The Use of Intelligent Tutoring System for Developing Web-based Learning Communities

Ahmed R. Salman [1,2]

[1] **Computer Science, Taif University**
Taif , Zip Code 21974, Saudi Arabia

[2] **Computer T. P., Mansoura University**
Mansoura , Zip Code 35516, Egypt

Abstract

The purpose of this paper is to describe a web- based intelligent tutoring system (ITS) framework. A formal model of ITS compose of an user environment and pedagogical environment is presented, which represents domain knowledge based on ontologies to improve the sharing and reusing of teaching materials. The system constructs the user environment based on users' knowledge levels, psychology characteristics, learning style, etc. Finally this paper attempts to emphasis the importance of intelligent tutoring system by confirming the effectiveness of tutorial programs based on artificial intelligence for developing web-based learning communities.

Keywords: *Intelligent tutoring system; Artificial Intelligence in Education; Web-based Learning Communities; E-learning.*

1. Introduction

Since we've become teaching, providing knowledge to all that seek for it, with individualized support whenever it's needed (anytime, anywhere), when solicited, ideally even when that support isn't asked but the need for it is there, has been the ultimate goal search for everyone. In fact, there are studies [20] that show that one-on-one human tutoring is more effective than other modes of instruction. ITS are distributed systems capable to support on-line tutoring functionalities for the learning and evaluation in multi-disciplinary domains.

They must be capable of accurately diagnose students' knowledge structures, skills, and styles; diagnose using principles, rather than pre-programmed responses; decide what to do next; adapt instruction accordingly; provide feedback. The advances in the development of the computer technologies has facilitated the use and design of ITS. The proliferation of ITS, has spawned many debates about their use and effectiveness: The Degree of Learner Control: How much learner control should be allowed by the systems? Individual vs. Collaborative Learning: Should learners interact with ITS individually or collaboratively? Situated Learning: Is learning situated, unique, and ongoing, or is it more symbolic, following from an information processing model? Virtual Reality and Learning: Does virtual reality uniquely contribute to learning beyond computer aided instruction or multimedia? Despite their advantages, ITS fail to prove their usefulness in wider academic environment, mainly because research is primarily driven by computer scientists, and doesn't address all the different issues from other fields [3]. The integration of this type of systems in an organization provokes obvious impacts at several levels, conditioning the success of its integration. Some aspects should also be considered like impartiality in evaluation, teachers' autonomy, time and resources administration and, accessibilities. They are seen as they are integrated and highly interrelated with the process of knowledge management (e.g., student models and profiles, knowledge base of each specific domain), dialoguing and argumentation (to facilitate tutor/ student knowledge exchange) and experimentation by simulation (simulated evaluation to provide student's knowledge feedback and pedagogical experimentation). All these processes are continuously supported by technologies that accurately address issues related namely to ontology management, intelligent agents, data warehousing, data mining, case-based and rule based reasoning, adaptive interfaces and user modelling.

Intelligent Tutoring Systems (ITSs) are computer-based instructional systems with models of instructional content that specify what to teach, and teaching strategies that specify how to teach [6][24]. They make inferences about a student's mastery of topics or tasks in order to dynamically adapt the content or style of instruction. Content models (or knowledge bases, or expert systems, or simulations) give ITSs depth so that students can "learn by doing" in realistic and meaningful contexts. Models allow for content to be generated "on the fly." ITSs allow "mixed-initiative" tutorial interactions, where students can ask questions and have more control over their learning. Instructional models allow the computer tutor to more closely approach the benefits of individualized instruction by a competent pedagogue. In the last decade ITSs have moved out of the lab and into classrooms and workplaces where some have been shown to be highly effective [26][9].While intelligent tutors are becoming more common and proving to be increasingly effective they are difficult and expensive to build. Authoring systems are commercially available for traditional computer aided instruction (CAI) and multimedia-based training, but these authoring systems lack the sophistication required to build intelligent tutors. Commercial multimedia authoring systems excel in giving the instructional designer tools to produce visually appealing and interactive screens, but behind the screens is a shallow representation of content and pedagogy.

2.Historical And Related Work Of ITS

2.1 Early mechanical systems.

The possibility of intelligent machines have been discussed for centuries. Blais Pascal created the first calculating machine capable of mathematical functions in the 17th century simply called Pascal's Calculator. At this time the mathematician and philosopher Leibniz envisioned machines capable of reasoning and applying rules of logic to settle disputes [4]. These early works contributed to the development of the computer and future applications.

The concept of intelligent machines for instructional use date back as early as 1924, when Sidney Pressey of Ohio State University created a mechanical teaching machine to instruct students without a human teacher.[5] His machine resembled closely a typewriter with several keys and a window that provided the learner with questions. The Pressey Machine allowed user input and provided immediate feedback by recording their score on a counter.[25]

Pressey himself was influenced by Edward L. Thorndike, a learning theorist and educational psychologist at the Columbia University Teacher College of the late 19th and early 20th centuries. Thorndike posited laws for maximizing learning. Thorndike's laws included the law of effect, the law of exercise, and the law of recency. Following later standards, Pressey's teaching and testing machine would not considered intelligent as it was mechanically run and was based on one question and answer at a time,[25] but it set an early precedent for future projects. By the 1950s and 1960s, new perspectives on learning were emerging. Burrhus Frederic "B.F." Skinner at Harvard University did not agree with Thorndike's learning theory of connectionism or Pressey's teaching machine. Rather, Skinner was a behaviourist who believed that learners should construct their answers and not rely on recognition.[5] He too, constructed a teaching machine structured using an incremental mechanical system that would reward students for correct responses to questions.[5]

2.2 Early electronic systems

In the period following the second world war, mechanical binary systems gave way to binary based electronic machines. These machines were considered intelligent when compared to their mechanical counterparts as they had the capacity to make logical decisions. However, the study of defining and recognizing a machine intelligence was still in its infancy.

Alan Turing, a mathematician, logician and computer scientist, linked computing systems to thinking. One of his most notable papers outlined a hypothetical test to assess the intelligence of a machine which came to be known as the Turing test. Essentially, the test would have a person communicate with two other agents, a human and a computer asking questions to both recipients. The computer passes the test if it can respond in such a way that the human posing the questions cannot differentiate between the other human and the computer. The Turing test has been used in its essence for more than two decades as a model for current ITS development. The main ideal for ITS systems is to effectively communicate.[25] As early as the 1950s programs were emerging displaying intelligent features. Turing's work as well as later projects by researchers such as Allen Newell, Clifford Shaw, and Herb Simon showed programs capable of creating logical proofs and theorems. Their program, The Logic Theorist exhibited complex symbol manipulation and even generation of new information without direct human control and is considered by some to be the first AI program. Such breakthroughs would inspire the new field of Artificial Intelligence officially named in 1956 by John McCarthy in 1956 at the Dartmouth Conference.[4] This conference was the first of its kind that was devoted to scientists and research in the field of AI.

The latter part of the 1960s and 1970s saw many new CAI (Computer-Assisted instruction) projects that built on advances in computer science. The creation of the BASIC programming language in 1958 enabled many schools and universities to begin developing Computer Assisted Instruction (CAI) programs. Major computer vendors and federal agencies in the US such as IBM, HP, and the National Science Foundation funded the development of these projects.[11] Early implementations in education focused on programmed instruction (PI), a structure based on a computerized input - output system. Although many supported this form of instruction, there was limited evidence supporting its effectiveness.[25] The programming language LOGO was created in 1967 by Wally Feurzeig and Seymour Papert as a language streamlined for education. PLATO, an educational terminal featuring displays, animations, and touch controls that could store and deliver large amounts of course material, was developed by Donald Bitzer in the University of Illinois in the early 1970s. Along with these, many other CAI projects were initiated in many countries including the US, the UK, and Canada.[11]

At the same time that CAI was gaining interest, Jaime Carbonell suggested that computers could act as a teacher rather than just a tool [27]. A new perspective would emerge that focused on the use of computers to intelligently coach students called Intelligent Computer Assisted Instruction or Intelligent Tutoring Systems (ITS). Where CAI used a behaviorists perspective on learning based on Skinner's theories. ITS drew from work in cognitive psychology, computer science, and especially artificial intelligence. There was a shift in AI research at this time as systems moved from the logic focus of the previous decade to knowledge based systems—systems could make intelligent decisions based on prior knowledge.[4] Such a program was created by Seymour Papert and Ira Goldstein who created Dendral, a system that predicted possible chemical structures from existing data. Further work began to showcase analogical reasoning and language processing. These changes with a focus on knowledge had big implications for how computers could be used in instruction. The technical requirements of ITS, however, proved to be higher and more complex than CAI systems and ITS systems would find limited success at this time.[11]

Towards the latter part of the 70's interest in CAI technologies began to wane.[11][12] Computers were still expensive and not as available as expected. Developers and instructors were reacting negatively to the high cost of developing CAI programs, the inadequate provision for instructor training, and the lack of resources.[12]

2.3 Microcomputers and intelligent systems

The microcomputer revolution in the late 1970s and early 80s helped to revive CAI development and jumpstart development of ITS systems. Personal computers such as the Apple 2, Commodore PET, and TRS-80 reduced the resources required to own computers and by 1981, 50% of US schools were using computers.[11] Several CAI projects utilized the Apple 2 as a system to deliver CAI programs in high schools and universities including the British Columbia Project and California State University Project in 1981.[11]

The early 80s would also see ICAI and ITS goals diverge from its roots in CAI. As CAI became increasingly focused on deeper interactions with content created for a specific area of interest, ITS sought to create systems that focused on knowledge of the task and the ability to generalize that knowledge in non-specific ways.[10] The key goals set out for ITS were to be able to teach a task as well as perform it, adapting dynamically to its situation. In the transition from CAI to ICAI systems, the computer would have to distinguish not only between the correct and incorrect response but the type of incorrect response to adjust the type of instruction. Research in Artificial Intelligence and Cognitive Psychology fueled the new principles of ITS. Psychologists considered how a computer could solve problems and perform 'intelligent' activities. An ITS program would have to be able to represent, store and retrieve knowledge and even search its own database to derive its own new knowledge to respond to learner's questions. Basically, early specifications for ITS or (ICAI) require it to "diagnose errors and tailor remediation based on the diagnosis".[25] The idea of diagnosis and remediation is still in use today when programming ITS.

A key breakthrough in ITS research was the creation of LISPITS, a program that implemented ITS principles in a practical way and showed promising effects increasing student performance. LISPITS was developed and researched in 1983 as an ITS system for teaching students the LISP programming language.[2] LISPITS could identify mistakes and provide constructive feedback to students while they were performing the exercise. The system was found to decrease the time required to complete the exercises while improving student test scores.[2] Other ITS systems beginning to develop around this time include TUTOR created by Logica in 1984 as a general instructional tool. [16] and PARNASSUS created in Carnegie Mellon University in 1989 for language instruction.[3]

2.4 Modern ITS and Web-based Communities

After the implementation of initial ITS, more researchers created a number of ITS for different students. In the late 20th century, Intelligent Tutoring Tools (ITTs) was developed by the Byzantium project, which involved six universities. The ITTs were general purpose tutoring system builders and many institutions had positive feedbacks while using them.[13] This builder, ITT, would produce an Intelligent Tutoring Applet (ITA) for different subject areas. Different teachers created the ITAs and built up a large inventory of knowledge that was accessible by others through the Internet. Once an ITS was created, teachers could copy it and modify it for future use. This system was efficient and flexible. However, Kinshuk and Patel believed that the ITS was not designed from an educational point of view and was not developed based on the actual needs of students and teachers.[14]

There were three ITS projects that functioned based on conversational dialogue: AutoTutor, Atlas [21] and Why2. The idea behind these projects was that since students learn best by constructing knowledge themselves, the programs would begin with leading questions for the students and would give out answers as a last resort. AutoTutor's students focused on answering questions about computer technology, Atlas's students focused on solving quantitative problems, and Why2's students focused on explaining physical systems qualitatively.[8] Other similar tutoring systems such as Andes [7] tend to provide hints and immediate feedbacks for students when students have trouble answering the questions. They could guess their answers and have correct answers without deep understanding of the concepts. Research was done with a small group of students using Atlas and Andes respectively. The results showed that students using Atlas made significant improvements compared with students who used Andes.[22] However, since the above systems require analysis of students' dialogues, improvement is yet to be made so that more complicated dialogues can be managed.

The idea of Web-based intelligent tutoring systems has been around for many years. Ritter and Koedinger suggested the possibility transitioning to Web-based tutors as early as 1996 [28] and he predicted many of the performance problems that we experience today.

Sever-based tutoring systems have many advantages over client-based approaches. Accessibility is an important concern for tutoring systems. Students, teachers, and content creators all must have access to the system. Because of widespread internet access, Web-based tutoring systems have the potential to provide access to many more users than can be reached with client-based software. Brusilovsky has claimed that Web-based systems have longer lifespans and visibility than client-based [29]. Web-based systems virtually eliminate much of time and cost of installing software on individual client machines. Another advantage of server-based architectures is greater control over content distribution. Software updates and configuration changes are easily manageable with server-based architectures. Data collection is simplified by a centralized system. In addition, the data are available immediately in the form of reports.

Adaptive Intelligent Web Based Education Systems (AIWBES) were developed as an alternative to traditional e-learning environments according to ‘one size-fits-all’ approach [19][17]. Affective tutoring systems (ATS) [23]. The system utilizes a network of computer systems, prominently, embedded devices to detect student emotion and other significant bio-signals and adapt to the student’s mood and display emotion via a life-like agent called Eve, whose tutoring adaptations are guided by a case-based method for adapting to student states - confused, frustrated or angry [1].UZWEBMAT (Turkish abbreviation of Adaptive and Intelligent WEB based

Mathematics teaching–learning system) -teaches secondary school level permutation, combination, binomial expansion and probability. [18]

This paper describes ongoing efforts and solutions for fault tolerance for online learning, which becoming increasingly important, especially web-based learning communities.

3. Architecture of an ITS

A typical ITS, has the following four basic components **[1]**. The section below lists them with their functionality, individually and then by way of their integration.

3.1 The Domain model :

The domain model (also known as the cognitive model/expert knowledge model) consists of the concepts, facts, rules, and problem-solving strategies of the domain in context. It serves as a source of expert knowledge, a standard for evaluation of the student's performance and diagnosis of errors. There are three modules within an expert system. These are the user interface which caters for smooth communication between the user and the system. The second module is the inference engine which is an interpreter for the knowledge base. It produces results and explanations for problems presented to it. The inference engine and the user interface are commonly viewed as a single component known as the expert system shell. The heart of the expert system and the final component is the knowledge base which contains the problem solving knowledge of a particular application. The knowledge base itself is isolated from the expert system shell to allow reuse of the shell in other application domains. A number of strategies for representing knowledge within the knowledge base have been explored :

- if-then rules
- if-then rules with uncertainty measures
- semantic network representations
- frame based representations

Briefly, the module interfaces with the domain knowledge. Domain knowledge embedded in the system represents an expert knowledge and problem solving characteristics.

3.2 Student Model :

The student model is an overlay on the domain model. It emphasizes cognitive and affective states of the student in relation to their evolution as the learning process advances. As the student works step-by-step through their problem solving process, the system engages itself in model tracing process. Anytime there is any deviation from the predefined model, the system flags it as an error.

The student module forms a framework for identifying a student's current state of understanding of the subject domain. The knowledge that describes the student's current state of mind is stored in a student model. McCalla and Greer (1991) suggests that in order to make any learning environment adaptable to individual learners it is essential to implement a student model within the system. The student module should permit the system to store relevant knowledge about the student and to use this stored knowledge to adapt the instructional content of the system to the student's needs. In order to identify a student's needs a number of student modelling architectures have been devised. Student diagnosis is the process to evolving the student model. In order to evolve the student model interactions between the student and the Intelligent Tutoring System need to be analyzed.

3.3 Tutoring Model :

The tutor model (also called teaching strategy or pedagogic module) accepts information from the domain and student models and devices tutoring strategies with actions. This model regulates instructional interactions with student. It is closely linked to the student model, makes use of knowledge about the student and its own tutorial goal structure, to devise the pedagogic activity to be presented. It tracks the learner's progress, builds a profile of strengths and weaknesses relative to the production rules (terms as " knowledge – tracking").

3.4 User Interface Model:

This is the interacting front-end of the ITS. It integrates all types of information needed to interact with learner, through graphics, text, multi-media, key-board, mouse-driven menus, etc. Prime factors for user-acceptance are user-friendliness and presentation. The Figure 1 presents a typical ITS architecture.

`

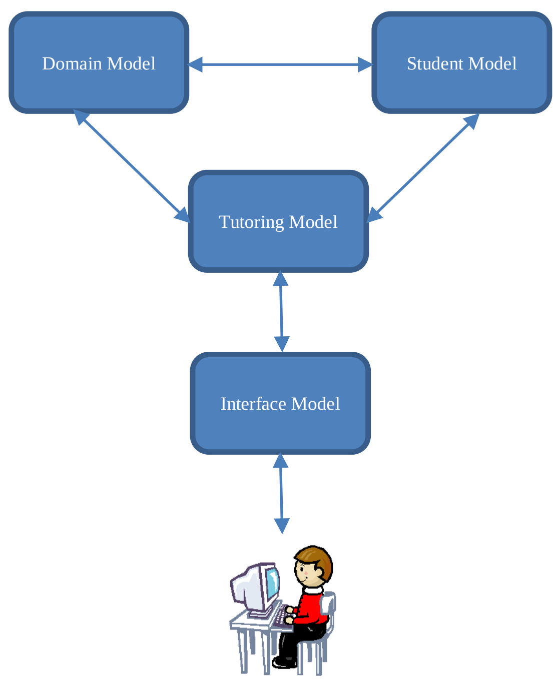

Fig. 1. Architecture of an ITS

4. A web-based of ITS

This paper provide Intelligent Tutoring System (ITS) is a software package developed for the purpose of tutoring a student in web-based leaning communities therefore researcher suggests the architecture as follows:

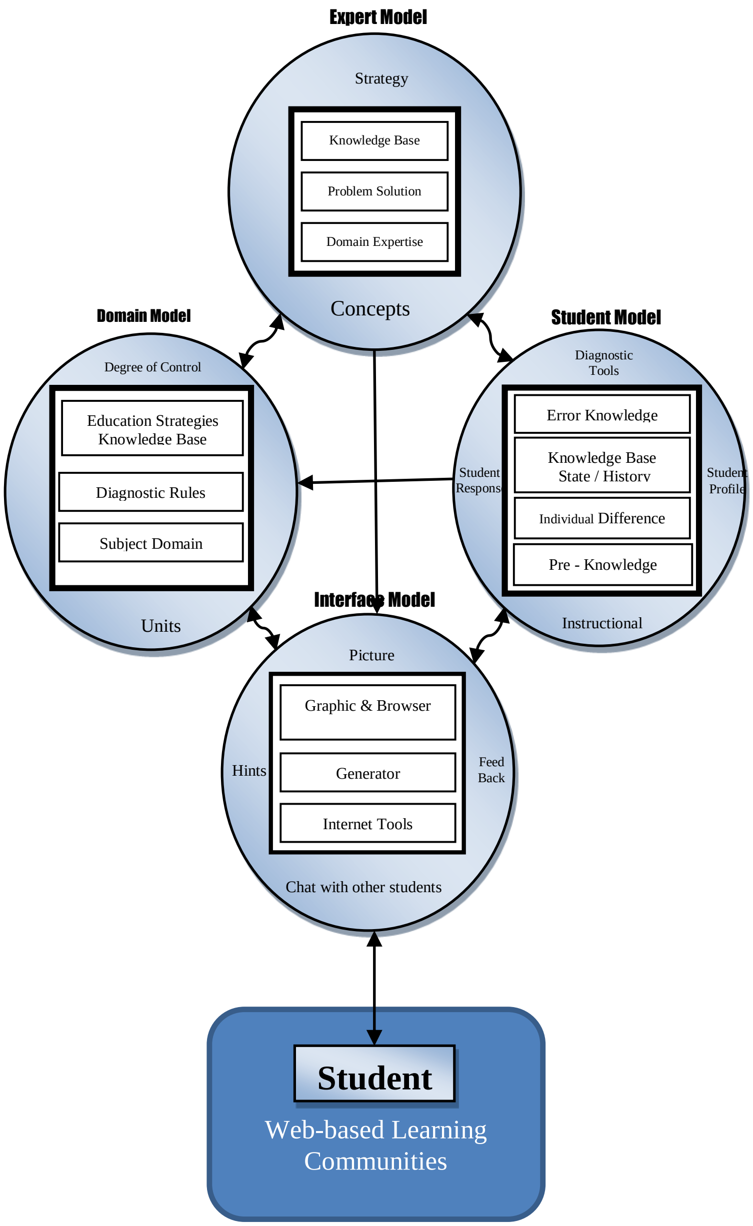

Fig. 2. New Architecture of a web-based ITS

In this section, I propose the formal model of a web-based intelligent tutoring system showed in figure 2 .The model consists of five parts. New ITS contains curriculum model with a set of cognitive states which records its cognition or knowledge of the student that a set of user states (student model) recording basic information such as personal data, student characteristics, pre-knowledge, error knowledge, etc. about the student .The student model updates its state after finishing a learning process. Due to the differences among uses' teaching strategy preferences, learning style, memory and knowledge levels unique to the expert model. The interface model make selection of a set of knowledge concepts that based on the student's knowledge of the domain, links between knowledge concepts, and student's desired detail level of the presented educational content. Evaluation of the student's performance updates the inferable student characteristics and may create a feedback for student such as tasks, life chatting with other students as a kind of learning called collaborative leaning.

Finally, the new ITS makes use of domain knowledge and teaching model to design a teaching process based its cognition or knowledge about the use of intelligent e-learning. A student state used to record information about the student vital for the system's student-adapted operation. It includes personal data that concerns information that for the identification of the student. Knowledge level such as novice, beginner, intermediate, advanced, etc. of the sub-domains and the whole domain. Interaction records which record the interaction of a student with a system. User characteristics and knowledge levels directly affect the teaching process whereas most of the interaction information indirectly.

5. Conclusion

A formal model of a web-based intelligent tutoring system is presented the system represents domain knowledge based on ontologies to improve the sharing and reusing of domain knowledge and the system constructs the user environment based on student' cognitive abilities, knowledge levels, pre-knowledge student, learning styles, psychology characteristics, etc. in order to improve the self-adaptability and pedagogical effects of the system. The student model distinguishes the information about user and what a pedagogical model. Researcher described using intelligent tutoring system to develop web-based learning communities a highly re-usable ITS framework suitable to web-based course with a set of "intelligent" functions allowing student modeling and automatic curriculum generation.

Acknowledgment

I am obliged to staff members of my college, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment. Lastly, I thank almighty, my parents, kids, sisters and wife for their constant encouragement without which this assignment would not be possible.

References

[1] A.Sarrafzadeh,, How do you know that I don't understand? A look at the future of intelligent tutoring systems, Computers in Human Behavior, Vol 24, no 4, pp 1342-1363, 2008.

[2] A.T. Corbett, & J. R. Anderson, LISP Intelligent Tutoring System Research in Skill Acquisition. In Larkin, J. & Chabay, R. (Eds.) Computer

assisted instruction and intelligent tutoring systems: shared goals and complementary approaches (pp.73-110) Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1992.

[3] A. Bailin, & L. Levin, Introduction: Intelligent Computer Assisted Language Instruction ,Computers and the Humanities, 23, 3-11, 1989.

[4] B. Buchanan, A (Very) Brief History of Artificial Intelligence. AI Magazine 26(4). Pp.53-60 , 2006.

[5] E. Fry, Teaching Machine Dichotomy: Skinner vs. Pressey. Psychological Reports (6) 11-14. Southern University Press. 1960.

[6] E. Wenger, *Artificial Intelligence and Tutoring Systems.* Los Altos, CA: Morgan Kaufmann, 1987.

[7] Gertner; C. Conati; and K. VanLehn, Procedural Help in Andes; Generating Hints Using a Bayesian Network Student Model. Articicial Intelligence, 106-111, 1998.

[8] Graesser, Arthur C., Kurt VanLehn, Carolyn P. Rose, Pamela W. Jordan, and Derek Harter. Intelligent Tutoring Systems with Conversational Dialogue. Al Magazine 22.4, 39-52, 2001.

[9] H. Anderson, & , M. Koedinger, Intelligent tutoring goes to school in the Big City. International Journal of Artificial Intelligence in Education, 8, 30-43, 1997.

[10] J .Larkin, & R. Chabay, (Eds.). Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1992.

[11] J. Chambers, & J. Sprecher, Computer-Assisted Instruction: Its Use in the Classroom. Englewood Cliffs, New Jersey: Prentice-Hall Inc, 1983.

[12] K. Anderson, Computer-Assisted Instruction. Journal of Medical Systems, 10(2), 163-171,1986.

[13] Kinshuk , Computer aided learning for entry level Accountancy students. PhD Thesis, De Montfort University, England, July 1996.

[14] Kinshuk, and Ashok Patel, A Conceptual Framework for Internet Based Intelligent Tutoring Systems. Knowledge Transfer, II, 117-24, 1997.

[15] Kinshuk. Does intelligent tutoring have future! Proceedings of the Inter. Conf. on Computers in Education, Los Alamitos, CA: IEEE Computer Society, 2002.

[16] L. Ford, A New Intelligent Tutoring System, British Journal of Educational Technology, 39(2), 311-318, 2008.

[17] M. Goyal, E-learning: Current State of Art and Future Prospects, IJCSI, Vol. 9, Issue 3, No 2, May 2012.

[18] O. Ozyurt,, Design and development of an innovative individualized adaptive and intelligent e-learning system for teaching–learning of probability unit: UZWEBMAT, Expert Systems with Applications, Vol 40 no 8, pp 2914- 2940, 2013.

[19] P. Brusilovsky, , C. Peylo, Adaptive and intelligent web based educational systems, International Journal of AI in Education, 13, pp 156–169, 2003.

[20] P.A. Cohen,, J.A Kulik, & C.L.C . Kulik, Educational outcomes of tutoring: A meta-analysis of findings. American Educational Research Journal, 19, 237-248, 1982.

[21] R. Freedman, Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. (1999) AAAI Workshop on Mixed-Initiative Intelligence, 1999.

[22] R. N. Shelby; , K. G. Schulze; D. J. Treacy,; M. C. Wintersgill,; K. VanLehn,; and A. Weinstein, The Assessment of Andes Tutor, 2001.

[23] R. W. Picard, Affective Computing, MIT Press, 1997.

[24] S. Ohlsson, Some Principles of Intelligent Tutoring. In Lawler & Yazdani (Eds.), *Artificial Intelligence and Education, Volume 1*. Ablex: Norwood, NJ, pp. 203-238, 1987.

[25] V. J. Shute, & J. Psotka, Intelligent Tutoring Systems: Past, Present, and Future. Human resources directorate manpower and personnel research division. pp. 2-52, 1994.

[26] V.J. Shute, and, J.W. Regian , Rose Garden Promises of Intelligent Tutoring Systems: Blossom or Thorn? Presented at Space Operations, Automation and Robotics Conference, June 1990, Albuquerque, NM, 1990.

[27] J. R .Carbonell, AI in CAI: an artificial intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, II, 190-202, , 1970.

[28] S. Ritter, & K. R. KoedingerAn architecture for plug-in tutor agents. Journal of Artificial Intelligence in Education,7(3-4), 315-347, ,1996.

[29] P. Brusilovsky, Adaptive Hypermedia, User Modeling and User-Adapted Interaction, v.11 n.1-2, p.87-110, 2001.