DESIGN AND DEVELOPMENT OF AN INTELLIGENT INSTRUCTIVE SYSTEM: (Scholastic Tutor (St*)

Adebiyi MARION O.
Akinboni OLUWAFUNMILAYO A.

Department of Computer and Information Sciences
School of Natural Sciences, College of Science and Technology.
Covenant University, Ota.
Ogun State, NIGERIA

ABSTRACT

Intelligent Tutoring Systems (ITS) is an act of impacting knowledge while computer teaches or acts as the tutors which is a supplement to human teachers. The ability to teach each student based on their individual abilities a major advantage posed by ITS and that is why it is being embraced in this work. This work describes the design of an Intelligent Tutoring System that was tagged Scholastic tutor (St*), which has the individual learning and collaborative problem-solving modules. The individual tutoring module was designed to provide appropriate lessons to individuals based on his/her background knowledge level, interest, and learning style and assimilation rate prior to using the tutoring system.

A software agent is used to monitor and process these parameters, arrange the learning topic, and exercises, for each individual. The collaborative problem-based tutoring module was designed to present tutorial problems and provides facilities to assist learners with some useful information and advice for problem solving. This is because the present lecturing methodology which is the conventional teaching methodology provides an interactive classroom setting that promotes the open exchange of ideas and allows for the lecturer to communicate directly with the students but has a great disadvantage of not teaching all the students according to their own learning rate and pace.

The intelligent tutor solves this problem by providing individualised learning for each student where they can learn according to their own pace and learning abilities it will provide remedy and advice when learners encounter difficulties during learning session.

The classical model of ITS architecture has four main modules; domain model, student model, tutoring model and the user interface model.

Keywords: **Intelligent tutoring system; domain knowledge base; student model; teaching modules.**

BACKGROUND INFORMATION

Artificial Intelligence is the study and design of "intelligent agents". An intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success (Gang Zhou et. al.). Education is the act of gaining knowledge while teaching is the act of impacting knowledge.

The incorporation of Artificial Intelligence (AI) techniques into education and teaching in order to produce educationally useful computer artefacts dates back to the early 1970s. By the early 1980s researchers in the already vibrant field had clearly split into two unequal camps with the emergence of two schools of thought.

The first and smaller of the two groups advocated; 'exploration environments': environments which encouraged discovery learning (i.e. learning by doing).

Perhaps the most famous is the LOGO language (Papert, 1980) which introduces students to the world of geometry through the use of robot 'turtles' and 'turtle graphics' techniques, i.e. the student learns by direct programming rather than by indirect instruction. Paper projects that 'computer presence will enable us to modify the learning environment outside the classroom so that much, if not all, of the knowledge schools presently try to teach with such pain, expense and limited success, will be learned as the child learns to talk, painlessly, successfully and without instruction.' He goes on to conclude that 'schools as we know them today have no place in the future.' Clearly, his dream is quite revolutionary: hence, he and his advocates in the LOGO camp are often referred to as revolutionaries. The second and larger of the camps is the "intelligent tutoring' group who refer to themselves as reformists as they prefer a gradual improvement (i.e. evolution) in the present quality of education using AI techniques. They advocate a paradigm where the computer acts as a tutor, i.e. students largely learn by being told.

Artificial Intelligent attempts to produce in a computer behaviour which, if performed by a human, would be described as 'intelligent': ITSs may similarly be thought of as attempts to produce in a computer behaviour which, if performed by a human, would be described as 'goad teaching' (Elsom-Cook, 1987). Intelligent tutoring system (ITS) is the electronic tutoring system that is able to serve the different needs of learners. The system is processed through artificial intelligent systems and specialized systems in managing the knowledge and analysing learners in accordance to their potential. At present, ITS have been applied more efficiently for being the supporting system for the diverse needs of learners and able to enhance the potential of the learners for better understanding of the lessons.

Association for the Advancement of Artificial Intelligence (AAAI) defined intelligent tutoring system as educational software containing an artificial intelligence component. The software tracks students' work, tailoring feedback and hints along the way. By collecting information on a particular student's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work. Ideally, an ITS tries to simulate human teacher and sometimes it may prove to be more advantageous than its human counterpart. One of the main advantages of ITS is individualized instruction delivery, which means the system will adapt itself to different categories of students.

A real classroom is usually heterogeneous where there are different kinds of students, from slow learners to fast learners. It is not possible to provide attention to them individually, thus the teaching may not be beneficial to all students. An ITS can eliminate this problem, because in this virtual learning environment the tutor and the student has a one-to-one relationship.

The students can learn at her/his own pace. Another advantage is that using this system teaching can be accomplished with minimum intervention from the teachers. Therefore, ITS can be really effective in areas where there is dearth of trained teachers.

REVIEW OF RELATED LITERATURE

The first generation of computer assisted education tools were called, Computer-Aided Instruction (CAI) systems. One of the early examples of such a tutoring system is the system by Uhr in the year 1969 (Sleeman & Brown, 1982). This system generated problems on arithmetic and questions on vocabulary. But main problem of this system were it had no user modelling or adaptation technique.

However, some contemporary systems, like (Suppes et al., 1967, Woods & Hartley, 1971, Sleeman & Brown, 1982) could be called adaptive because here the problems were according to the user's performances. But the user model they used was quite primitive in nature. The model was only a parametric summary; the actual knowledge state of the user was not stored. These systems can be termed as the precursor to Intelligent Tutoring System (ITS). In the meantime, another genre of tutoring system came up. These types of systems were called *Drill and Test*. Here only problems were presented to the students in form of tests. The students on the other hand are provided with the test results. A simple variation of this system was the Adaptive Drill and Test. In this version instead of just presenting the problems, the student's performance and response were collected, tabulated and later used to select problems. Thus at this point of time, it was felt that the student needed to be considered as an important factor, and no longer predetermined rules will work. An adaptation technique was quite necessary to tackle all possible responses from the students.

Emergence of ITS

The existence of the "two-sigma problem," which states that students who receive one-on-one instruction perform two standard deviations better than students who receive traditional classroom instruction (Bloom, 1984). It is impossible for any institution to provide personal teachers to each and every student. This drawback strongly supported the use of computers, as a replacement to human teachers. Motivated by these reasons, lots of research group started to work in this field and developed various systems with various features. In 1982, Sleeman and Brown reviewed the state of the art in computer aided instruction and firstcoined the term Intelligent Tutoring Systems (ITS) to describe these evolving systems anddistinguish them from the previous CAI systems. They defined ITS as being computer-based (1)problem-solving monitors, (2) coaches, (3) laboratory instructors, and (4) consultants. (Sleeman&Brown, 1982). And for the first time the use of Artificial Intelligence was seen, which made the systems "intelligent". At this point of time emerging trends in Artificial Intelligence were applied in these systems. With new AI techniques coming up it seemed that the computers were almost capable of"thinking" like humans. This motivated ITS research further. Application of AI in ITS made it possible to achieve the goals more easily.

Review of Existing Systems

The teaching methodology of an existing system is based on the method proposed by Fissilis*et al.*, 1996 [4]: after studying a certain unit, the intelligent tutor sets a test. If the student passes the test, they go on to the next unit; otherwise, they are provided with a remedial unit with its corresponding test. If the student fails, the process continues until all the remedial units have been failed, at which point the student will repeat the process from the initial remedial unit. If the student passes the remedial unit, they will return to the lesson where they committed the original error to repeat the test they failed. At the end of the course, the students are examined on all the learning units so that they can be graded by the teacher. This educational process provides the student with a certain degree of flexibility (whether to go back to study the units again and/or do the corresponding tests).

Andes

Andes (Conati et al, 2002; Gertner & VanLehn, 2000) is an ITS which was developed to teach physics for the students in Naval Academy. Bayesian networks were primarily used in Andes for decision making. The major foci of the system are;

- **Select the most suitable strategy for the student**
- **Predict Student's actions**
- **Perform a long term assessment of the student's domain knowledge.**

Andes is a domain dependent ITS. Each problem in the system was broken into some steps and Bayesian network was formed using those steps as nodes. So, the problems were represented in the system as Bayesian networks. The Bayesian network would predict the most probable path for the student during a course. Each student could have different approaches to a problem, the network would be adjusted accordingly (the probabilities would change) and finally for a new problem it would predict the best strategy for the student. There is also a problem-solver in the system. This problem solver partially or wholly solved a problem to help the students. The Bayesian networks had two parts: static and dynamic. The static part had Rule nodes and Context-rule nodes. The rule node represented general physics rules and had binary values, T and F.

The probability P(Rule=T) was the probability the student could apply the rule properly in any situation. Initially these prior probabilities had values 0.5 but the authors claimed more realistic values could be obtained after a trial run in the naval academy. The dynamic part contained the Context-rule node as well as four other nodes: fact, goal, rule application and strategy-nodes. The fact and the goal nodes were also binary. P(Fact=T) was the probability that the student knew the fact and P(Goal=T) was the probability that the student is pursuing the goal. There might be more than one way to reach the goal or the fact nodes and that lead to having so many parents. The conditional probabilities P (Fact=T| parenti) represented the probability of reaching the fact from parenti.

The strategy nodes were there, where, students had more than one options to choose from. And the rule application nodes represented the children of those strategy nodes. The rule application nodes basically represented the different applications of the strategy nodes. The strategies were mutually exclusive; i.e. the student could choose one and exactly one strategy at a time. Thus if an evidence increased the probability of one of the strategies it would definitely decrease the others. The probability values (P (Strategy-node = {x: x ε {child1, …,childn}}) of these nodes would depend upon the number of children the node had.

Finally the Rule-application nodes were the connectors between context-rule nodes, Strategy nodes, fact and goal nodes to new derived Fact and goal nodes. In other words, those nodes had a single Context-rule node and strategy node and one or more than one fact and goal nodes (preconditions) as parents, and children of those nodes included some facts and goal nodes (Conclusion).

They had binary values and P(R-A=T) meant the probability of a student applying the parent Context rule to the preconditioned fact and goal nodes to get the derived fact and goal nodes.

The probability values would vary with students and thus application of rules, choosing from alternate paths etc would depend upon each student. But how the probabilities were derived was not stated.

SQL-Tutor

SQL-Tutor (Wang & Mitrovic, 2002; Mitrovic, 2003) is an ITS, which as the name suggests is to teach SQL. Artificial Neural Network (ANN) was used in SQL-Tutor for decision-making. An agent was present who analysed the student and selected an appropriate problem from the database. That agent was modelled using an ANN. This ITS was developed to teach university students SQL. Here the solutions to the problems were represented in the system as constraints.

Whenever a student submitted a solution the system calculated the correctness by comparing the number of constraints violated by the student. The next problem to be chosen or any other teaching decision to be taken depended on this information, how many mistakes or violated constraints the student has committed. To make this prediction the system used an artificial neural network (ANN). The ANN was a feed-forward network and had four inputs, one output and a hidden layer. Delta-bar-delta back propagation and linear tanh (LT) transfer function was used and the inputs consisted of different information relating to the student like:

- **Time needed to solve the problem**
- **The level of help provided to the student**
- **The complexity of the problem.**
- **Also, the level of the student in hand.**

In the output the ANN tries to predict the number of errors (i.e. the no. of constraints violated) will be committed by the student. This prediction is used to take the next teaching decision (like the next problem to choose from the problem database). However how the weights of the ANN were chosen and how exactly the ANN was trained were not explained clearly. About the performance of the system, the authors claimed that the ANN could predict the correct number of errors with an accuracy of 98.06%.

An added advantage of this system was it provided feedbacks to the student after checking her solution. The feedback might contain hints, partial solution or complete solution as required.

C++ Tutor

C++ Tutor is a rule-based ITS (Baffes & Mooney, 1996). Here the concepts of C++ were explained using some rules. These rules were in form of Horn sentences and were called the Theory. The problems were produced to the students in the form of feature vectors. The students were supposed to label the vectors choosing from a set of labels. An algorithm called NEITHER took these labelled vectors (student's solution) as input and modified the correct rule base, so that the modified rules implied the student's solution rather than the correct solution. This process is called Theory-revision.

So now the modified rule base reflected the student's state of understanding, representing the student's correct knowledge as well as the misconceptions.

After the theory-revision was complete the system tried to explain the bugs in student's concept by showing examples, which enumerated the areas where the student had gone wrong. This was done automatically by the system, by comparing the modified rules with the correct ones.

STATEMENT OF THE PROBLEM

The traditional classroom teaching methodology has remained rigid and does not consider most of the lecturer's mood and some of the student's issues such as;

- **Difference in the learning/ assimilation rate of individual students.**
- **Limited teaching period for each lecture.**
- **Inability to conduct practical sessions and tests after each lecture.**
- **Absence of supervised learning for each student during teaching, especially the large classes.**

AIM AND OBJECTIVES OF THIS RESEARCH

The aim of this work is to develop an intelligent system called Scholastic tutor(St*) that is capable of teaching and emulating an instructor's all round behaviour aside his mood, and helps students in acquiring knowledge. The objectives of Scholastic tutor (St*) are:

- **To improve the grades of students by providing individualised learning for each student.**
- **To facilitate understanding and knowledge through the use of visuals in lecture delivery.**
- **To carry out tests after each module to ensure that students understand the module.**

RESEARCH METHOD

The system was implemented using the following tools: Java Expert System Shell (JESS) for frontend, MYSQL as the DBMS for backend, JAVA, HTML (Adobe Dreamweaver (CS4) and Apache server as the middleware.

RESULT AND DISCUSSION

The system developed tagged Scholastic tutor (St*) has the following modules after design and development:

- **The expert knowledge module: it is also known as the Domain module. It provides the knowledge that the student will be taught, and consists of declarative knowledge (lessons, tests, exams, etc.) and procedural knowledge (sets of rules to execute a task). The expert knowledge module comprises the facts and rules of the particular domain to be conveyed to the student, i.e. the knowledge of the experts.The domain model has two major parts. The firstis called the Domain Organization Model, it is the knowledge base of the system. The other is theRepository, which stores the metadata annotated learning and test materials.**
- **Student module: this records information about the student (personal information, interaction and learning process parameters). It is the storage part that stores the information of each learner, as well as follow them up in order to store the feedback from the leaner in each system.**
- **Tutoring Module: the tutoring module can be divided into two models which are the;**
  - ***Teacher model*: this records information about the teacher (such as their personal information).**
  - ***Teaching model*: this defines the students' learning cycle. For this, it adjusts the presentation of the material to each student's knowledge according to the information contained in the student model. The teaching model comprises seven modules: evaluation model, problem generation model, problemsolving model, and model for analysing students' answers, model for generating plan of revision units, model for predicting students' grades, and the syllabus generation model.**

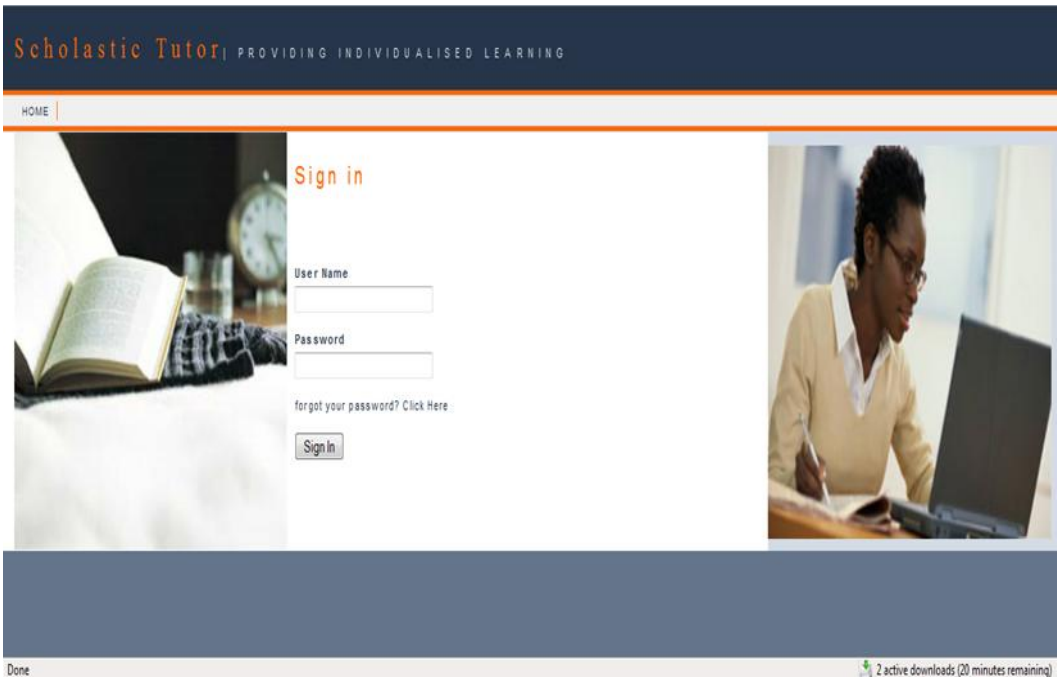

Figure: 1
GUI for the sign in page of Scholastic tutor

The teaching model is the pedagogical agent of the system and it executes the actual teaching process. In logical sense this module lies in the centre of the system and by communicating with the other modules it provides adaptive instructions to the students. It is rule-based, where the rules define the teaching strategy. The rules are author-able that enables the teaching model to be customized. The features of the authoring tool have been extended to support the modification of the teaching model. A domain expert can utilize this facility and apply her skills in reorganizing the teaching method to suit in different environments.

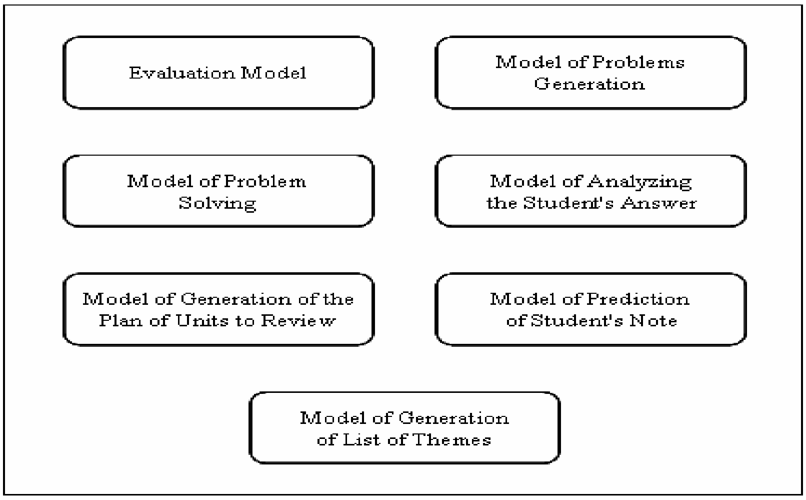

Figure: 2
The Teaching Module Adopted for scholastic tutor

- **Graphic interface: this is responsible for user interaction with the intelligent tutoring system.It controls the communication between the learners, teachers, and the system as well as following up the behaviours of them. Then the information is sent back to the student module afterwards. IT is the gateway ofthe system for communicating with the student. All the learning materials and test sets are presented tothe student through this interface and test results are fed back to the system for assessing the student.**

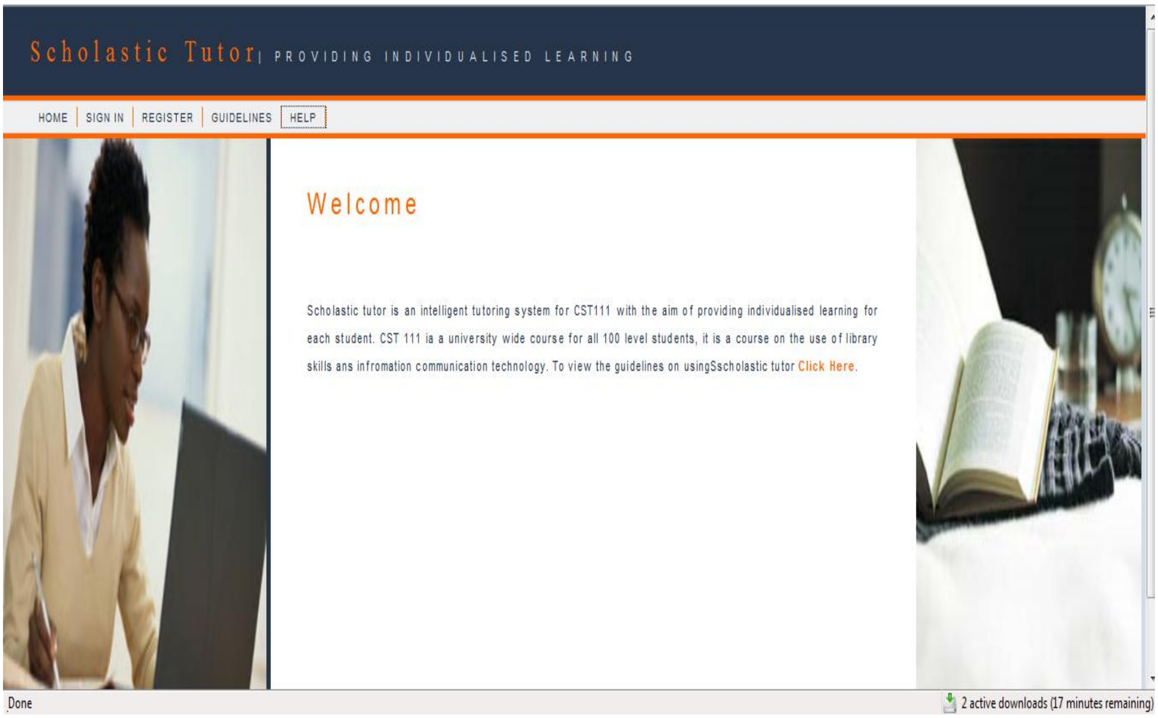

Figure: 3
Graphic user interface for the Home page of Scholastic tutor

CONCLUSIONS

St* proposes the framework for constructing a system to be used as an intelligent tutorial system for individual and collaborative learning. It supports both individual and collaborative learning and can be utilized through the web and served for real time eLearning. St* can run its individual learning module in anyplace and anytime, as well as the collaborative learning, each learner can switch back to the individual module at any time as well from the collaborative learning.

BIODATAand CONTACT ADDRESSES of AUTHORS

Adebiyi MARION O. is a PhD student in the Department of Computer and Information Sciences Covenant University, Ota, Nigeria. She holds B.Sc. and M.Sc. in Computer Science (Bioinformatics option). Her current research interests are in the following areas: Computational Elucidation of Drug Resistance Mechanisms, Protein and Structure Prediction, Biochemical pathway Analysis (Proteomics) and Data communications and Networking.

She is a member of a number of both international and local professional bodies such as International Society for Computational Biology Students Council (ISCBSC), Africa Society of Bioinformatics and Computational Biology (ASBCB). She is currently the President of West African Region of ASBCB (RSG WA) she also belongs to the Nigerian Computer Society (NCS), Computer Professional Registration Council of Nigeria (CPN) Association of Storage Networking Professionals (ASNP) and Association of World Scientific Computing Club.

Adebiyi MARION O.
Department of Computer and Information Sciences
School of Natural Sciences, College of Science and Technology.
Covenant University, Ota. Ogun State, NIGERIA
Tel: +23408033853585,
Email: onlimarion@yahoo.com

Akinboni OLUWAFUNMILAYO A. is a dedicated 400Level student in the Department of Computer Science and Information Sciences at Covenant University. She is well experienced in Programming and Database Management. A Highly motivated, innovative, resourceful and growth oriented person, able to adapt effectively to a new, dynamic and challenging environment. She is able to work without supervision and with initiative, whilst efficiently contributing as a team member. She possesses excellent analytical, presentational and strong communication skills.

Akinboni OLUWAFUNMILAYO A.
Department of Computer and Information Sciences
School of Natural Sciences, College of Science and Technology.
Covenant University, Ota. Ogun State, NIGERIA
Tel: +23408023059476,
Email: abisoye4christ@yahoo.com

REFERENCES

C. Koutsojannis, J. Prentzas, I. Hatzilygeroudis. *A web based intelligent tutoring system teaching nursing students fundamental aspects of biomedical technology.* Proceedings of the 23rd annual EMBS international conference in Estambul (Octubre 2001).

Gang Zhou, Jason T.-L. Wang, Meter A. Ng. *Curriculum knowledge representation and manipulation in Knowledge based tutoring systems. IEEE Transactions on knowledge and data engineering,* vol. 8, no 5, octubre1996.

Hyacinth S. Nwana *Intelligent Tutoring Systems: an overview* Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK

M. Negnevitsky. (1998).*A knowledge based tutoring system for teaching fault analysis. IEEE transactions on power systems,* vol. 13, no. 1 Febrero 1998.

P. Fissilis, A. Kameas, P. Pintelas. (1996). *ORIENTMAN: an intelligent tutor for the ORIENT software development methodology. Software Engineering Journal*, Julio.

Skinner, B, F. (1958) *Teaching Machines. Science*, 128,969-977.

Sleeman, D. H. (1975) *A problem-solving monitor for a deductive reasoning task. InternationalJournal of Man-Machine Studies*, 7, 183-211.

Sleeman, D. H. (1983) *Intelligent tutoring systems: a review*. Proceedings of EdCompCon '83meeting. IEEE Computer Society, pp. 95-101.

W. Fajardo Contreras1, E. Gibaja Galindo*2, E. Marín Caballero1 and G. Marín Caballero, *An Intelligent Tutoring System for a Virtual E-learning Center*Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S. de IngenieríaInformática,Universidad de Granada, 18071 Granada, Spain. Departamento de Informática y AnálisisNumérico, Campus de Rabanales, Edificio Albert Einstein,Universidad de Córdoba, 14071 Córdoba, Spain.SunandanChakraborty, Devshri Roy, AnupamBasu*, *Development of Knowledge Based Intelligent TutoringSystem.*

James, O. & Sowmya R. (n.d.). *Intelligent Tutoring Systems: Using AI to Improve Training Performance and ROI.*

Reye, J. (1996). *A Belief Net Backbone for Student Modeling*, Proceedings of third InternationalConference on Intelligent Tutoring Systems, June 12-14, 1996, pp. 596-604.

Reye, J. (1998). *Two phase updating of Student Models based on Dynamic Belief Networks,Proceedings of fourth International Conference on Intelligent Tutoring Systems,* Aug 16-19, 1998, pp. 274-283.

Ross, T. J. (1997). *Fuzzy Logic with Engineering Applications*. McGraw-Hill, 1997.

Sleeman, D., & Brown, J. S. (1982). *Introduction: Intelligent Tutoring Systems. Intelligent TutoringSystems,* D. Sleeman, J. S. Brown, Ed. Academic Press, 1982, pp. 1-11.

Thompson, J. E. (1996). *Student Modeling in an Intelligent Tutoring System. MS Thesis.*

VanLehn, K., Martin, J. (1997). *Evaluation on an assessment system based on Bayesian studentmodeling.*International Journal of Artificial Intelligence in Education, Vol. 8.

Villano M. 1992. *Probabilistic Students Models, Bayesian Belief Networks and Knowledge SpaceTheory,* Proceedings of the second International Conference on Intelligent Tutoring Systems, June 10-12, 1992, pp. 491-498.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems.* Los Altos, CA: Morgan Kaufmann.

Xu, D., Wang, H., Su, K. (2002). *Intelligent Student Profiling with Fuzzy Models,* In Proceedings of the35th Hawaii International Conference on System Sciences, Hawaii, USA (2002), p. 81b.

Zapata-Rivera, D. & Greer, J. (2004). *Interacting with Inspectable Bayesian Student Models.* InternationalJournal of Artificial Intelligence in Education, Vol 14. pg., 127 – 168

Zhongping H. & Naoyuki T. *A syntactical approach to diagnosing multiple bugs in an intelligent tutoring systems. IEEE Transactions on systems, man and cybernetics, part*A: Systems and humans, vol. 26, no. 2, Marzo 1996.

Zhou, Y., Evens, M. W. 1999a*. A Practical Student Model in an Intelligent Tutoring System*, ICTAI.