A Multiagents based Intelligent Tutoring System for teaching Arabic Grammar

MONA HAFEZ MAHMOUD
Informatics Research Department
Electronic Research Institute
El-Tahrir st., Giza
EGYPT
Monah1957@hotmail.com http:\\www.eri.sci.eg

Abstract

Intelligent agent has been around for years, but the actual implementation is still in its early stages. This research is a scientific mix between two big topics of Artificial Intelligence. These topics are: the Intelligent Agents and the Intelligent Tutoring System. An Intelligent Agent is a set of independent software tools that are linked with other applications and database software running within a computer environment. The primary function of an Intelligent Agent is to help a user (client) to better interact with a computer application. It is assumed that artificial intelligence (AI) is involved and certain degree of autonomous problem solving ability is presented in agent-based technology systems[1].

Intelligent Tutoring Systems (ITSs) simulates the one-to one human tutor for delivering knowledge interactively instead of using books and the traditional learning environment. To come up with the most learning outcomes, ITSs have incorporated several techniques such as: error identification and correction, and building consistent explanations through integrating techniques of cognitive science and Artificial Intelligence. Different tutoring systems have been implemented to cover different subjects and languages such as: English, Arabic, Chinese, German and many others [2]. In this research ITS is covering grammar of Arabic language. The global structure of ITS consists of mainly four modules: a pedagogic module, a question selector module, an expert module and a student module in addition to a user interface module. But in this system we didn't need an expert module because we used Constraints Based Model (CBM) technology (that will be explained below). This work is implemented under a project that is called AG_TUTOR (Arabic Grammar tutor). This project simulates the behavior of instructors and students and the relations between them in teaching the course of the Arabic Grammar of the fourth grade of the elementary stage in Egypt. In this system the technology of Intelligent Agents is used. This research concentrates on the Intelligent Agents part of AG_TUTOR.

Keywords Terms — Artificial Intelligence and education, Intelligent Tutoring System, Intelligent Agents, Multi-Agents systems, knowledge base, domain knowledge.

1. Introduction:

In computer science, an intelligent agent is a computer program that acts for a user or other program in a relationship of agency. In particular, exhibiting some aspect of artificial intelligence such as learning and reasoning are related and derived concepts include intelligent agents. There are many types of intelligent agents: autonomous agents (capable of modifying the way in which they achieve their objectives), distributed agents (being executed on physically distinct computers), multi-agent systems (distributed agents that do not have the capabilities to achieve an objective alone and thus must communicate), and mobile agents (agents that can relocate their execution onto different processors).**[3]**

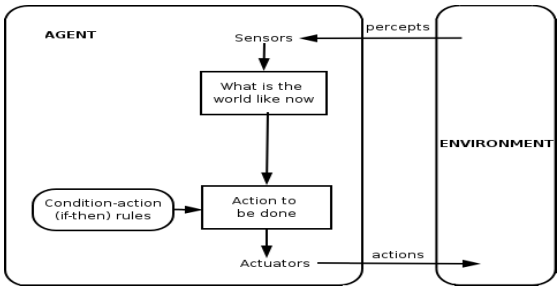

Fig. (1) Simple autonomous agent

As shown in Fig. (1), an intelligent agent (IA) is an autonomous entity which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals. Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex. [4]
A simple agent program can be defined mathematically as an agent function which maps every possible percepts sequence to a possible action the agent can perform to a coefficient, feedback element, function or constant that affects eventual actions: [4]

$$f : P^* \rightarrow A$$

Intelligent Tutoring Systems (ITSs) are complex computer programs that manage various heterogeneous types of knowledge ranging from domain to pedagogical knowledge. ITS typically consists of:
1. The Pedagogic module, which designs and regulates instructional interactions with the students;
2. The Question Selector Module, which selects a question from a question bank, presents it to the student and gets his response;
3. The Expert module, which simulates human experts in decision making or the instructor in education to get the correct answer of the question that presented to the student. (we didn't need this module because we used CBM that will be explained later).
4. The Student Module, which is a dynamic representation of the students current state of knowledge;
5. The user Interface module, which controls interaction between the student and the system [5].

2. Related work:

Here are some examples of systems that use the intelligent agent's technology:

- ***FlexiTrainer:*** it is an authoring framework that allowed a fast design of pedagogically rich and performance-oriented learning environments with tradition content and tutoring strategies. This authoring tool specifies a dynamic behavior of tutoring agents that interact to deliver instruction. FlexiTrainer has been used to develop an ITS for training helicopter pilots in flying skills. It consists of two components: the authoring tools and the routine engine. Core components for it are Task-Skill-Principle Editor, Exercise Editor, Student Model Editor, and Tutor Behavior Editor. Each of these editors has their own specific functionality. An instructional agent is used to carry out instructional goals. It used Bayesian inference to incorporate student modeling strategies.[6]

- ***MathTutor:*** it is a multi-agent ITS building tool. Math Tutor integrates different formalisms in order to facilitate the teacher task of developing the contents of a tutorial system and at the same time to provide adaptively and flexibility in the presentation. Multi-Agent Systems (MAS) technology have been of great help in reducing the distance between ideal systems and what can really be implemented, because it allows to simplify the modeling and structuring tasks through the distribution, among different agents of the domain and student models. The proposed tool is based on a conceptual model, called MATHEMA that provides a content-directed methodology for planning the domain exposition and teaching strategies. [8] [7]

- ***Popular Tetris computer game***: In this game a user must try to make a wall out of irregularly shaped falling blocks. The agent in the game takes the part of the user, who must control where the blocks fall. Using traditional AI techniques would require representing knowledge about the game and the role of the user in terms of symbolic data structures such as rules, and so on. This approach would be entirely unrealistic for a game like Tetris, which has hard real-time constraints. Wavish and colleagues thus use an alternative reactive agent model called RTA (Real Time Able). In this approach, agents are programmed in terms of behaviors which are simple structures. These agents are loosely resemble rules but do not require complex symbolic reasoning.[9]

- ***UCEgo*** is a natural-language system that helps the user to solve problems in using the UNIX operating system. It is the intelligent agent component of UC (UNIX Consultant). UCEgo provides UC with its own goals and plans by adopting different goals in different situations. It creates and executes different plans, enabling it to interact intelligently with the user. Also, it adopts goals from its *themes*, sub-goals during planning, and *meta-goals* for dealing with goal interactions. It also considers goals when it

notices that the user either lacks necessary knowledge or has incorrect beliefs. In these cases, UCEgo plans to volunteer information or correct the user's misconception, as appropriate.[10]
The organization of this paper will be as follows: section 3 presents the Domain Knowledge, section 4 presents the Knowledge Base while section 5 presents the general structure of AG_TUTOR, section 6 shows the Multi Agent System in AG_TUTOR, Finally, section 7 concludes the whole work.

3. Domain knowledge:

Domain knowledge in artificial intelligence is the knowledge about the environment in which the target system operates. The domain model organizes the course structure, its various components and the relationship among the components. This model mainly deals with the what-to-teach part of an ITS.[11] A domain model is created in order to represent the vocabulary and key concepts of the problem domain. It also identifies the relationships among all the entities within the scope of the problem domain, and commonly identifies their attributes. An important advantage of a domain model is that it describes the scope of the problem domain.[12] The adopted domain is the curriculum of the grammar of the Arabic language of the fourth grade of the elementary schools in Egypt. The knowledge of this curriculum is acquired from the Arabic instructor transcripts. Each lesson of the curriculum is considered a concept. Each concept may have sub-concepts. Specifically, they cover the following concepts and sub-concepts:

demonstrative nouns, pronouns (1st pronoun, 2nd pronoun, 3rd pronoun), speech (Nouns, verbs, particles), dual, plural, nominal and verbal sentence, Interrogative and Negative tools and agreement of verb with the subject in gender.

أسماء الإشارة,الضمائر (ضمير المتكلم, ضمير المخاطب، ضمير الغائب), الكلام (الأسماء، الأفعال، الحروف)، المثنى, الجمع،الجملة الفعلية, الجملة الإسمية , أدوات النفى والإستفهام ، توافق الفعل مع الفاعل فى التذكير والتأنيث.

Our course domain is richly articulated in topics and subtopics (or concepts and sub-concepts). It is required that each question in the domain is attached to exactly one course topic or sub-topic. So, a method based on course structure is used. It uses a structure called a prerequisite structure that defines each course topic which other topics the student must already know before proceeding further.

4. Knowledge base:

A **knowledge base** (**KB**) is a technology used to store complex structured and unstructured information used by a computer system for artificial intelligence domain. A knowledge-based system consists of a knowledge-base that represents facts about the world and an inference engine that can reason about those facts and use rules and other forms of logic to deduce new facts or highlight inconsistencies. [13]
Two relational databases are used in AG_TUTOR which are implemented using mySQL. One of them is considered the "lexicon" of the system which contains all the words (nouns, verbs, particles) of the exercises that will be represented to the student and their features. The other one includes a bank of questions. Also, it includes all constraints, skills of the student, feedback messages and all information about the student and his knowledge.

5. The general structure of AG_TUTOR:

In our proposed system, the system will deal with group of intelligent agents or multi-agent systems that deal with the modules specially student model. These agents are used for learning and reasoning, also for modifying the learning strategy, so every student can have a different learning strategy according to his individual problem diagnosis, or according to the type of the student (such as talent, smart, shy, slow or fast in understanding and so on.....). Each module of the Educational system will deal with one or more of intelligent agents. As will be seen later, each one or more of the system agents will represent an environment in the system such as the student, the teacher, the learning process and relations between them. Fig.(2) illustrates the structure of an ITS.

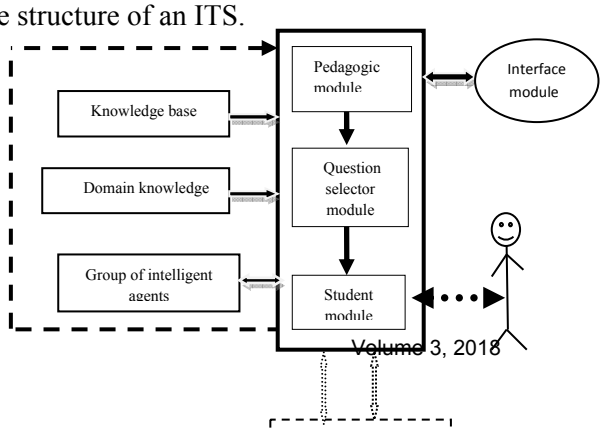

Fig. (2) The structure of AG_TUTOR

5.1. Pedagogic module:

The Pedagogic Module is a computer tutor that mimics the course patterns and educational tactics of a real human tutor [2]. It is the instructional module that designs and regulates the instructor transcripts. The function of the tutoring module is essentially to perform continuous assessment of the student, and thereby interact with the expert module to prescribe further action. [13][14]

In AG_TUTOR, this module is representing the concepts in a very attractive interface with high quality of graphic, animation and sound. Also, it represents group of examples for each concepts. At the case the student's answer is wrong; the system will take the student back to the tutoring module to give him more explanation about the concept. This module deals with Teaching Assistance Agents (that we will talk about later) that helps in the teaching process for all the lessons in our curriculum.

5.2. Question selector module:

The main goal of the question selector module is to select a question randomly according to the lesson that the student selects, display it to him and give him the chance to answer. [15]

This module drives these questions from the question bank in the data base. The question bank contains a huge number of questions. The bank is divided into many groups of questions as a group for each lesson mainly:

1. Multiple Choices Questions (MCQ)
2. Match the related correct sentence
3. Press on something (like اسم إشارة أو ضمير مخاطب)
4. Fill in the space with the correct answer from the brackets
5. Get out a verb, a noun, or a particle or ……..
6. Parse a sentence
7. Reorder a nominal sentence to be a verbal sentence and vice versa.
8. Generate the plural, double or single of a noun

In this module, Constraints and Hints Agent (that will be explained later) is helping in the selection of the question from the data base and presenting to the student.

5.3. Student module:

A Constraint Based Modeling (CBM) system is adopted in implementing this module. The concept of state constraints was invented to solve a deep puzzle about skill acquisition: Human beings can catch themselves making errors. This ability forces a distinction between generative and evaluative knowledge. The function of generative knowledge (e.g., a rule set) is to produce actions according to the current problem. And, the function of evaluative knowledge (a set of constraints) is to evaluate action outcomes as desirable or undesirable. Different learning theories have different implications for the design of ITSs. The state constraint theory suggests that the knowledge base of a constraint-based tutoring system should contain the constraints that the student would have. Hence, such a tutoring system plays the role of an amplified evaluative knowledge base. [16]

The CBM is represented by a set of constraints; each constraint represents a pedagogically significant state [17]. The basic definition of a constraint is formalized as:

<Constraint-id> <relevant condition> < Satisfaction condition > <feedback action>

Where the relevant condition is the condition that represents situations where constraint applies, satisfaction condition is the condition that has to be true in order for the constraint to be satisfied, feedback actions is the action associated with the violation of the constraint. Constraint-based modeling has many benefits mainly:

- Decreasing the time required to build an ITS by providing detailed and specific feedback associated with the constraints.
- The incorrect answers are implicitly implemented in the constraints, so no need to implement them in the domain model in form of buggy-rules like model tracing.
- Changing any constraint in CBM has no effect on the other constraints at all.
- No need to the Expert Module to get the correct answer.

For modeling the student knowledge or skill in the linguistic domain, the constraint form is modified to be as following:

<Constraint-id> <Skill> <domain concept> < Satisfaction condition > <Positive feedback message> <Negative feedback message>

The relevance condition is replaced by the skill and domain concept fields to categorize the constraints according to the different skill and concepts and facilitate the testing of relevance condition. Moreover, we augment positive feedback to encourage the student in the case of correct answer by a suitable feedback. Also, a negative feedback message is used to indicate the violated constraint that represents the student's error. Constraint based modeling has the responsibility of testing the satisfied condition constraints against the student answer to define the satisfied and violated constraints.

Student model records the history of each constraint. This record implies information about satisfaction or violation of the relevant constraints against the student answer and describes the student confidence level for each constraint. This information can be used by the pedagogical module to adapt the question selection.

This module deals with the Constraints and Hints Agent (that we will talk about later) for helping in the evaluating the student answers, diagnose his problem and treat it.[18]

5.4. Interface module:

Generally, the interface module has two types: interface-as-agent and interface-as-tool. In the interface-as-agent view, it is seen as a separate entity that mediates between the user and the machine. The student tells the interface what to do, and then the interface acts on these instructions. In the competing interface-as-tool view, the interface is seen as directly representing the machine. The interface merely facilitates the user's direct manipulation of the machine. In this view, all student actions should have immediate, directly perceptible results.[19]

In AG_TUTOR high graphic, sound and animated interface-as-agent is used. The interface of each module is suitable for it. In the interface of the student module, the course structure with the states of each concept appeared (as mentioned before the red, green and black states). The state of each concept is defined by student performance in the exercises that are represented to student (this interface is under implementing). In Fig. (3) there are snapshots from the pedagogic module and the question selector module. The Interface module is dealing with the interface agent (that will be explained later) to communicate with the student.

(a) A snapshot from the pedagogic module

(b) A snapshot from the question selector

Fig. (3) A sample from the Interface Module of the system

6. The multi intelligent agents system in AG_TUTOR:

Advantage of using Intelligent Agents in Education is representing a powerful tool for making system more flexible. Agents should behave like an 'expert assistant' with respect to some application, knowledgeable about both the application and the user, and capable of acting with user in order to achieve the user's goals. Agents are also good in improving the efficiency of Software Development. [20]

6.1. Types of Agents in the AG_TUTOR:

A multi agents system is used in AG_TUTOR. This system consists of 5 agents, each one deals with one or more modules of the Educational system. Fig. (4) shows the types of Agents in AG_TUTOR and how each one deals with the modules of the Educational system. These types are:

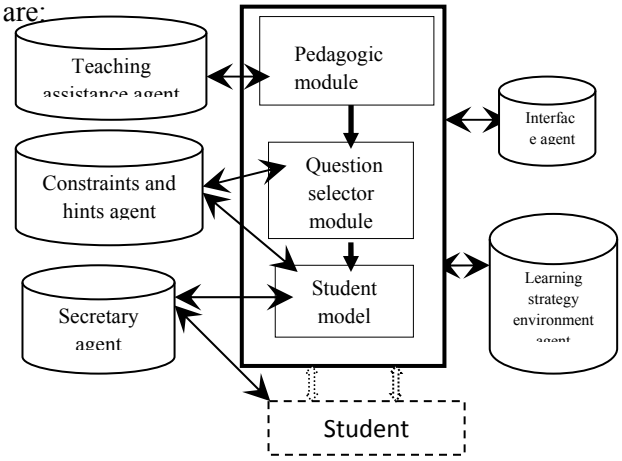

Fig. (4) Types of Agents in the AG_TUTOR

- *Teaching Assistance Agents:* that simulates the instructor in a very attractive form. It deals with the Pedagogic module to monitor the course to the student. The teaching agent does not interact directly with a student but through personal agent and serves as an intelligent tutor. The basic components of a teaching agent are a domain expert module and pedagogy.
- *Constraints and Hints Agent:* that simulates student. It deals with the student module and the question selector module that used Constraint Based Modules (CBM) technology (as explained before). It deals with the student model to define which constraint he violated. Then, it chooses the hints that will be appeared for each mistake he falls down in, as a guide to arrive to the correct answer with him.
- *Learning Strategy Environment Agent:* that simulates the educational process and helps to make the adaptation to change the learning strategy according to the student's level and mistakes.
- *Secretary Agent:* that represents the secretary of the system; it deals with the student module and student record. It helps to fill the student's profile with his/her personality data and the student's status information. However, before allowing for navigation, the secretary agent (personal agent) of the system checks the past record of the performance of the student to present the right content. If the performance of the student is not satisfactory then the appropriate remedial content is presented to the student. Only after the learning objectives are achieved, the student is allowed to move to a new topic. This is done with the help of teaching assistance agent.
- *Interface Agent:* that controls the communication between the learners, teachers, and the system as well as following up the behaviors of them. Then the information is sent back to the student module afterwards.

6.2. The structure of an agent in the AG_TUTOR:

This pattern has been abstracted out of a number of agents playing different roles in different educational settings, yet having much in common.

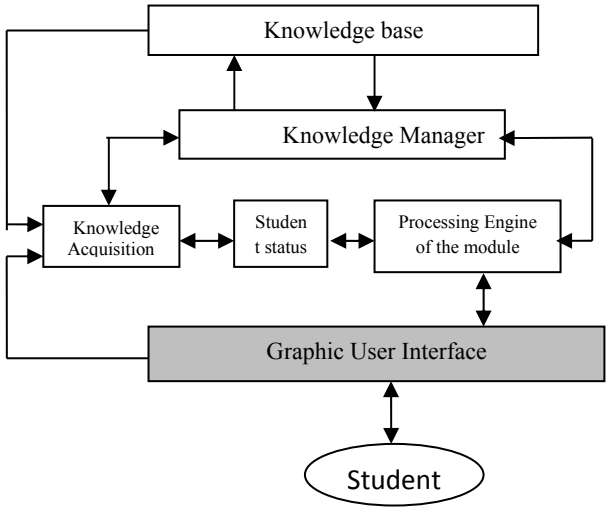

Fig. (5) Structure of an agent in the AG_TUTOR

As shown in Fig. (5), this pattern consists of:

- **Knowledge Base:** includes domain knowledge, pedagogical knowledge and deals with the knowledge manager that manages the knowledge base to realize the job of the agent.
- **Graphic User Interface:** perceiving the dynamic learning environment and acting upon it. It deals with the student until his status is defined
- **Student Status:** a general abstraction for many kinds of student states a pedagogical agent can be made. It can also contain current values of parameters of the agent's relationships with other agents.

- **Knowledge Acquisition**: reflects the learning capability of the agent to update and modify its knowledge over time.
- **Processing Engine of the module:** is responsible for analysis of the agent's current internal state and possible modifications of that state.
- **Knowledge Manager:** that manages the domain knowledge and pedagogic knowledge and deals with the processing engine of the module to process the module. Also, it deals with the knowledge acquisition to modify the student status and raise his performance.

7. Conclusions:

As agents gain a wider acceptance and become more sophisticated, they will become a major factor in the future of many domains. This paper demonstrates an intelligent tutoring system that built with help of group of intelligent agents. An intelligent agent is based on goals and plans, because such a system can more easily detect positive and negative goal interactions. Within this planning paradigm, the key problem for building an intelligent agent system is how to detect the right goals for the agent in appropriate situations. Once an agent has adopted appropriate goals, planning to satisfy those goals is a better-understood problem. In this research, an educational system is implemented. This system uses a multi intelligent agent that helps to execute the educational process. The system consists of a pedagogical module, a question selector module, a student module and interface module in addition to a multi intelligent system that connect all of these modules together. These agents are: Teaching Assistance Agents, Constraints and Hints Agent, Learning Strategy Environment Agent, Secretary Agent, Interface Agent. This educational system is dealing with the student in an intelligent interface that can diagnose his problem and give him the solutions to this problem. The interface that knows more than its user needs to be an intelligent agent in order to respond properly to its user.

References

[1] A. Jafari, "Conceptualizing Intelligent Agents For Teaching and Learning", Educause Quarterly number 3, pp. 28-34, 2002.

[2] M. Al Emran, *K. Shaalan*, "A Survey of Intelligent Language Tutoring Systems", International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi-India, pp. 393-399, 2014.

[3] https://en.wikipedia.org/wiki/Software_agent

[4]https://en.wikipedia.org/wiki/Intelligent_agent

[5] V. M. Ramesh, N. J. Rao, "Tutoring and Expert Modules of Intelligent Tutoring Systems", A Technical Report presented to the International Institute of Information Technology, Electronics City, Bangalore, 2010.

[6] S. Ramachandran, E. Remolina, D. Fu, "FlexiTrainer: A Visual Authoring Framework for Case-Based Intelligent Tutoring Systems", Proceeding of ITS 2004: Intelligent Tutoring Systems conference, pp 848-850, 2004.

[7] C. S. Blanc, "High-level Petri nets with data structures", In European Workshop on Application and Theory of Petri nets, pp. 141–170, Helsinki, Finland, June 1985.

[8] E. de B. Costa, M.A. Lopes, and E. Ferneda, " MATHEMA: A learning environment based on a multi-agent architecture", In LNAI, volume 991, pages 141–150, October 1995.

[9] N. R. Jennings and M. Wooldridge, "Applications of Intelligent Agents", Queen Mary & Westfield College University of London, Agent Technology © Springer-Verlag Berlin Heidelberg. 1998

[10] D. Chin, "The UNIX Consultant as an Intelligent Agent**",** Newsletter, ACM SIGCHI Bulletin, Volume 20 Issue 1, page 77, ACM New York, USA, July 1988.

[11] S. Chakraborty, D. Roy, A. Basu, "Development of Knowledge Based Intelligent Tutoring System", Chapter 5, TMRF e-Book Advanced Knowledge Based Systems: Model, Applications & Research (Eds. Sajja & Akerkar), Vol. 1, pp 74 – 100, 2010.

[12] P. Brusilovsky, D. W. Cooper, "ADAPTS: Adaptive Hypermedia for a Web-based Performance Support System", The Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Pittsburgh, USA, 1998.

[13] F.H. Roth, D. A. Waterman, D. A, Lenat,*"Building Expert Systems*", Addison-Wesley Publishing, ISBN 0-201-10686-8, 1983.

[14] Y. Akbulut, C. S. Cardak, "Adaptive Educational Hypermedia Accommodating Learning Styles: A Content Analysis of Publications from 2000 to 2011", Computers and Education 58, Elsevier, pp. 835-842, 2012.

[15]https://en.wikipedia.org/wiki/Intelligent_tutoring_system

[16] C. Mills, B. Dalgarno, "A Conceptual Model for Game Based Intelligent Tutoring Systems", The Proceedings of Ascilite Singapore: Mills and Dalgarno, pp.692-702, 2007.

[17] A. Mitrovic, S. Ohlsson, "Evaluation of a Constraint-Based Tutor for a Database Language", International Journal of Artificial Intelligence in Education, pp. 238-256, 1999.

[18] N. Khodeir, "Student Modeling for Arabic Language Skills in Intelligent Language Tutoring Systems", International Journal of Computer Science and Information Security (IJCSIS), Vol. 13, No. 11, November 2015.

[19] D. N. CHIN, "Intelligent Interface as Agents", CHAPTER9 in "Intelligent User Interfaces, Addison-Wesley Reading, MA, pp. 127–206, 1991.

[20] O. P. Rishi, R. Govil, M. Sinha, "Distributed Case Based Reasoning for Intelligent Tutoring System: An Agent Based Student Modeling Paradigm", Engineering and Technology International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering Vol:1, No:5, pp. 111-114, 2007.