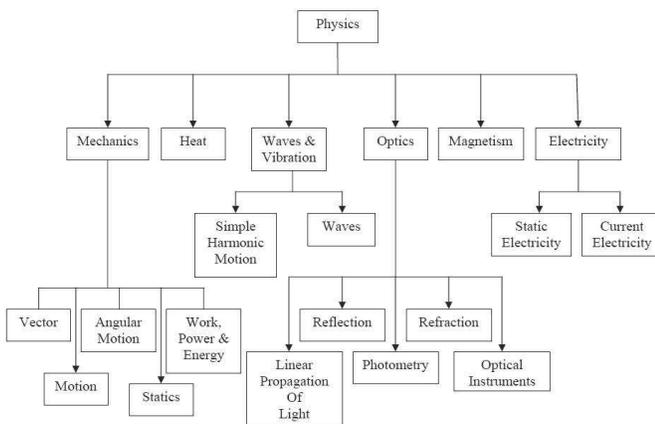


Fig. 2. Course Tree (CT) for High School Physics (Threshold score not shown)

The second part of the domain model is the repository. **Repository** is a collection of materials or documents for learning or teaching purpose. Our system supports various

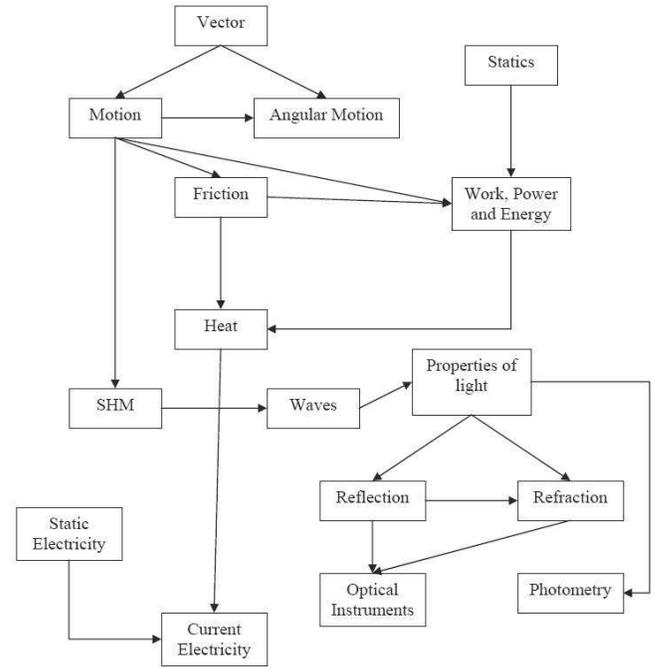


Fig. 3. TDG constructed from the CT in Fig. 2

kinds of materials, like text, audio, video, presentations etc. The teacher can create their own materials or incorporate existing materials from outside, even from the World Wide Web. The collection is stored in a structured way, with proper annotation. The set of tags which were used to annotate the materials are shown in Table III [25] along with their meaning and possible values. This helps the control engine to select the proper materials from the pool of documents. Although the Repository and CT/TDG are independent entities but they have a relation. Each topic in the CT/TDG has numerous documents associated with it. This means during a learning session when a topic is chosen, the actual teaching would be done by selecting the documents associated with that topic. Again a single document can have more than one topics associated with it. Apart from these, the repository also contains test materials. Test materials are required for evaluating the student. This is important because it is only through these tests the system gets an updated knowledge about the student.

## C. Student Model

We adopt a state based approach to model a student. Though, there are several parameters for pedagogical modeling of a student during a learning process, in this work we consider two parameters,

- Coverage: The topics covered by a student ;
- Performance of a student, measured through her ability to comprehend and her problem solving skills.

Consequently, the state of a student is a composite state  $S = S1 \times S2$ , where  $S1$  denotes the set of coverage-states and  $S2$  denotes the set of performance-states. The student model is defined as a state-transition system with a three tuple  $\langle S, I, \delta \rangle$ , where,

TABLE III  
FEATURE SET DESCRIBING THE DOCUMENTS IN THE REPOSITORY

Feature	Purpose	Possible values
Target	Meant for which section of student	Range of $\langle C_i, P_i \rangle^a$ of SM
Hardness	Hardness of the document	$[0,1]$
Type	The type of learning resource	<i>Explanation, application, exercise or experiment</i>
Level	The grade level for which it is meant	Grade
Focus	The document meant to enhance $C_i$ or $P_i$	$C_i/P_i^a$
Format	Specifies the format of the material, which also includes the medium	<i>Text, picture, audio, video, presentation, animation</i> along with the file format
Concept	For which topics this document is meant for <sup>b</sup>	Topic names.
		For each topic: Relevance of that topic in that document
		Is that topic defined here or used. ( <i>yes/no</i> )
Keywords	Keywords can help to select a document properly	List of keywords and their frequency

<sup>a</sup> $C_i$  and  $P_i$  are two attributes taken from the student model (*see sec IVC*)

<sup>b</sup>If a material is under more than one topic, it will occur in each of those topic's M set

1. S is the set of composite states,  $S = \{S_i\}$ , where each composite state  $S_i$  is a tuple  $\langle \text{Performance}_i, \text{Coverage}_i \rangle$

$\text{Coverage}_i$  is the list of **topics** that a student has covered successfully. As mentioned earlier, the successful coverage of a topic implies that the student has scored above the set threshold in the test conducted.

In order to represent a student, we propose a fuzzy state transition based student model, where the performance-state of a student is represented in linguistic terms, represented by fuzzy sets.

$\text{Performance}_i$  is denoted as a fuzzy term as defined below.  
 $\text{Performance}_i = \{\text{Excellent}, \text{Very Good}, \text{Good}, \text{Average}, \text{Poor}\}$

2. I is the input set, where  $i \in I$  is a real number.
3.  $\delta$  is the state transition function.

In the following, unless necessary we shall use the term state to mean composite-state.

**Computation of Fuzzy State:** For a student k, all the state information is stored in the student profile. At any "Present State" a student *has already covered* some of the topics successfully and *has demonstrated* a particular level of performance, which might have been modified through *her last test*.

Computation of  $\text{Coverage}_k$  is simple as it is augmentation to a list. The list is augmented when a student clears the test corresponding to a topic with a score greater than the set threshold score.

For computation of  $\text{Performance}_k$ , presently we are using only two aspects, namely *Comprehension-ability* ( $C^k$ ) and *Problem-solving skills* ( $P^k$ ). The values of  $C^k$  and  $P^k$  lie in the range  $[0, 1]$ . These values are computed every time a student appears in a test on a topic. After each update of the  $\langle C^k, P^k \rangle$  pair for a student in her student profile, the

fuzzy state is computed as follows:

Assume that, after the student k has covered the  $j^{\text{th}}$  topic her value of comprehension-ability and problem-solving skills have been  $C_j^k$  and  $P_j^k$  respectively. Now let,

$$v = \frac{(C_j^k + P_j^k)}{2} \quad (1)$$

The membership functions of the fuzzy states are defined as:

$$\begin{aligned} \mu_{\text{Excellent}}(v) &= 1 & \text{if } 0.9 \leq v \leq 1 \\ &= 0 & \text{otherwise} \end{aligned}$$

$$\mu_{\text{VeryGood}}(v) = v$$

$$\begin{aligned} \mu_{\text{Good}}(v) &= 2v & \text{if } 0 \leq v < 0.5 \\ &= (2 - 2v) & \text{if } 0.5 \leq v \leq 1 \end{aligned}$$

$$\mu_{\text{Average}}(v) = 1 - v \quad (2)$$

$$\begin{aligned} \mu_{\text{Poor}}(v) &= 1 & \text{if } 0 \leq v \leq 0.1 \\ &= 0 & \text{otherwise} \end{aligned}$$

The fuzzy performance-state of the student is determined by the maximum of these membership values. In case of a tie, the higher state is assigned.

For example, if at some point of time for student k  $C_j = 0.8$  and  $P_j = 0.3$ .

Then the value of  $v = 0.55$  [Eq. 1]

The individual membership values of each state will be [Eq. 2]:

*Excellent:* 0  
*Very Good:* 0.55  
*Good:* 0.9  
*Average:* 0.45  
*Poor:* 0

*Good* has the highest membership value. So, the pair  $\langle C_j = 0.8, P_j = 0.3 \rangle$  will map to the state *Good*. In the student profile, the individual values of  $C_j$  and  $P_j$  are also stored, along with the fuzzy state.

The computations of the  $C^k$  and  $P^k$  values are shown later.

**Computation of the Input:** The input value ( $i$ ) to the state transition of the composite state machine is the quantitative value with respect to the student's performance in a topic. This value is computed in two phases. In the first phase, the following two parameters are used.

- Correctness of the answer (c)
- Response Time (t) [26] [16]

The correctness value is presently a binary value, since the questions are of objective nature. For the response time (which denotes the time a student takes to answer a question), the teacher puts a threshold time ( $t'$ ) for each question. If the

student answers correctly within  $t'$  then she gets full credit. Then from  $t'$  to  $3t'$  the value diminishes according to the curve in Fig. 4. Finally, after  $3t'$  the credit becomes 0. The first phase score, for each question, is calculated using the following formula.

$$\alpha = \begin{cases} c & \text{if } t \leq t' \\ c \times (1 - (\frac{t-t'}{3t'-t'})^2) & \text{if } t' < t \leq 3t' \\ 0 & \text{if } t > 3t' \end{cases} \quad (3)$$

where,  $c$  is the correctness of the answer, having value 1 for a correct answer, 0 otherwise,  $t$  is the response time, and  $t'$  is the threshold time.

In the second phase, the first phase score is further modified with the following two factors.

- Relevance of the question with respect to the topic
- Hardness/difficulty of the question [16]

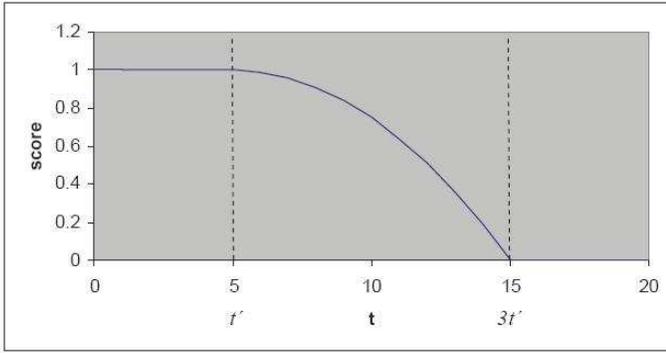


Fig. 4. Variation of score with time

The relevance and hardness values of the questions are set by the teachers during the authoring phase. The students who have answered difficult or relevant questions correctly are credited. The final score for a question is calculated using the relevance and the difficulty factors, as shown below:

$$i = \alpha^{(1 - \frac{diff}{2})(1 - \frac{rel}{2})} \quad (4)$$

where,  $i$  is the input value to the state transition machine (reflecting the net performance of the student),  $diff$  is the degree of difficulty and  $rel$  is the relevance of the question, both of which are in the range  $[0,1]$ .

**Transitions:** The state transitions are defined by the transition function  $\delta$ :

$$\delta: (S \times I) \rightarrow S \quad (5)$$

where,  $S$  is the set of composite-states and  $I$  is the set of inputs. Each transition leads to a “Next State” from a “Present State”.

**Computation of the Next State:** It may be noted that in a state, a student has a list of topics covered. Based on the performance and coverage of the student, a **Topic Planner** (discussed in the next section) assigns a new topic to be taught, in accordance to the **TDG** discussed earlier. Associated with the new topic that is assigned by the Topic Planner, a threshold

score ( $i_{threshold}$ ) is assigned. The state transition occurs only if the input value  $i \geq i_{threshold}$ .

The Next State,  $S_{i+1}$  is computed as follows.

Coverage $_{i+1}$  is augmentation of the list with the last topic completed successfully while migrating from state  $S_j$ .

Performance $_{i+1}$  is computed based on the performance level of the student  $k$  in the last state and her performance in the last test, as follows.

**Computation of  $C^k$  and  $P^k$ :** Each question in a test has a focus. It focuses either on C or P or both. The focus is determined by the teacher. Now from the input value and the focus of the question, the C and P values are found in the following way:

On successful completion of a topic  $j$

$$C_j = \frac{\sum_m i \times f_c}{m} \quad (6)$$

where  $f_c$  is 1 if focus of the question includes C and 0 otherwise.

$$P_j = \frac{\sum_m i \times f_p}{m} \quad (7)$$

where  $f_p$  is 1 if focus of the question includes P and 0 otherwise.

Here,  $i$  is the input value and  $m$  is the no. of questions in the test. 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  $\langle C, P \rangle$  pair in the previous state. The modified  $\langle C, P \rangle$  pair is computed as:

$$\hat{C} = \frac{(C_{j-1} + C_j)}{2} \quad (8)$$

$$\hat{P} = \frac{(P_{j-1} + P_j)}{2} \quad (9)$$

The  $\hat{C}$  and  $\hat{P}$  is the  $\langle C, P \rangle$  pair for the next state, depicting an average of the student’s previous and current performances. The fuzzy state is calculated according to Eq. 2 presented earlier.

**Example:** If the last performance-state was *Good* with (0.8, 0.3) as the  $\langle C, P \rangle$  value and the  $\langle C, P \rangle$  value from the input causing the transition is calculated as (0.7, 0.9), the  $\langle C, P \rangle$  value in the next performance-state will be:

$$\hat{C} = \frac{(C_{j-1} + C_j)}{2} = \frac{(0.8 + 0.7)}{2} = 0.75 \quad (10)$$

$$\hat{P} = \frac{(P_{j-1} + P_j)}{2} = \frac{(0.3 + 0.9)}{2} = 0.6 \quad (11)$$

Thus, from Eq. 1 the value of  $v = 0.675$

The individual membership values of each state will be [Eq. 2]:

*Excellent:* 0

*Very Good:* 0.675

*Good:* 0.65

*Average:* 0.325

*Poor:* 0

Therefore, the next state will be *Very Good*. The change from *Good* to *Very Good* indicates that the student has shown improvement from her last performance. This is taken as the

overall performance of the student so far and will be used by the control engine for planning purpose (described in the next section).

Recalling that the state of a student is a composite state, the new composite state arrived through a transition may have changes in the coverage-state or the performance-state or both.

In the student model, a copy of the TDG is kept for each student. In that local copy, the nodes of the graph are annotated by the performance-state, showing how the student performed when she studied that topic. In Fig. 5, an example of such an annotated TDG is shown. This TDG is a part of the TDG shown in Fig. 3.

Next, we discuss the advantages of this student model. Firstly, the use of linguistic terms such as *Excellent*, *Very Good* etc. makes it simple and natural to understand. The annotated TDG described above stores the performance history. It shows the individual performances of a student in each of the topics. Again, other information like, how the student performed in the prerequisites to a particular topic can be known. For example, 'Refraction' has two prerequisites, 'Properties of Light' and 'Reflection'. From this structure it can easily be found out, how the student performed in them. Finally, in the state transition machine each transition signifies a change in the student's performance.

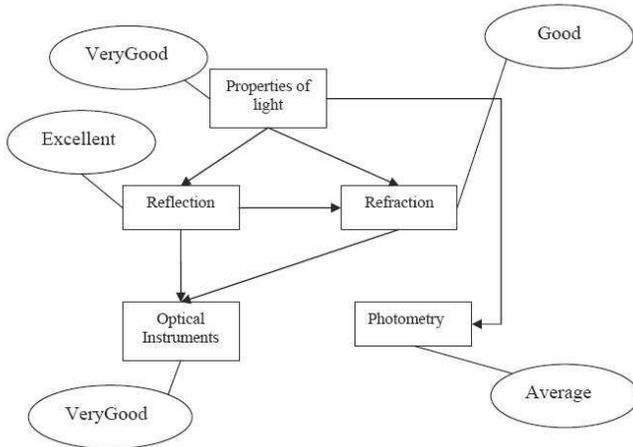


Fig. 5. Annotated TDG in Student Model

#### D. Control Engine

One of the most important parts of an ITS is this pedagogical planner. The primary task of this module is to plan a sequence of teaching actions to be taken during a teaching process. These actions and their sequence should match with the student's capability, requirement and goals.

The planning is done at two levels. At the upper level, sequence of the topics for the student needs to be planned. This part is called the **Topic Planner**. The topic planner starts from the initial state and terminates when all the topics are included in the plan or the sequence. At the lower level, after a topic is chosen another planning is necessary to compute the exact method of teaching that topic. This involves selecting the correct type of document and also the correct medium.

This part is known as **Material Selection Module**. For both the levels, if a particular decision does not seem to work (the student cannot achieve her potential), the control engine should re-plan that part. This defines the adaptation policy of the system. In this section we discuss the Topic Planner and the Material Selection Module. Then we describe in brief the adaptation policy of this system.

1) *Topic planner*: The task of the topic planner is to build a path along the TDG starting from the initial topic to the final topic of a subject. In other words, it devises a sequence of topics for each student separately. The task of this planner is to compute the most suitable sequence for a student. For example, from the TDG in Fig. 6 four sequences can be drawn, which are shown in Fig. 7. Each sequence should be consistent, i.e. no topic should come before any of its prerequisite topics.

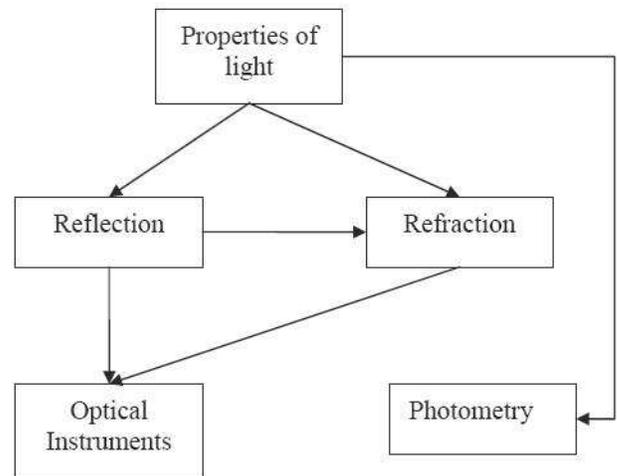


Fig. 6. TDG for Optics (a part of the TDG shown in Fig. 3)

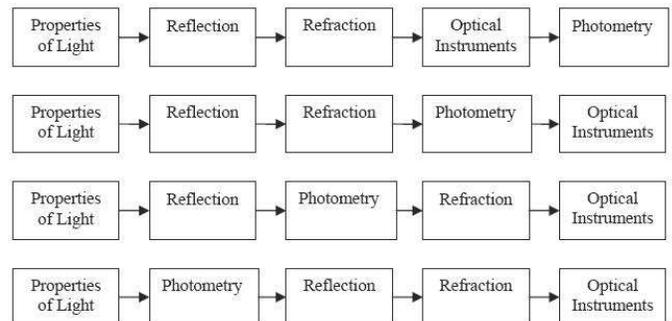


Fig. 7. All possible sequences drawn from the TDG in Fig. 6

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 Fig. 6, after 'Properties of Light' has been covered the next topic can be either 'Reflection' or 'Photometry'. So, the system needs to choose either 'Reflection' or 'Photometry' to add to the sequence. To take such a decision, the control engine derives

a suitability value of a topic considering two parameters, the nature of the topic and the nature of the student. The topic with the highest suitability value is added to the sequence. The topic related information is gathered from the domain model, whereas, the student model provides information about the student. The different topic and student related information, their meanings and the values are shown in Table IV. Each of the terms is a separate fuzzy variable, whose values are defined using specific fuzzy sets.

TABLE IV  
ANTECEDENTS OF THE FUZZY-RULES

Topic Attributes		
Name	Values	Meaning
Hardness	{Tough, Moderate, Simple}	Hardness of the material
Importance	{VeryHigh, High, Medium, Low, VeryLow}	How important this topic is in this course
Scope	{Excellent, VeryGood, Good, Average, Poor}	How many opportunities or avenues does this topic open up in the course
Student Attributes		
Performance	{Excellent, VeryGood, Good, Average, Poor}	Overall performance of the student in this course upto this point
Prerequisite Performance	{Excellent, VeryGood, Good, Average, Poor}	Performance in the prerequisite topics of this topic
Interest	{VeryHigh, High, Medium, Low, VeryLow}	How much interest the student has in this topic

These values constitute the antecedent of the fuzzy rules. The rules derive the suitability value. Fuzzy term ‘Suitability’ has fuzzy values “Very\_High”, “High”, “Medium”, “Low” and “Very\_Low”. The ‘Suitability’ value is defuzzified using *centroid* defuzzification method [27]. These rules are fired with all the available topics and individual Suitability values are obtained. Finally, the topic with the highest Suitability crisp value is chosen as the next topic. A typical rule looks like:

**IF** Performance is *Excellent* **AND** Prerequisite\_Performance is *Excellent* **AND** Interest is *VeryHigh* **AND** Hardness is *Simple* **AND** Importance is *VeryHigh* **AND** Scope is *Excellent* **THEN** Suitability is *Very\_High*

The rules are constructed by experienced teachers using the teacher’s interface. The rule base contains various such rules. During a decision-making process the inputs are fired with each and every rule in the rule base to get the individual outputs. Then the outputs are accumulated to take a decision.

2) *Material Selection and Adaptation*: After a topic has been selected by the Topic Planner, the next task is to select

the proper material to deliver the contents. Teaching involves selecting the correct sequence of suitable materials to deliver the contents. Each material is characterized by several features, shown in Table III. The material selection module identifies the contents having the correct type and medium, according to the student’s needs. The Student Model provides the necessary information for these operations. Now, if a particular type of contents cannot help the student to reach her goal, this module identifies that and chooses an alternative medium of instruction or try a new category of contents. The material selection is also achieved by a fuzzy rule based system.

The threshold value set with every topic varies with students, depicting different capabilities of different student. Now, if a student fails to achieve her own threshold, the system needs to reconsider the plan. Initially, the system would try alternative documents having different approaches or different media, i.e. try to vary the material. This is equivalent to loop back to the material selection module. If failure persists, then topic level re-planning is necessary, when the system will leave the current topic temporarily and try out a new topic. Through this iterative mechanism the system attempts to help the students reach their potential.

#### E. User Interfaces

Interfaces are an important part of the system. There are two categories of users, teachers and the students. Separate interfaces are provided for their interactions with the system. The teachers interface is the shell above the ITS for configuration and modification of the system. The teacher’s interface or the authoring interface has three parts, used to configure the different parts of the system, one to configure the Student Model, one for authoring the Domain Organization Model and the third for maintaining the Repository (see Fig. 1). Through these interfaces a teacher can configure various aspects of the system, like CT, TDG, the fuzzy-rule base, initial information about the student etc. Thus, this interface provides the system with the required flexibility and robustness. Also, due to this interface the system can become domain independent. A screenshot of the teacher’s interface is shown in Fig. 8.

Student interface is the front-end for the student to interact with the system. The interface has a bidirectional communication mechanism (see Fig. 1). The system presents all the learning documents and test materials to the student through this interface. Performance of the student in the tests is conveyed back to the system, specifically to the student model by it. This feedback is vital because the adaptation process would depend on this. So, the success of adaptive planning depends on it and its communication with control engine. Apart from these, the student interface also provides various courses and performance related information to the student, such as the performance history of the student, how much of the course has been covered, detailed test results etc. These help a student to identify and rectify her shortcomings.

## V. RESULTS AND OBSERVATIONS

The system presented has been deployed at a few schools in rural West Bengal, with educational contents for the primary

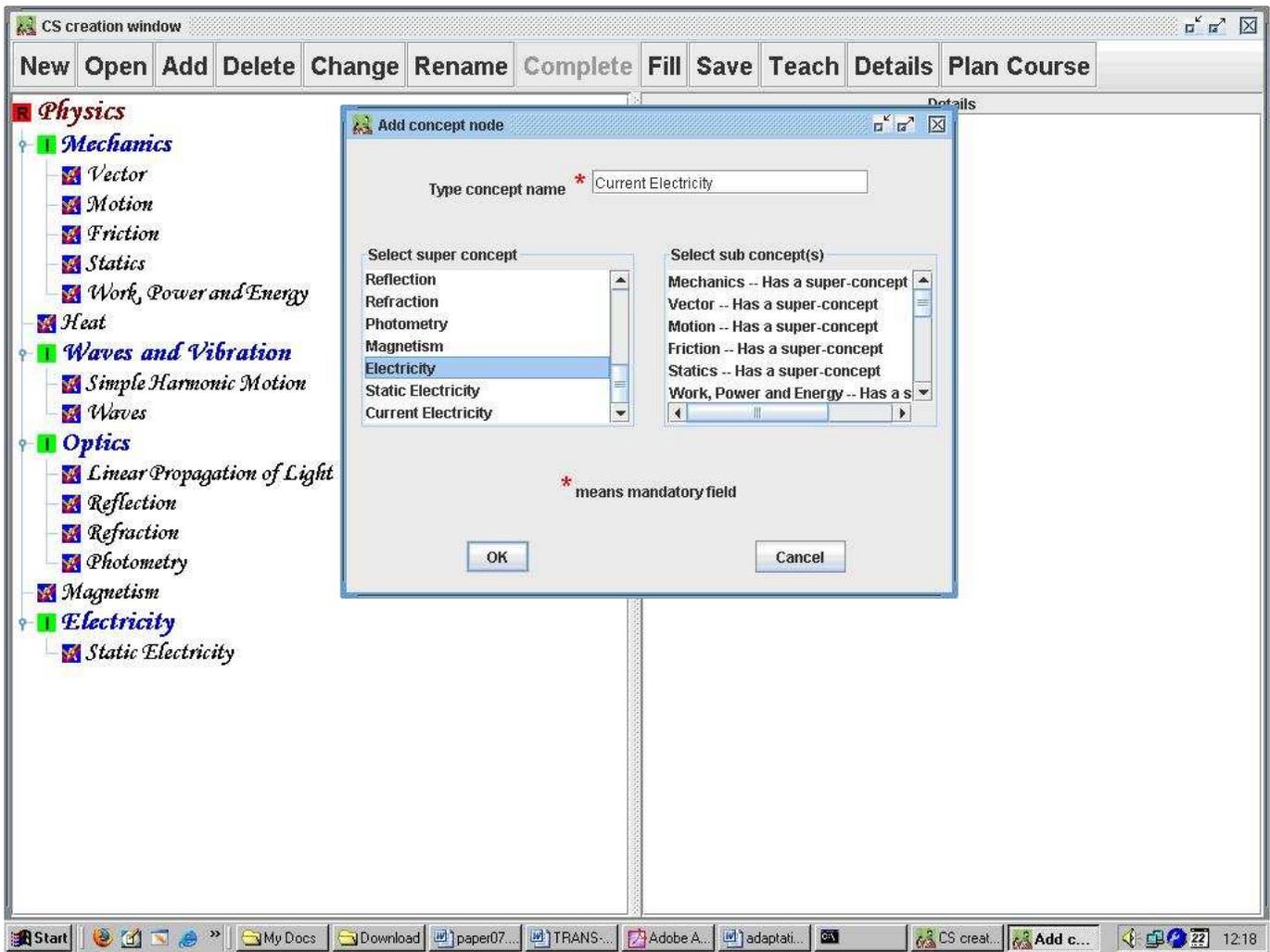


Fig. 8. Screenshot showing the interface, used to build the CT

level (Grades I to IV). For evaluation of the system we are carrying out experiments at some of these schools. To illustrate the efficacy of the system, we present here some results from the experiments conducted at a rural school, called Disha, in Hijli, Paschim Medinipur, West Bengal. The experiment was performed on 56 students of grade III. Three subjects were taught during the experiment; Science, Geography and History. A part of the **Content Tree** for Science, used in the experiment is shown in Fig. 9. And the corresponding TDG, drawn by the system is shown in Fig. 10. TDG contains the topics or the leaf nodes of the CT. The contents were in Bengali, i.e. the vernacular of the students, as the medium of instruction.

First, the students were taught some topics from History, Geography and Science in standard classrooms and were evaluated through some tests. Then the same students were taught the different topics of the same subjects using our system. Care was taken to see that the degree of hardness of the topics were same as before. Again they were evaluated through a separate on-line test of the same degree of difficulty as the previous one. We found a marked improvement in the average performance of the students. The results and observations are summed up in the following figures and tables. We found an

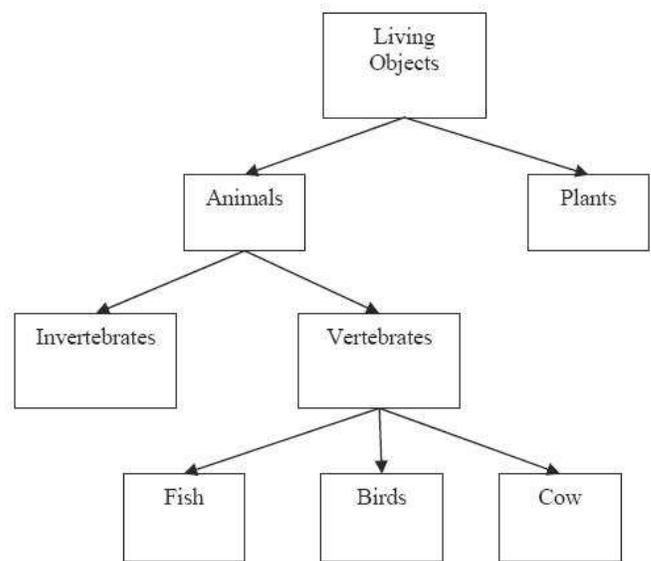


Fig. 9. A part of CT for science used in our experiment

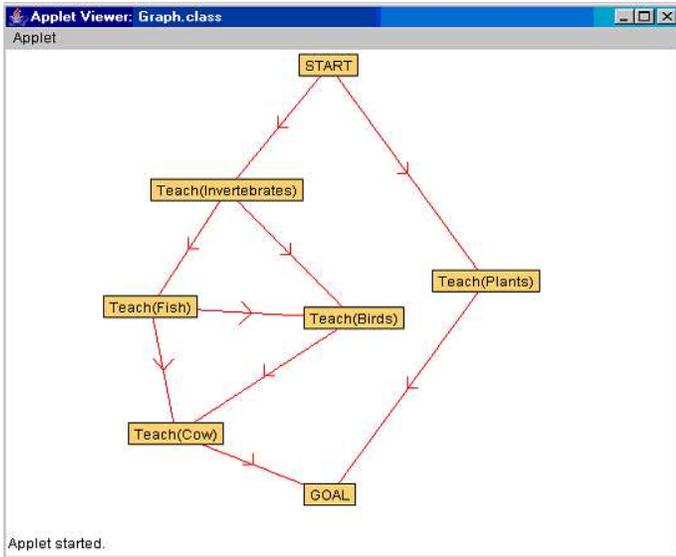


Fig. 10. TDG drawn by the system from the CT in Fig. 9

improvement of **7.5%** in the test results when our system was used to teach these students. The comparison of the average performance before and after the use of Shikshak is shown Fig. 11.

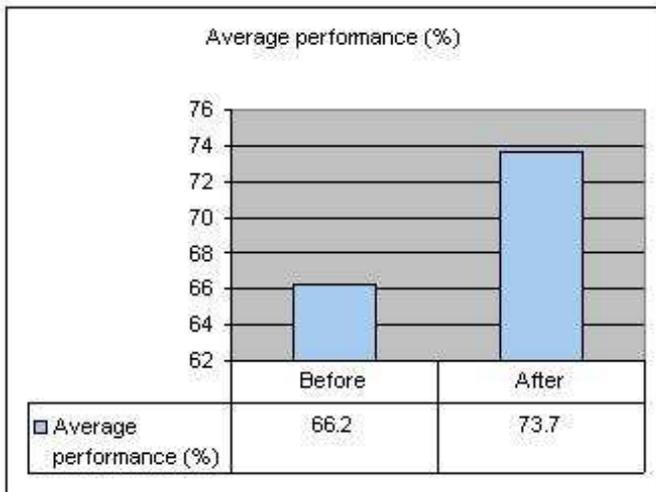


Fig. 11. Average performance (Before and After)

We also observed some notable features. Only **17.8%** of the total student’s performance decreased during this experiment, i.e. **82.2%** showed improvement, after using our system. The result is summarized in Fig. 12.

The whole class was divided into three categories, fast learners, medium achievers and slow achievers. We monitored the performance variations of these categories separately. Our findings are enlisted in Table V. The data show that even the slow and medium achievers are getting some advantages from the use of our system.

Finally, we tried to conduct a study about which medium of material is suitable for which type of students. Three kinds of electronic contents were used for teaching.

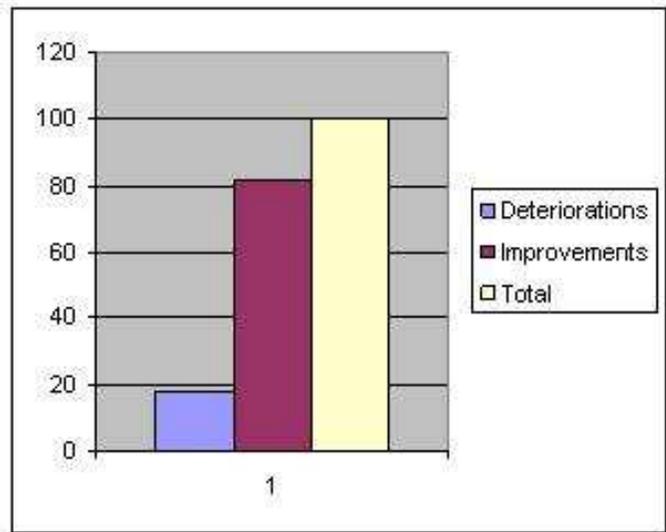


Fig. 12. Net improvement and deterioration, after Shikshak was used

TABLE V  
CATEGORY-WISE IMPROVEMENT OF THE STUDENTS AFTER USING SHIKSHAK

Category	No. of students with improved performance (%)
Fast	100
Medium	55
Slow	67

1. Text documents
2. Power Point Presentations
3. Flash Presentations (with animation).

First of all, it was found that text document could not hold their attention. In the case of PowerPoint and Flash presentations with animation, some interesting results emerged. In the case of High Achiever category, their performance was near about the same after seeing both kinds of presentation. In the case of Slow and Medium Achiever category, it was found that, their performance showed a mixed response to the PowerPoint presentations compared to a significant improvement in performance, after seeing the Flash presentations. From the above observations, the following inferences can be drawn,

- Animations may distract the students with higher capability. For them plain text format might be more appropriate.
- For the students, belonging to the other categories, contents with animations and visual explanations can be more effective. These formats may attract them more to the learning process.

Thus, we can get an idea about which types of contents are preferable to which categories of student. Fig. 13 shows a snapshot taken during our experiment.

## VI. CONCLUSION

In the previous section, we saw an instance where our system was deployed at a rural school. From the results of the experiments performed, we observed, that our system is showing positive outcome. Thus, our system can be effective in the field of child education, particularly in the rural areas.



Fig. 13. Students using Shikshak at Disha

Though we could grossly see the effectiveness of the system, we need to carry out more extensive field trials and experiments, under more controlled environments. For example, in the experiment presented here the same set of students was presented with different modes of teaching. This should have been done on separate cluster of students of the same capability mix. The variation of parental education and social background may reveal more interesting features.

The system is still in its development phase. While the student model and the topic planner have been automated, the material selection mode is functional but requires addressing more implementation issues.

The expected effectiveness of the system is based on some assumptions. We assume that the infrastructure in the rural areas is sufficient to install and execute our system. From the present facts, we see that about 86.77% villages in India have electricity, in West Bengal the figure is 83.63% [28], the most vital precondition for our system. Again, an all India programme called 'Sarva Shiksha Abhiyan' [29], which looks after infrastructure building for education in various parts of the country can further promote the use of our system. This programme can help in building the infrastructure, necessary for our system. Thus, our system can play a major role in the field of education or illiteracy eradication.

#### ACKNOWLEDGMENT

This project is funded by Media Lab Asia, New Delhi. Sunandan Chakraborty is supported by the Ministry of Social Justice and Empowerment, Government of India.

#### REFERENCES

- [1] [www.censusindia.net/t\\_00\\_006.html](http://www.censusindia.net/t_00_006.html)
- [2] [www.censusindia.net/profiles/wbe.html](http://www.censusindia.net/profiles/wbe.html)
- [3] *Economic Survey 2004-05*, Economic Division, Ministry of Finance, Government of India, quoting UNDP Human Development Report 2004.
- [4] [www.uis.unesco.org/template/html/Exceltables/WEI2005/Table2.8.xls](http://www.uis.unesco.org/template/html/Exceltables/WEI2005/Table2.8.xls)
- [5] [www.moedu.gov.bd/edu\\_system\\_edu\\_commission\\_2003.htm](http://www.moedu.gov.bd/edu_system_edu_commission_2003.htm)
- [6] [www.nepalhomepage.com/general/people.html](http://www.nepalhomepage.com/general/people.html)
- [7] T. Majumdar, "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.
- [8] <http://education.nic.in/stats/detail/18.pdf>
- [9] A. Gertner, K. VanLehn, "Andes: A coached problem solving environment for physics" in *Proc. 5th International Conference. Intelligent Tutoring Systems*, Berlin, 2000, pp. 133-142.
- [10] Khuwaja, A. Ramzan, M. W. Evens, J. A. Michael, and A. A. Rovick, "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.
- [11] D. Sleeman and J. S. Brown, "Introduction: Intelligent Tutoring Systems". in *Intelligent Tutoring Systems*, D. Sleeman, J. S. Brown, Ed. Academic Press, 1982, pp. 1-11.
- [12] A. Mitrovic, "An intelligent SQL tutor on the web". *International Journal of Artificial Intelligence in Education*, 13(2-4), 2003, pp. 171-195.
- [13] T. Wang, A. Mitrovic, "Using neural networks to predict student's performance". In *Proc. of International Conference on Computers in Education*, 2002, pp. 969-973.
- [14] D. Zapata-Rivera and J. Greer, "Interacting with inspectable Bayesian student models". *International Journal of Artificial Intelligence in Education*, Vol 14, 2004, pp. 127 - 168.
- [15] C. Conati, A. Gertner, K. VanLehn., and M. Druzdzel, "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.
- [16] S. Weon and J. Kim J, "Learning achievement evaluation strategy using fuzzy membership function" in *Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference*, Reno, NV, 2001, October pp. 10-13.
- [17] J. E. Beck, "Modeling the student with reinforcement learning". *Machine Learning for User Modeling Workshop at the Sixth International Conference on User Modeling*, 1997.
- [18] K. Ferguson, I. Arroyo, S. Mahadevan, B. Woolf, and A. Barto, "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.
- [19] M. W. Evens, S. Brandle, R. Chang, R. Freedman, M. Glass, Y. H. Lee, L. S. Shim, C. W. Woo, Y. Zhang, Y. Zhou, J. A. Michael, and A. A. Rovick "CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue" in *Proc. Twelfth Midwest AI and Cognitive Science Conference*, Oxford, 2001, pp. 16-23.
- [20] S. K. D'Mello, S. D. Craig, B. Gholson, S. Franklin, R. W. Picard, and A. C. Graesser, "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.
- [21] S. Ramachandran, E. Remolina, D. Fu, "FlexiTrainer: A visual authoring framework for case-Based intelligent tutoring systems". *Intelligent Tutoring Systems*, Maceio, Brazil, 2004, pp. 848-850.
- [22] V. Kodaganallur, R. Weitz., and D. Rosenthal, "VersaTutor - An architecture for a constraint-based intelligent tutor generator" in *Proceedings of the Thirteenth Annual World Wide Web Conference*, New York, May 17-22, 2004, pp. 474-475.
- [23] P. Suraweera, A. Mitrovic, and B. Martin, "The use of ontologies in ITS domain knowledge authoring" in *2nd Int. Workshop on Applications of Semantic Web for E-learning SWEL'04, ITS2004*, Maceio, Brazil, 2004, pp. 41-49.
- [24] T. Murray. "An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art". *Authoring Tools for Adv. Tech. Learning Env.* Ainsworth and Blessing, Eds. Kluwer Academic Publishers. Printed in the Netherlands, 2003, pp. 493-546.
- [25] D. Roy, S. Sarkar, S. Ghose, "Learning material annotation for flexible tutoring system". *Journal of Intelligent System*, Vol. 16, No. 4, 2007, pp. 293-305.
- [26] J. E. Beck, "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.
- [27] T. J. Ross, *Fuzzy Logic with Engineering Applications*. McGraw-Hill, 1997, pp. 134-146.
- [28] <http://planningcommission.nic.in/midterm/english-pdf/chapter-10.pdf>
- [29] [www.azimpremjifoundation.org/downloads/SarvaShikshaAbhiyan.pdf](http://www.azimpremjifoundation.org/downloads/SarvaShikshaAbhiyan.pdf)