Intelligent Tutoring Systems and cognitive abilities

Sabeen Durrani
Department of Computer Science
FAST-NU
Lahore, Pakistan
sabeen.durrani@lhr.nu.edu.pk

Dr. Qaiser S. Durrani
Department of Computer Science
FAST-NU
Lahore, Pakistan
qaiser.durrani@nu.edu.pk

Abstract – **Intelligent tutoring system (ITS) is a computer software that provides customized instructions to students according to their learning style, knowledge, and abilities. In reality, most of the existing ITSs do not consider individual cognitive differences and try to teach students primarily on the basis of their domain knowledge and performance. Several cognitive tests [38, 39, 40, 41, 42, 43] are available that measure different cognitive abilities of individuals which make it possible to identify individual differences. In this paper we present a methodology for an ITS that provides individualized environment and teach students according to their cognitive abilities. A prototype has been developed and tested to prove the effectiveness of this methodology. The data analysis shows that the students who were taught according to their cognitive abilities performed significantly better than those who were not provided with teaching material as per their cognitive abilities.**

I. Introduction

Computer based intelligent tutoring systems are gaining popularity because they have shown to be highly effective in increasing learning rate of students [1]. Some of the systems such as Algebra Tutor, Geometry Tutor are being used in large number of schools in USA. Students working with these systems perform a letter grade better than others [63]. Students who studied economics using Smithtown [8] performed equally well with the students studying in traditional classroom environment; however former students required almost half time as compared to later students [1]. The traditional classroom teaching has been very successful; however it has some strengths and limitations. The major benefit includes teacher's interaction with students and creating a collaborative learning environment [24, 25]. While limitations include optimized lecturing to cater general needs and abilities of the students; time constraints in class and lecture's pace. To cater these limitations ITSs were developed and are in use since late 1970s [2, 3]. Such systems tend to model each student individually and make pedagogical decisions regarding teaching strategy based on his/her knowledge, expertise and the interaction history. However, generally these systems do not assess student's domain knowledge before starts teaching and provide same tutoring material to both beginner and expert students. They also do not consider cognitive abilities of students before delivering the lectures. The cognitive abilities such as working memory capacity, verbal, spatial and numerical reasoning are important to explore in addition to domain knowledge [6]. Domain knowledge can be applied in a particular domain only while cognitive abilities can help a student excel and work in any domain because they are domain independent [5].

The purpose of this effort is to develop an intelligent tutoring system that would make use of cognitive abilities in addition to domain knowledge and interaction history of the students. The hypotheses are as follows:

- Students with different cognitive abilities when exposed to lecture contents as per their abilities can achieve equal or better performance as compared to those who are not given contents as per their cognitive abilities.

- Students with high cognitive abilities when presented with lecture contents according to their abilities can learn efficiently as compared to students with low cognitive abilities who are also presented with lecture contents according to their cognitive abilities.

Basic architecture of an ITS is shown in figure 1.

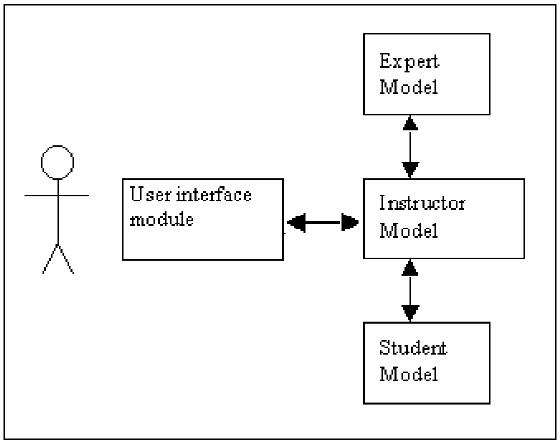

Figure 1: Architecture of Intelligent Tutoring System [35]

Expert model or **domain model** contains complete knowledge of the subject being taught. **Instructor model** or **tutor model** contains instructional methodologies appropriate for different students. **User interface module** is responsible for the interaction between student and the tutor. **Student model** or **user model** contains information of student's current state of knowledge, personal characteristics, preferences, interests, misconceptions and knowledge gaps. Many user modeling techniques are used including explicit user model [14], stereotypes [13, 14], student's interaction history [17], Bayesian methods [26, 27, 28, 29] [30] and machine learning methods [31, 32, 33] [34]. Most of the existing ITSs construct user model based on domain knowledge of the user. Such kind of model is called ***Domain Dependent User Models*** **(DDUM)** because the domain knowledge can be used with in the domain specific application [5]. On the other hand user model built on the basis of cognitive abilities are called ***Domain Independent User Models*** **(DIUM).** Cognitive abilities are involved in all sorts of activities that humans can perform. Models built around cognitive abilities are applicable in virtually all domains [5].

Cognitive abilities determine a person's typical style of perceiving, thinking, remembering and problem solving [5]. The knowledge of differences among people is valuable especially in educational environment because a student's performance may enhance or depreciate depending upon the used teaching methodologies. A teaching method that may work well for a number of students may be counterproductive for students with different deviating abilities [23]. If cognitive abilities of a student are considered in an ITS before presenting lectures, feedback and help then following benefits can be achieved: A system built around the cognitive abilities of student will be domain independent and may also be used in other domains. Identifying user abilities can help in designing better human computer interface to facilitate learning of students. Different tests are available to identify cognitive abilities of a person e.g. cognitive abilities test (CAT or CogAt) [38], Woodcock-Johnson Tests [39], Differential aptitude test (DAT) [40] and many other free online tests [41, 42, 43].

In a "Biocybernetic approach" [23] it was suggested to incorporate student's learning style and abilities in the student model along with student's motive and knowledge. However any data or results to prove this approach are not made available yet. In the system Wayang Outpost [62] SAT Math is to be taught chooses teaching strategy based on gender, spatial abilities and Math fact of students. Results achieved from Wayang Outpost are encouraging [45] and it supports the idea of incorporating cognitive abilities of student in student model.

From current research trends it is found that in most of the ITSs the needed attention is not being given to the cognitive abilities of users. This paper is an attempt to prove the usefulness of incorporating cognitive abilities of user in the user model.

A prototype of tutoring system, called C++ Loop Tutor (CLT), was developed that deals with cognitive abilities, knowledge and interaction history of student. It was then tested on a group of students. In the coming sections brief working of the system, experimental design and results are discussed.

This paper is organized as follows: Section 2 presents the proposed methodology. Section 3 presents the proposed system architecture; section 4 discusses the prototype development. Section 5 presents experiment conducted and section 6 discusses the results and analysis. Conclusion and future work are discussed in section 8.

II. BRIEF WORKING OF CLT

The system comprises of student model, expert model, instructor model, user interface module, evaluation module and feedback module starts by building individual student model by assessing cognitive abilities and domain knowledge of the student using some diagnostic tests. Based on these assessments the system assigns the student to one of the stereotypes from the set of stereotypes. The system then initializes the student model. Based on the student model, system selects the contents from the expert model. The contents have been designed to support students whose knowledge varies between beginner and expert. The selected contents are then presented to the user as per their knowledge and cognitive abilities. As the student goes through the contents the system keeps track of student's interaction such as time spent on the lecture and quiz, number of attempts to solve a quiz and number of times help sought. Based on the interaction history, the system adapts to provide lectures accordingly. At the end of each lecture the system takes an assessment test. Depending upon the assessment the system either moves forward to next lecture or stays on the same lecture with additional examples or reduce the complexity level of the lecture. Based on this assessment test, the system updates the student model.

In the following subsections, system architecture, prototype development, evaluation, results and analysis are discussed.

III. SYSTEM ARCHITECTURE

Following are the components of system architecture: *Instructor model, Student model, Expert model, Evaluation module, Feedback module, User Interface module*. Figure 2 shows the overall proposed architecture of the system. Feedback module and Evaluation module are not considered as separate modules in the literature, they are usually part of instructor model. But we propose to separate them from Instructor model as the major purpose of this model is to instruct the student and present appropriate lecture material. Details of each component are explained below.

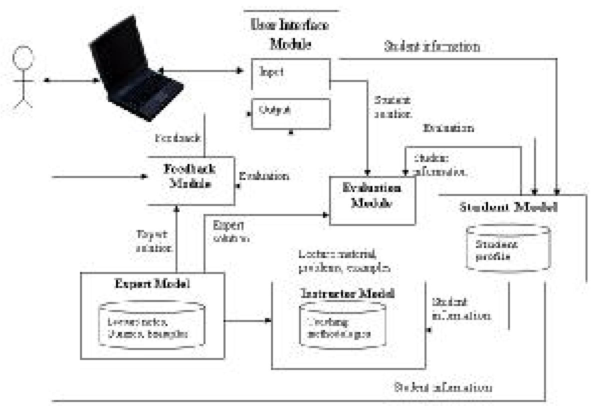

Figure 2: Proposed Architecture of ITS (CLT)

Instructor Model: This model is responsible to instruct the student. It interacts with the student model to get student's information and extract the appropriate tutoring material from expert model according to student's cognitive abilities, knowledge base, and usage of system.

Student Model: This model contains complete information about the student. It is populated through cognitive and knowledge diagnostic test.

Expert Model: This model has a complete repository of lectures, quizzes, problems, solutions, and examples. A variety of lecture contents have been developed to cater the students with different cognitive abilities, knowledge background, deficiencies and preferences.

Evaluation Module: This component evaluates quizzes solved by the students and provides scores to the student model.

Feedback Module: This module gives feedback to student based on interaction with the tutor. It gives immediate feedback to beginner and intermediate students and feedback on demand to expert students. Hence this module is responsible to advise the student on problem solving activities.

User Interface Module: The module is divided into the two sub modules: input and output. An **input module** receives inputs from student in the form of solutions and passes it on to evaluation module for assessment. An **output module** displays the lecture notes, problems, quizzes, feedback etc. through an adaptable graphical user interface.

IV. PROTOTYPE DEVELOPMENT

This section discusses the prototype system that we have developed. The system is called as "C++ Loop Tutor (CLT)". We have chosen C++ iterative constructs (while, do-while, for loop) as the domain to be taught. The details of prototype development are given below:

A. Student modeling parameters

The student model has been constructed based on cognitive abilities, student's domain knowledge and system's usage history.

• *Cognitive abilities:* Spatial, numerical/quantitative and verbal reasoning abilities against each student were assessed via Cognitive abilities test (CAT or CogAt) [44] before delivering lectures to students. This test was chosen because it is widely used in schools of UK and USA to assess cognitive abilities of students. After this test student's cognitive abilities were categorized among low, medium or high. *Spatial reasoning* is the ability to generate, maintain, and manipulate mental visual images. Students with high spatial abilities perform better with abstract and graphic or spatially-oriented content than those with low spatial ability [4]. This ability is chosen because it is related to success of students in programming [64]. N*umerical reasoning* is person's ability to use numbers in a logical and rational way. Students with high numerical reasoning are quick in reasoning with numbers, able to solve problems with relative ease, quickly understand number series, numerical transformations and relationships between numbers [46]. *Verbal reasoning* is person's ability to perceive and understand concepts and ideas expressed verbally. Students with high verbal reasoning have little difficulty in understanding written and verbal communications. However students with low verbal reasoning will need more time to grasp complex verbal and written communications [46].

• *Domain knowledge:* The students were required to have basic knowledge of C++ before start taking lectures from CLT. A diagnostic test was designed to test the domain knowledge of students.

• *System usage history*

Following are the usage based parameters that were recorded while student interacted with CLT: marks obtained in quiz, time spent to learn loops, number of attempts made to pass a quiz, score obtained in first attempt of the quiz, and number of mistakes made in the quiz. This information was used to decide about whether to keep the student involved in the same concept with different set of examples or move him to next concept.

B. Contents development

The contents to be taught have been divided into complexities from lowest to highest depending upon the domain knowledge of students. A typical lecture was divided into *Verbal content (*Textual description), s*patial content (*Figures and flow charts) and n*umerical content (*Examples and problems). Each content type was further divided into categories of low, medium and high (figure 3) to cater different levels of students (based on cognitive abilities).

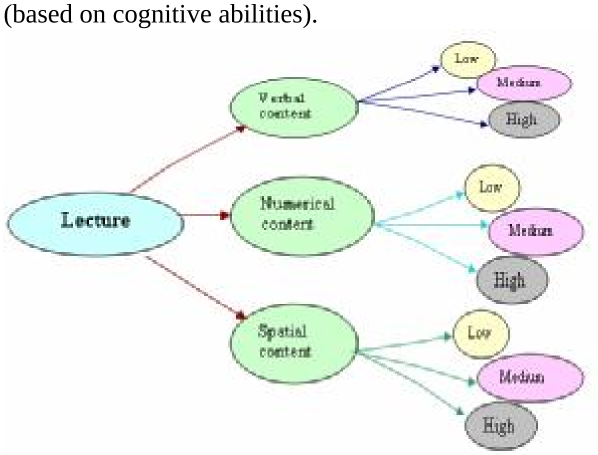

Figure 3: Different content types of lecture

C. Stereotypes

Stereotypes of students were defined on the basis of cognitive abilities of students. Following are some of the stereotypes that were used in the prototype.

Stereotype 1: verbal ability – low, numerical ability – high, spatial ability – medium
Stereotype 2: verbal ability – medium, numerical ability – high, spatial ability – low

This makes total of 27 possible stereotypes.

D. Learning of the system

In this section we present the learning algorithm used to enable the tutor to learn with experience how to teach new students.

The tutor presented selected lectures to the student based on his/her model. After student had gone through the lecture an assessment quiz was given to the student. If the student obtained greater than or equal to 80% marks in the quiz s/he was moved to take next lecture. On the other hand if s/he obtained less than 80% marks in the quiz, then same lecture was given to the student but with the addition of more examples. After student gone through the lecture s/he was exposed to another quiz. This process continued until either student achieved at least 80% score in the quiz or s/he failed to achieve 80% score in three consecutive quizzes. In the later case the student was believed to have problems with the current teaching methodology. This gave a message to the tutor to update the student model and to provide a different form of lecture with different presentation selected from the repository of lectures. With this, the student model was changed temporarily and same topic was taught using different form of presentation and example set. Lectures got modified unless student performed up to the mark i.e. obtained marks greater than or equal to 80% or all the options for lectures were exhausted. If the student performed better with updated model and obtained required marks in the quiz his/her model was assumed to be stable and changed permanently. Future lectures were presented according to this new model. When a student's model was changed permanently, it was added to list called modified list to keep count of number of students belonging to same stereotype whose model was changed permanently. When this count reached five and new student belonging to same stereotype came to the system, then lectures were presented to him/her in the same format as were now presented to those five students. In this way the system learnt how to teach a student based upon previous experiences.

In this algorithm it was proposed to change the student model if s/he does not perform well with the current model. There can be following reasons behind the need to change student's model and why the student does not perform well. One reason may be that cognitive abilities test could not properly evaluate or identify the cognitive abilities of the student. The other reason could be that the problems presented in the evaluation were ill designed or insufficient to evaluate the individual.

V. EXPERIMENT

In this section we present details of the experiment conducted in order to test our hypotheses and to provide the solution of the identified problem.

66 first semester computer science students were chosen to test the hypothesis. These students were divided into three groups named TestGroup1_eLC, TestGroup2_eLNC, and TestGroup3_CR. **TestGroup1_eLC** consisted of students who did not have any formal programming exposure prior to this experiment. They were exposed to CLT and contents were presented on the basis of their cognitive abilities, domain knowledge and system usage. **TestGroup2_eLNC** consisted of a mix of students. Some students had formal education of programming while some did not. These students were exposed to CLT but they did not go through cognitive tests hence, the system taught them the contents on the basis of their domain knowledge and system usage only. Students of **TestGroup3_CR** learnt C++ loops in traditional classroom environment and some of them had formal educational background of programming while some did not.

Students in TestGroup1_eLC gave cognitive abilities test (CAT). Result of test is shown in figure 4.

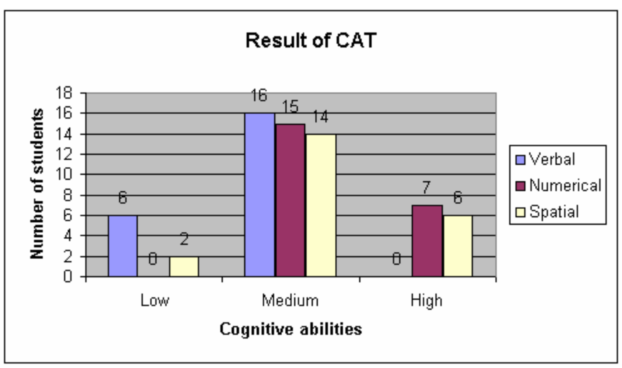

Figure 4: Result of CAT

There were three supervised sessions of 90 minutes each during the execution period. TestGroup1_eLC and TestGroup2_eLNC used the CLT to learn C++ loops and TestGroup3_CR went through these concepts in traditional classroom setting. Apart from supervised sessions for TestGroup1_eLC and TestGroup2_eLNC, these students could access CLT at any time in the university labs. TestGroup1_eLC and TestGroup2_eLNC had an advantage of prepared lecture notes available via CLT, while TestGroup3_CR had to learn through classroom or books. On the other hand, TestGroup3_CR had an advantage of the presence of teacher during lectures.

At the end of supervised sessions, five tests were conducted for all three groups in class room to access how much they learned the loop concepts and how one group performs as compared to the others. Results of these five tests are presented in figure 11 (appendix). Detail of the results and data analysis is discussed in next section.

VI. RESULTS AND ANALYSIS

In this section the data gathered in experiment is analyzed and the results are discussed.

Average score in FA/FSc/A-levels (12^{th} grade), first midterm exam (taken prior to this experiment) and final exam by the three test groups is shown in figure 5. The average result of final exam shows that TestGroup1_eLC performed better than the other two groups, while TestGroup3_CR average score is least in the two exams. Hence, TestGroup1_eLC was able to learn C++ loops better than other groups.

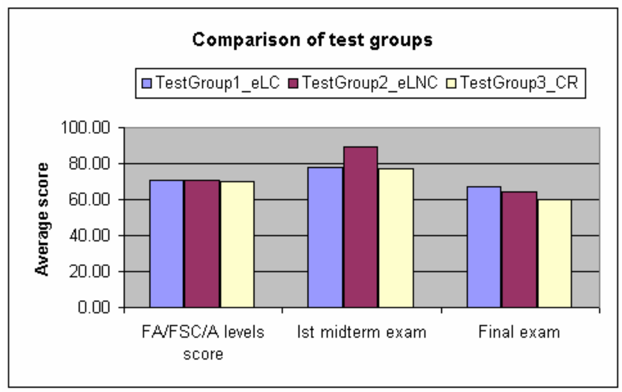

Figure 5: Comparison of average scores of Evaluation Test

Comparison of averages of the three groups in five tests is presented in figure 6. Score of TestGroup1_eLC is the highest among the three test groups.

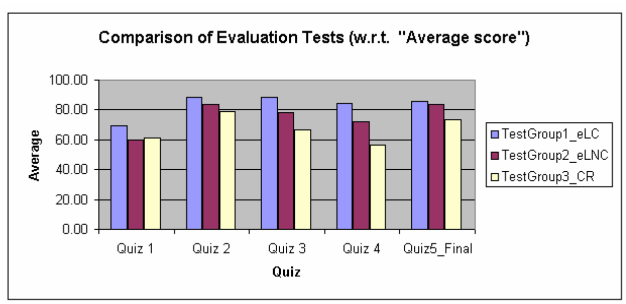

Figure 6: Comparison of average scores of Evaluation Test

In all the quizzes on average TestGroup1_eLC got 83.39 marks, TestGroup2_eLNC got 75.53 and TestGroup3_CR got 67.18 marks. There is a significant gap between the scores of these three groups. Statistical analysis of the quiz scores of the three test groups is done below in order to identify the significance of difference between the quiz results. Since there are more than two samples, to compare the means of corresponding populations analysis of variance (ANOVA with single factor) is used. From the data missing observations are ignored and hence the sample size from the three populations is unequal. Analysis of variance table is shown below:

ANOVA

Source of Variation	SS	df	MS	F	P-value
Between Groups	13923.96	2	6961.978	10.12478	5.48067E-05
Within Groups	215912	314	687.6179		
Total	229836	316			

The analysis of this data shows that the p value (5.48067E-05) is less than the significance level (0.05) and hence the difference between the quiz scores of the three test groups is significant. As ANOVA gives positive difference between quizzes scores of the three test groups, now individual comparison between the three test groups is made below by using t - test. Computed value of t at significance level = 0.05 with degree of freedom = 317 for the test groups is given below:

't' value for TestGroup1_eLC and TestGroup2_eLNC	2.1
't' value for TestGroup1_eLC and TestGropu3_CR	4.4
't' value for TestGroup2_eLNC and TestGropu3_CR	2.3

On comparing the computed 't' values with value of standard normal variable $Z_{0.025} = 1.96$ it is found that all the t values are larger. From this it is concluded that when pair wise comparison is made all the means of different populations are significantly different.

Average semester score of TestGroup1_eLC was 68.25, TestGroup2_eLNC was 72.27 and TestGroup3_CR was 70.19. The semester work did not include loops and advanced topics. TestGroup1_eLC scored least on average in semester work and TestGroup2_eLNC performed best in semester work. Although this difference is not significant enough, however, if we compare semester work with the result of evaluation test, we find that TestGroup1_eLC has significantly improved its performance.

The average time spent by TestGroup1_eLC with the CLT in order to learn C++ loops is six hours and TestGroup2_eLNC spent approximately seven hours on average. Average time taken by students with different abilities in TestGroup1_eLC was also compared (figure 7). This analysis shows that students with high abilities were able to complete the tasks assigned efficiently (approximately one hour less) as compared to students with low abilities.

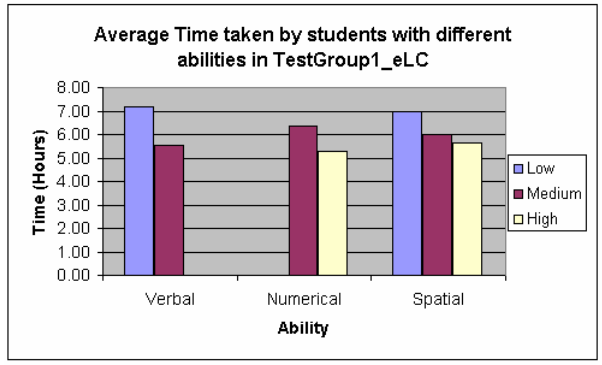

Figure 7: Average time spent by students of TestGroup1_eLC with different cognitive abilities

TestGroup3_CR spent 4.5 hours in class while attending the lecture, TestGroup1_eLC took on average six hours and TestGroup2_eLNC took 6.5 hours on average. With CLT the students learnt according to their pace as they had to study on their own that is why they took more time as compared to students who learnt loops in class room.

Comparison of evaluation tests score of students of TestGroup1_eLC with respect to the scores obtained in the three cognitive test batteries is shown in figures 8, 9 and 10. The data in figure 8 shows that average scores obtained, in five evaluation tests, by students with low verbal ability and students with medium verbal ability are very close to each other. Surprisingly, students with low verbal ability performed better than students with high verbal ability in last test i.e. Quiz 5. This shows that lecture contents used language according to the verbal abilities of students and effectively handled different requirements of students regarding their verbal ability.

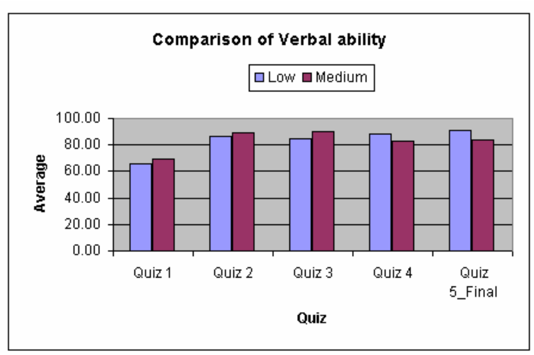

Figure 8: Comparison of students of TestGroup1_eLC w.r.t verbal battery

In quantitative battery (figure 9) students were categorized either medium or high and not low. In general high level students performed better than medium level in three tests. In other tests their average scores are close to each other. This shows that numerical ability was taken care of effectively and the students were able to solve most of the problems in evaluation test. Hence, their weakness and limitations were addressed properly by the lectures.

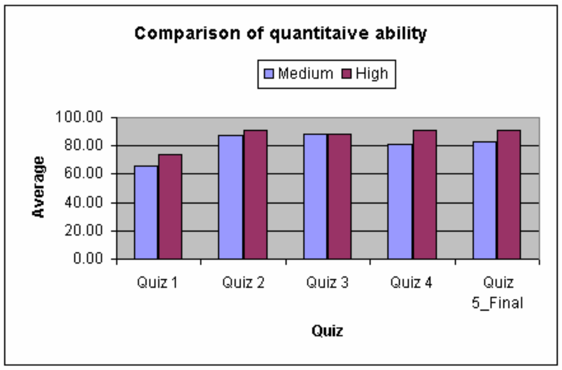

Figure 9: Comparison of students of TestGroup1_eLC w.r.t. quantitative battery

In spatial battery (figure 10) high level students have performed significantly better than other students. Low level students have not performed well, but the number of students in this category is only 2. Hence, this data is not significant enough to make any conclusion about this category of students. However, data shows that these students have considerably improved their performance and their score is constantly increasing through out the tests. Medium level students have also improved their performance significantly as can be seen from the data.

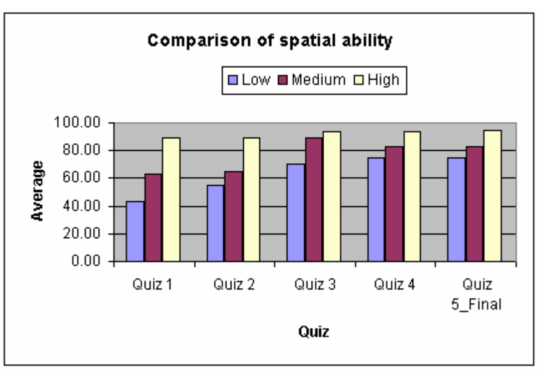

Figure 10: Comparison of students of TestGroup1_eLC w.r.t spatial battery

We can conclude from above discussion and data presented that spatial ability plays a very important role in learning loops concept. Major gap between test scores is observed among students with respect to spatial ability. However, motivating fact is that students with low and medium spatial abilities have improved their performance continuously. For other cognitive abilities, students have performed almost equally and their scores are very close. Hence, our hypothesis is proved that students can perform significantly better if they are taught individually according to their cognitive abilities along with domain knowledge and interaction history. Results are more satisfactory as the interface has been designed properly according to the design guidelines and it gets personalized for each student's requirements.

VII. CONCLUSION

In this paper we have presented an idea of developing an intelligent tutoring system using cognitive aspects of the users. Information about the student like domain knowledge, skills, characteristics and cognitive abilities is very important for an intelligent tutoring system (ITS). Without this information, it is not possible for an ITS to present a completely individualized teaching environment to each student. Existing ITSs, which are being utilized in many different domains, construct student model but they largely lack to consider cognitive abilities. Due to differences in cognitive abilities students perform differently even though they are provided with the same instructions and lectures. To test our idea of incorporating cognitive abilities in student model we developed a prototype of an ITS and it was tested on freshmen students who were split randomly into three groups. The results are briefly discussed.

- As per first hypothesis the data showed that students who were given contents as per their cognitive abilities performed approximately 12% better than others.
- Students with high cognitive abilities when presented with lecture contents according to their abilities spent approximately one to two hours less.
- When one on one tutoring based on cognitive abilities was given to students, they secured 83.39% marks. Students who were given one on one teaching but without considering their cognitive abilities secured 75.53% marks. However, both these groups performed better than students who were taught in classroom and secured 67.18% marks.

Although the results are quite encouraging, there are some limitations that need to be handled and also some more avenues that need to be explored. Due to small student base all different types of lectures and learning algorithm could not be tested so, more thorough testing needs to be done to prove significance of lectures and the learning algorithm. Lecture contents should be developed to cater other cognitive abilities of students including working memory, attention, learning, visual thinking etc.

REFERENCES

[1] Joseph Beck, Mia Stern, and Erik Haugsjaa *Applications of AI in Education* 2004, The Association for Computing Machinery, Inc.

[2] Corbett, A. T., Koedinger, K. R., Anderson, J. R. *Handbook of Human-Computer Interaction*, Second, Completely Revised Edition in M.Helander, T. K. Landauer, P. Prabhu (Eds), Elsevier Science B. V., 1997, Chapter 37

[3] C.J.Butz, S.Hua, R.B.Maguire *A web-based Bayesian Intelligent Tutoring System for computer programming*, Web Intelligence and Agent Systems Volume 4 , Issue 1 2006, Publisher: IOS Press Amsterdam, The Netherlands, The Netherlands

[4] Nancy MILIK, Antonija MITROVIC and Michael GRIMLEY *Fitting Spatial Ability into Intelligent Tutoring Systems Development*, Artificial Intelligence in Education, R.Luckin et al. (EDs.) IOS Press, 2007

[5] Qaiser Sheharyar Durrani Cognitive Modeling. *A Domain Independent User Modeling.* IEEE International conference on System man and cybernetics, Hyatt Orlando, Orlando Florida, USA, 1997.

[6] Hassan Zia, Qaiser S. Durrani, Rana Adnan Farrakh, Amir Riaz, Farhan Ahmed *CITS - C++ Intelligent Tutoring System: A Domain Independent User Centered Curriculum* Approach IJCAI'99 workshop on Learning about Users 1999

[7] John R. Anderson, C-Franklin Boyle, Albert T. Corbett and Matthew W. Lewis *Cognitive Modeling and Intelligent Tutoring* Artificial Intelligence 42(1990) 7-49

[8] Kalyani Raghavan, Arnold Katz Smithtown: *An Intelligent Tutoring System* Journal article T H E Journal (Technological Horizons In Education), Vol. 17, 1989

[9] http://www.autotutor.org

[10] Alfred Kobsa, *User Modeling: Recent work, Prospects and Hazards Adaptive User Interfaces*: Principles and Practice (1993)

[11] Edward Ross Intelligent *User Interfaces : Survey and Research* Directions January 2000

[12] Josef Fink, Alfred Kobsa, Jrg Schreck *Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features: User Modeling, Privacy and Security Issues* Technology for Cooperative Competition. Berlin Heidelberg: Springer, 459-467

[13] Alfred Kobsa, *User Modeling in Dialog Systems: Potentials and Hazards* AI & Society 4(3), 214-240, 1990

[14] Elaine Rich, *User Modeling via Stereotypes* COGNITIVE SCIENCE 3, 329-354 (1979)

[15] http://www.cmu.edu/stattutor

[16] http://www.andes.pitt.edu

[17] Alfred Kobsa, *Supporting User Interfaces for All Through User Modeling* Proceedings HCI International '95, Yokohama, Japan, 155-157, 1995

[18] Mark Urban-Lurain Intelligent Tutoring Systems: *An historic review in the context of the development of Artificial Intelligence and educational psychology*, http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm, August 3, 2004.

[19] John R. Anderson ACT - *A simple theory of complex cognition* April 1996 American Psychologist Association Vol. 51, No. 4, 355-365

[20] Anderson, J. R. & Gluck, K. (2001). *What role do cognitive architectures play in intelligent tutoring systems?* In D. Klahr & S. M. Carver (Eds.) Cognition & Instruction: Twenty five years of progress, 227-262. Erlbaum.

[21] Albert T. Corbett, John R. Anderson, *Locus of Feedback Control in Computer-Based Tutoring: Impact on Learning Rate, Achievement and Attitudes* In Jacko, J., Sears, A., Beaudouin-Lafon, M. and Jacob, R. (Eds.) Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems, 245-252. New York: ACM Press.

[22] Santosh A. Mathan, Kenneth R. Koedinger *Fostering the Intelligent Novice: Learning From Errors With Metacognitive Tutoring* EDUCATIONAL PSYCHOLOGIST 2005, 40(4), 257-265 Lawrence Erlbaum Associates, Inc.

[23] Stamos T. Karamouzis, Andreas Vrettos *A biocybernetic approach for intelligent tutoring systems*, Proceedings of the 25th LASTED International Multi- Conference Artificial Intelligence and Applications February 12-14,2007, Innsbruck, Austria

[24] http://www.comet.ucar.edu/presentations/ams96/index.htm

[25] http://lone-eagles.com/curr2.htm

[26] Anderson, J. R. (1993). *Rules of the Mind.* Mahwah, NJ: Lawrence Erlbaum.

[27] Corbett, A., Anderson, J *Knowledge tracing: modeling the acquisition of procedural knowledge*, User Model. User Adap. Inter. 4, 253-273 (1995)

[28] Reye, J. *Student modeling based on belief networks*. Int. J. Artif. Intell. Educ. 14, 1-33(2004)

[29] C. Conati, A. Gertner and K. Vanlehn, *Using Bayesian networks to manage uncertainty in student modeling*, User Modeling and User-Adapted Interaction, 12(4) (2002) 371-417.

[30] Michal V. Yudelson, Olga P. Medvedeva, Rebecca S. Crowley *A multifactor approach to student model evaluation*, User Modeling and User-Adapted Interaction 18: 349 - 382 (2008)

[31] Geo_rey I.Webb, Michael J. Pazzani, and Daniel Billsus. *Machine Learning for User Modeling*, User Modeling and User-Adapted Interaction, vol.11, no. 1-2 p.p. 1929, 2001.

[32] Ferguson, K., Arroyo,I., Mahadevan, S., Woolf, B., Barto, A. *Improving intelligent tutoring systems: Using EM to learn*

student skill levels, Intelligent Tutoring Systems, pp. 453-462. Springer-Verlag, Jhongli, Taiwan (2006)

[33] Jonsson, A., Johns, J., Mehranian, H., Arroyo, I., Woolf, B., Barto, A., Fisher, D., Mahandevan, S. *Evaluating the geasibility of learning student models from data*, AAAI05 Workshop on Educational Data Mining, pp. 1-6. Pittsburgh, PA (2005)

[34] Chang, K., Beck, J., Mostow, J., Corbett, A. *Does help help? A Bayes net approach to modeling tutor interventions,* Proceedings of the 21st Annual Meeting of the American Association for Artificial Intelligence, pp. 41-46. Boston, MA (2006)

[35] Jim Ong and Sowmya Ramachandran, *Intelligent Tutoring Systems: The What and the How*, Learning Circuits (February 2000)

[36] Kinshuk Massey University, New Zealand *Does intelligent tutoring have future!* Proceedings of the International Conference on Computers in Education (ICCE02) 0-7695-1509-6/02 2002 IEEE

[37] Shlomo Berkovsky , Tsvi Kuflik, Francesco Ricci *Cross-representation mediation of user models*, User Model User-Adap Inter (2009)

[38] http://www.gl-assessment.co.uk/education/resources/cat3/cat3.asp

[39] http://www.riverpub.com/products/wjIIICognitive/

[40] http://www.shrm.org/testing/products/psychcorp/DAT.asp

[41] http://www.iqtest.com/

[42] http://www.intelligencetest.com/

[43] http://www.free-iqtest.net/

[44] http://www.gl-assessment.co.uk/education/resources/cat3/cat3.asp

[45] Ivon Arroyo, Carole Beal, Tom Murray, Rena Walles, Beverly Woolf *Wayang Outpost: Intelligent Tutoring for High Stakes Achievement Tests* University of Massachusetts Amherst

[46] http://www.beilby.com.au/Employers/Employer+Resource+Centre/Reasoning+Abilities+/default.asp

[47] D.S. Malik C++ Programming: Program Design inclusing Data Structures

[48] Deitel & Deitel C++ How to program

[49] Nell Dale, Chip Weems, Mark Headington Programming and problem solving with C++

[50] Alfred Kobsa, Dietmar Mller, Andreas Nill KN-AHS: *An Adaptive Hypertext Client of the User Modeling System BGP-MS* Proceedings of the fourth international conference on user modeling, Hyannis, MA, 1994

[51] Albert T. Corbett, John R. Anderson and Alison T. O´brien *The predictive validity of student modeling in the ACT Programming Tutor* (2008). Psychology. Paper 80.

[52] Mingye Feng, Neil Heffernan, Kenneth Koedinger *Addressing the assessment challenge with an online system that tutors as it assesses* User Model User- Adaptive Interaction (2009), 243-266

[53] Liliana Ardissono, Cristina Gena, Pietro Torasso, Fabio Bellifemine, Angelo Difino, Barbara Negro *User Modeling and Recommendation Techniques for Personalized Electronic Program Guides*

[54] Elliott, D. and Hopfgartner, F. and Leelanupab, T. and Moshfeghi, Y. and Jose, J.M. (2009) *An architecture for life-long user modelling.* In: Lifelong User Modelling Workshop held in conjunction with User Modeling, Adaptation and Personalisation (UMAP) 2009 , 22-26 June 2009 , Trento, Italy.

[55] Kyriacou E, D. (2009) *Enriching Lifelong User Modelling with the Social e-Networking and e-Commerce "Pieces of the Puzzle".* In: First and Seventeenth International Conference on User Modeling, Adaptation, and Personalization (UMAP2009) - Doctoral Consortium, 22-26/06/2009, Trento, Italy.

[56] Yuan Wang and Julita Vassileva *A user-centric authentication and privacy control mechanism for user model interoperability in social networking sites* Workshop on Adaptation and Personalization for Web 2.0, UMAP'09, June 22-26, 2009

[57] Chevalier, M., Julien C., Soule-Dupuy C. 2008. *Collaborative and social information retrieval and access: techniques for improved user modeling* IGI Global, 2008.

[58] Mohamed Neji *The innovative approach to user modeling for information retrieval system*, Hypertext SRC'09, June 29 - July 1st, 2009, Torino, Italy

[59] Nunes, M.A.S. 2008. *Recommender systems based on personality traits,* PhD thesis. University Montpellier 2-LIRMM, December, 2008

[60] http://wayang.cs.umass.edu/animalwatch/

[61] http://www.andes.pitt.edu

[62] http://wayang.cs.umass.edu/wayang/

[63] H. Chad Lane Intelligent Tutoring Systems: *Prospects for Guided Practice and Efficient Learning* July 27, 2006

[64] Sue Jones, Gary Burnett *Spatial ability and learning to program* Human Technology An Interdisciplinary Journal on Humans in ICT Environments 2008, 47-61

[65] Sabeen Durrani, *Intelligent Tutoring System and Cognitive Modeling,* Ms Thesis Report 2009

Appendix

Evaluation Quiz comparison

Test Group 1
(Taught via ITS according to cognitive abilities)

Std Id	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Quiz 5
1	100	100	100	100	90
2	40	40	80	50	80
3	87	90	100	70	100
4	97	100	100	100	100
5	70	90	80	100	100
6	80	100	90	100	80
7	87	90	80	100	100
8	80	100	100	100	90
9	17	50	90	30	10
10	93	100	100	100	100
11	37	80	70	80	95
12	73	100	100	90	-
13	47	70	60	100	70
14	80	90	90	100	95
15	27	100	70	30	-
16	97	100	100	100	90
17	73	90	90	60	90
18	87	80	90	100	85
19	67	100	80	100	-
20	40	90	90	100	-
21	70	90	90	50	-
22	77	100	100	100	85
Average	69.24	88.64	88.64	84.55	85.88
Min	16.67	40.00	60.00	30.00	10.00
Max	100.00	100.00	100.00	100.00	100.00
s_dev	24.15	16.42	11.67	24.44	21.38

Test Group 2
(Taught via ITS not considering cognitive abilities)

Std Id	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Quiz 5
23	27	90	80	0	40
24	93	90	100	80	100
25	37	80	90	0	60
26	23	90	0	80	95
27	77	90	70	100	90
28	-	100	90	50	100
29	60	-	-	0	-
30	73	80	70	90	70
31	63	100	80	90	85
32	77	90	90	80	80
33	83	100	100	100	100
34	23	60	70	80	80
35	40	90	90	100	85
36	80	0	-	-	90
37	50	90	90	70	-
38	47	100	80	100	100
39	80	100	90	100	100
40	97	100	100	100	100
41	33	50	80	0	55
42	47	60	0	100	75
43	-	100	90	90	90
44	93	100	100	100	80
Average	60.17	83.81	78.00	71.90	83.75
Min	23.33	0.00	0.00	0.00	40.00
Max	96.67	100.00	100.00	100.00	100.00
s_dev	24.67	24.18	28.40	37.90	16.93

Test Group 3
(Taught via classroom lectures)

Std Id	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Quiz 5
45	40	70	50	0	60
46	27	70	80	40	60
47	30	80	20	0	35
48	37	50	60	40	60
49	43	40	20	20	70
50	47	100	100	100	55
51	53	100	80	20	40
52	53	100	80	90	70
53	53	100	70	100	40
54	53	100	90	100	50
55	57	80	80	100	75
56	57	70	0	0	80
57	60	100	90	100	100
58	63	40	0	0	85
59	67	70	30	0	100
60	73	90	80	80	80
61	80	70	80	70	85
62	83	80	100	100	100
63	87	90	70	50	100
64	87	60	80	60	90
65	90	80	100	80	85
66	100	90	100	90	100
Average	60.91	78.64	66.36	56.36	73.64
Min	26.67	40.00	0.00	0.00	35.00
Max	100.00	100.00	100.00	100.00	100.00
s_dev	20.35	19.10	32.15	40.30	21.22

Figure 11: Evaluation test results