International Journal of Engineering Education Vol. 27, No. 2, pp. 248–256, 2011
Printed in Great Britain
0949-149X/91 $3.00+0.00

Intelligent Multiagent Tutoring System in Artificial Intelligence*

VICTOR R. L. SHEN
Department of Computer Science and Information Engineering, College of Electrical Engineering and Computer Science, National Taipei University, 151, University Rd., Sanhsia, New Taipei City, 237 Taiwan. E-mail: rlshen@mail.ntpu.edu.tw

CHENG-YING YANG[1]
Department of Computer Science, Taipei Municipal University of Education, 1, Ai-Kao W. Rd, Taipei 100, Taiwan.
E-mail: cyang@tmue.edu.tw

An Intelligent Multiagent Tutoring System (IMATS) in Artificial Intelligence (AI) is presented in this paper. The authors describe ways of improving the quality of teaching AI through the use of agent technology. An interface agent, an authorization agent, an SQL agent, and a student agent can be integrated into the intelligent learning environment. These agents can guide and assist students as needed, probe their understanding, and promote learning and retention. They also exploit the natural human tendency to respond socially to computing systems. Here a brief historical perspective of intelligent learning environments is discussed and followed by an explanation of the IMATS framework. Next, a detailed description of the intelligent tutoring system, IMATS, is given. Then the results of usability evaluation, effectiveness evaluation, and satisfaction level evaluation of the software system are presented. The IMATS framework incorporates an incremental approach to obtain the dynamics of knowledge accumulation in the domain of interest and the learned knowledge content over time.

Keywords: agent technology; artificial intelligence; intelligent tutoring system; web-based instruction

1. Introduction

There has been a fast rise in interest in the Internet and the World Wide Web as vehicles for instructional delivery. Although the web-based course management tools have appeared and are widely used, the on-line courses are often patterned after classroom instruction and thus suffer from the limitations of large group instruction [1].

Computer-aided instruction (CAI) systems were developed in the early 1960s, when they were used to schedule resources, handle teaching aids, and grade tests. However, the computer was mainly applied to interact directly with the student rather than used to serve as an assistant to the human instructor. CAI systems thus advanced to intelligent computer-aided instruction (ICAI) systems and then intelligent tutoring systems (ITSs) when the principles of artificial intelligence were employed in them [2]. The 'intelligence' in these ITSs is that these systems can adapt themselves to the learning features of the students, namely, the speed of learning something, particular in an area in which the student excels or falls behind, and the rate of learning as more knowledge is acquired. In such intelligent learning environments, a set of agents can be co-opted to finish pedagogical tasks [3–5].

The above reasons have motivated us to develop a new educational program called Multimedia and Web-based Technology of the Computer Science and Information Engineering, National Formosa University, Taiwan. In the academic year of 2007, this program aimed to educate students to be engineers who might design and implement intelligent systems for multimedia information and knowledge processing and who could realize and apply fully interactive man–machine interfaces.

As part of this program, a two-semester undergraduate course on AI was developed with following objectives:

1. to introduce the basic ideas of knowledge-based systems and the relevant AI techniques regarding search and problem solving algorithms, reasoning and production systems, expert systems, knowledge representation, fuzzy logic, natural language understanding, machine learning, speech recognition, neural networks, and agent technology;
2. to explain by examples and instruct the issues regarding AI programming in general and intelligent multi-agent applications in particular.

The IMATS framework is intended to create a learning experience that is a significant improvement on the traditional classroom learning experience and gets closer to the ideal state of individualized instruction, since the advantages of the Web can assist students in learning anywhere

[1] Corresponding author.

* Accepted 1 February 2011.

and anytime. In other words, the instruction should be student-centered, and thus encourage student motivation and foster active cognitive processing during learning [6–10]. Therefore, it can provide students with individualized feedback in order to accomplish the learning objectives completely.

The rest of this paper consists of the following topics. Section 2 discusses the intelligent learning environments. The IMATS framework is introduced in Section 3. A comparison between a control group and an experimental group is given in Section 4. Section 5 gives the conclusion.

2. Intelligent learning environments

The AI course focuses on letting students learn how to apply the basic AI algorithms to solve real-world problems, instead of learning how to use complicated software tools. Consequently, the requirements that an intelligent tutoring system (ITS) for the AI course should fulfill can be summarized as follows [11]:

1. The ITS should provide the development of intelligent agents aiming to monitor, filter, and retrieve relevant information from the World Wide Web.
2. The ITS should be simple and user friendly, and include built-in, Java-implemented, agent templates that employ example-based learning, and can be edited to contain the required AI algorithms based on the goals of a programming assignment.
3. It should also include the concepts of concurrency, multi-agency, and persistency.
4. It should meet SCORM (Sharable Content Object Reference Model) specifications, which are considered as international standards in education.

Nowadays, due to the relatively low cost of modern technology, computer hardware and its peripheral devices become a less and less impediment to the development of an intelligent tutoring system. The largest obstacle is one's ability to develop effective and interactive learning environments [2]. Intelligent tutoring systems must provide their own learning environment and let students participate in it. Most intelligent tutoring systems contain their instructional model represented by procedural rules [12]. These rules may point out certain student errors and provide correct solutions. Other informative factors include the history of the tutorial session, significant student actions, the type of knowledge taught, and comparisons between the student's and the tutor's knowledge [2].

The agent-based learning method, coming directly from a learner to respond intelligently, was developed a long time ago [13]. The idea of intelligent agents has been broadly investigated for several years. In essence, an agent can be thought of as an entity that perceives its environment by using sensors and performs actions by using effectors [14]. From the system's points of view, an agent possesses a specific plan of action restricted by a certain domain and a behavior pattern that interacts with the world stimuli from the environment. Based on AI literature, agents have such properties as being autonomous, reactive, temporal, goal driven, communicative, flexible, and can learn [3].

Compared with a human-based approach, the agent-based learning environment has several benefits. The student has flexibility in terms of time needed, convenience, adjusting interaction preferences, encouragement to reflect on the thinking process, and the presence of a willing collaborator [3]. In some online learning environments, one agent alone performs the entire set of functions, namely, managing the lesson plan, interacting with the student, organizing the AI content, taking feedback from the student, periodically evaluating the student performance, and updating the AI content [15]. However, other online learning environments contain a set of coordinating agents that work together to manage an environment that is conducive to learning [16].

Although numerous agent frameworks have been proposed in the huge body of literature [17], and several software packages based on agents are now available [18], few of these existing agent frameworks satisfy the above requirements. The existing Java-based agent frameworks contain Agent Factory, IBM Aglets, AMETAS, Beegent, Cougaar, CIAgent, DECAF, FIPA-OS, Grasshopper, Hive, JACK, JADE, JAFMAS/ Jive, Kaariboga, LIME, MadKit, NOMADS, OpenCybele, Pathwalker, SeMoA, Tagent, Tryllian, Voyager, SAF, and ZEUS [11]. Consequently, the above reasons lead the author to create the IMATS framework that would fulfill most of the previous requirements and yield an educational tool used to introduce an AI course.

3. Intelligent multiagent tutoring system (IMATS) framework

The Intelligent Multiagent Tutoring System (IMATS) focuses on the instruction of undergraduate computer science students. The materials covered are those typically presented in the Artificial Intelligence (AI) course. IMATS monitors the student's activities in learning and understanding the main material, pointing out any difficulties that the student might have and tutoring the student when necessary. The tutoring includes identifying a student's weak points in one or more learning objec-

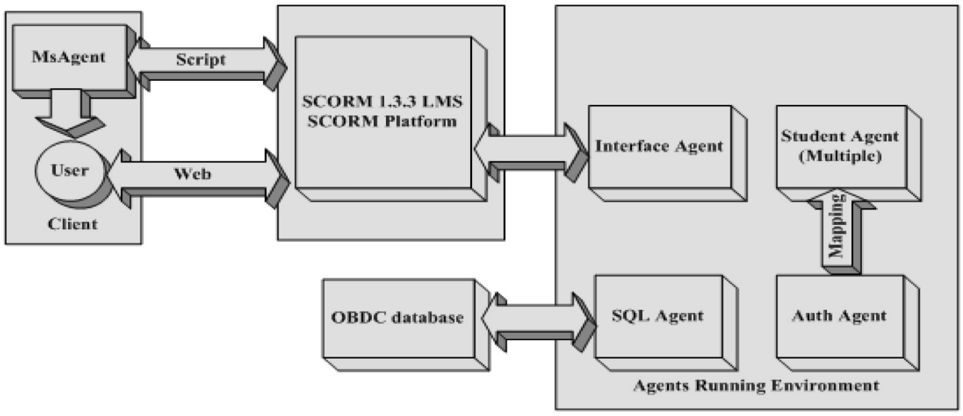

Fig. 1. Intelligent Multiagent Tutoring System (IMATS) framework.

tives. Based on the nature of the weak points, IMATS then modifies the instructional procedures and/or the amount of materials provided. These modifications are made to let the student correct any logical mistakes and explain any misunderstanding regarding the material.

In order to construct the IMATS framework, we applied the following software tools: JADE, JAVA execution environment, Tomcat JSP server, Apache web server, Microsoft agent, Reload editor for SCORM, MASH, JBuilder, Visio, ODBS database, and Macromedia Flash MX 2004.

The IMATS framework courseware contains the following topics [19–22]:

1. AI Introduction
2. Search and Problem Solving
3. Reasoning and Production System
4. Expert Systems and Knowledge Engineering
5. Uncertainty, Probabilistic Reasoning and Fuzzy Logic
6. AI Programming Languages
7. Knowledge Representation
8. Natural Language Understanding
9. Cognition
10. Machine Learning
11. Robotics
12. Speech Recognition
13. Neural Networks
14. Interface and Information Agents
15. Multiagent

3.1 IMATS framework

The proposed Intelligent Multiagent Tutoring System (IMATS) consists of a network of agents that work cooperatively to deliver lessons effectively to students (Fig. 1). In this system there are four main agents, namely, interface agent, authorization agent, SQL agent, and student agent. Compared with 1-tier and 2-tier (client–server) architectures, this 3-tier architecture has the following advantages:

1. The user does not need to install a client software package, but only one general browser is needed. Hence this system is more user-friendly than the traditional one.
2. When a new version of the software package is developed, each client software component needs to be upgraded, but this 3-tier architecture only needs to upgrade a JSP web server. Thus this will keep the cost down a lot and become much easier to maintain.
3. When the number of users is increased, the workload of the client–server must become heavier. However, this system can allot more tasks to different computers. For example, the JSP web server can be one computer, the agent's running environment can be more than one computer, and the ODBC database can also be another computer. Therefore, this system is superior to the traditional one in terms of adaptability and load balancing.

This IMATS framework consists of three subsystems, which are Web server/SCORM 1.3.3 LMS, ODBC database, and agents running the environment. First, the Web server/SCORM 1.3.3 LMS provides the user with web services. We can key in and modify its commands in order to link to the interface agent and to perform the data analysis and decision making. Second, the ODBC database is used to store the user's personal and learning information. No specific database server is needed and thus it is more flexible to manage. Third, the agent's running environment is used to provide agents with scheduling, message communication, and life-cycle management. In this IMATS framework, we define four agents as follows:

1. *Interface Agent*: Based on FIPA specifications, each agent must have its own specific message-

exchange formats. Since SCORM LMS cannot directly exchange message with the agent's running environment, the interface agent is thus needed to communicate with SCORM LMS by translating messages into TCP/IP format.

2. *Authorization Agent*: The authorization agent is used to perform two main tasks. First, it must verify the user's identification and authorization. Second, it must construct a student agent corresponding to the user. Meanwhile, it must appropriately send a query to the SQL agent.
3. *Student Agent*: Since this is a multi-user system, one authorized user must correspond to one's student agent developed in the agent's running environment.
4. *SQL Agent*: Similar to the design concept of interface agent, SQL agent serves as a communicating channel between the OBDC database and the agent's running environment. To extend the scalability of the modules, access to the OBDC database should be in a higher frequency.

3.2 System operation description

As depicted in Fig. 2, the system operations sequence is described as follows:

1. The user starts to log in the IMATS by using ID and password.
2. SCORM 1.3.3 LMS accepts the request from the user and sends message to the interface agent.
3. The interface agent verifies the log-in message and sends the related message to the authorization agent.
4. The authorization agent, ready to authorize and match the user's ID and personal data, carries out communications with the SQL agent.
5. The SQL agent sends a query message to the OBDC database.
6. The OBDC database replies to the SQL agent.
7. The SQL agent sends the queried results back to the authorization agent.
8. The authorization agent authorizes the queried results and ensures that they are correct, and one student agent is assigned accordingly.
9. The authorization agent sends a success message back to the interface agent.
10. The interface agent sends a success and the related message to SCORM 1.3.3 LMS.
11. SCORM 1.3.3 LMS presents the correctly queried results to MsAgent and the user.

The IMATS message exchange sequence and message communication between agents are shown in Fig. 3.

3.3 Sample screen shot in JADE

Once the GUI (graphical user interface) parameters are input to the management tool, the JADE is activated. The sample screen shot of a sniffer agent in JADE is illustrated in Fig. 4. As the name implies, the sniffer agent is basically a FIPA (Foundation for Intelligent Physical Agents)-compliant agent with sniffing features. When the user is determined to sniff an agent or a group of agents, every message directed to/ from that agent/ agent group is tracked and displayed in the sniffer agent's GUI. The user can view every message and save it to the disk. The user can also save all the tracked messages and reload it from a single file for later analysis.

3.4 Audio-visual teaching materials in SCORM

First, we can record all the teaching activities and materials using an Audio-Visual (A-V) recorder. Then we put the A-V files into SCORM. Students can access these A-V files any time and anywhere. The Audio-Visual teaching materials in SCORM

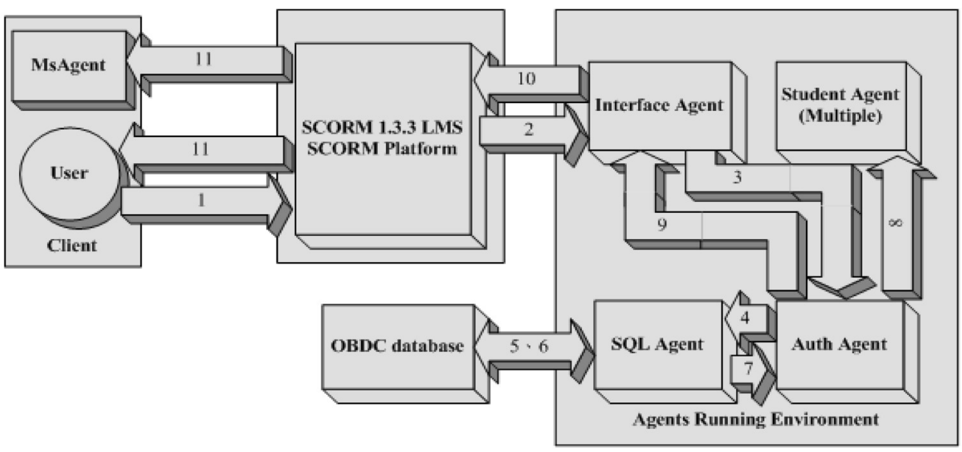

Fig. 2. Operations sequence diagram.

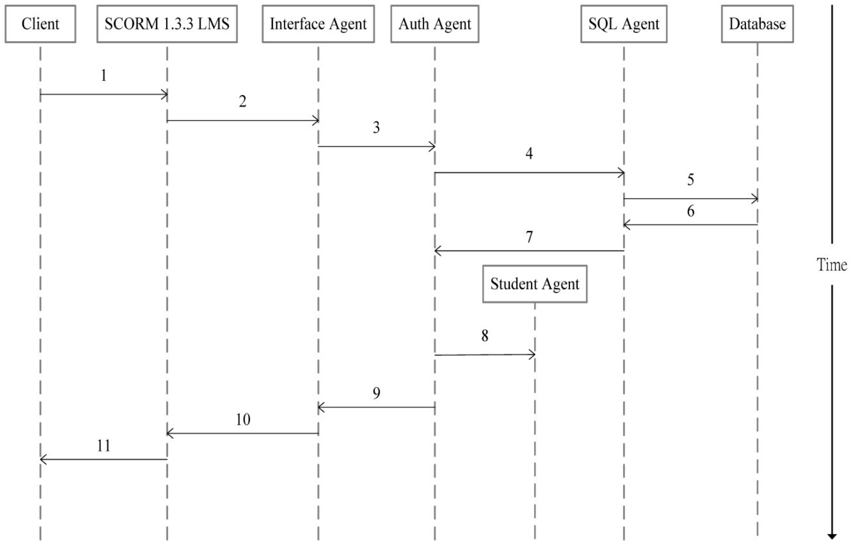

Fig. 3. Message exchange sequence.

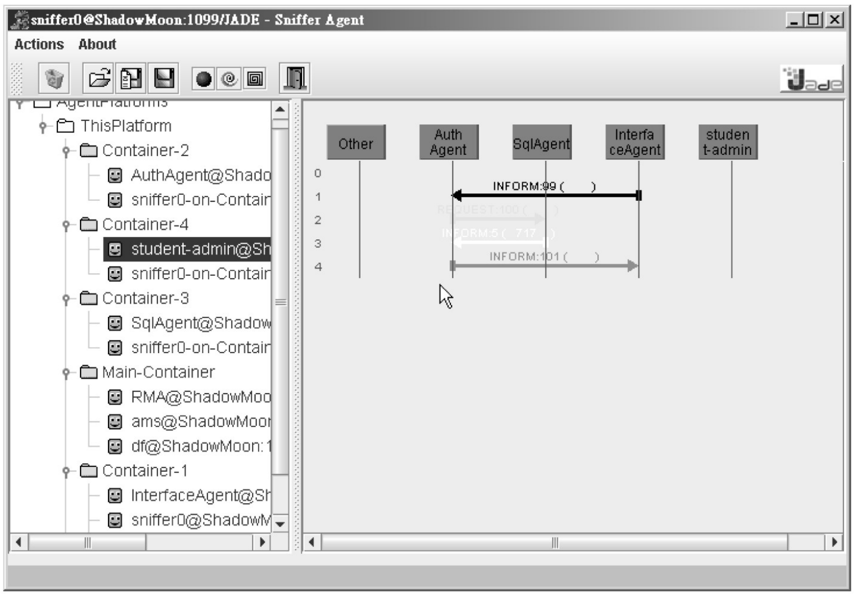

Fig. 4. Sample screen shot of JADE.

can provide students with the opportunity to view and review the course contents if necessary.

3.5 Creative methods

Based on [23], the promising creative methods include the following possible activities: to reverse, to transfer, to combine, to change direction, to extend, and to reduce.

1. To *reverse* is to subtract on one side and to add on the other side.
2. To *transfer* is to add or to subtract according to, or similar to, a pattern that already exists somewhere else.
3. To *combine* is to add, and is probably the second most widely used method to solve problems and to create new things.
4. To *change direction* has a special case, *to reverse*. For instance, *to reverse* is a change direction of 180 degrees. But, the change of direction can be any number of degrees.
5. To *extend* can actually be viewed as to *combine* something similar to something already in existence. So to *extend* can be considered as a special case of combining.
6. To *reduce* is to subtract. To divide is a special case of *reducing*. *To reduce* is the *reverse* of *extending*, and is also the *reverse* of *combining*.

3.6 Macromedia flash

All the typical problems in each chapter are solved and demonstrated by macromedia Flash. Students can obviously get some key ideas from the macromedia Flash, instead of just reading verbal and symbolic algorithms. Also, they can create new problem-solving abilities.

Example: Missionaries and Cannibals [19]
Three missionaries and three cannibals are on one side of a river, with a canoe. They all want to get to the other side of the river. The canoe can only hold one or two people at a time. At no time should there be more cannibals than missionaries on either side of the river, as this would probably result in the missionaries being eaten.

As depicted in Fig. 5, the taller and green clothed people are missionaries and the shorter and red clothed people are cannibals. After students have learned this example, they can easily get a clear picture of how to solve this problem. Moreover, students are asked to transfer this problem solving ability to another, for example, a farmer crossing the river with three animals including a cat, a rat, and a dog. Also, students are asked to extend the number of missionaries and cannibals to four, or five.

4. Evaluation of IMATS

The format of this IMATS evaluation is cited from the [2]. The evaluation of the proposed IMATS framework contains two aspects: a usability evaluation and an effectiveness evaluation. Usability information focuses on the IMATS framework's functional effectiveness, efficiency, ease of learning and use, quality assurance and motivational influence. Effectiveness evaluation provides information about the impact on students' learning.

4.1 Usability evaluation

1. *Sample and Instrumentation*: Usability data were obtained from students taking the AI course who were participating in the effectiveness study. In total, 56 students successfully completed the IMATS Usability Questionnaire, which included 20 items using a five-point Likert-type scale (where 1 = strongly disagree and 5 = strongly agree) and open-ended items. Usability data were also obtained from the user's log files.
2. *Usability Results*: Students rated IMATS as a useful learning tool for the acquisition of AI related concepts (mean = 3.62) and an essential source of information (mean = 3.41). About 56% of students thought that the more they used the IMATS framework, the more they learned. Although students do not agree that they enjoy a learning tool as much as they do a leisure one, the evaluator directly asked them this question (mean = 3.09). Most students found IMATS framework easy to use (mean = 4.02) and intuitive (mean = 3.72). They gave the same high marks to use of the different office components (mean = 3.81) and to use of the virtual lab (mean = 3.21). They reported limited navigation within sections (mean = 3.23).
3. Students reported that the various components of the IMATS framework were helpful; they

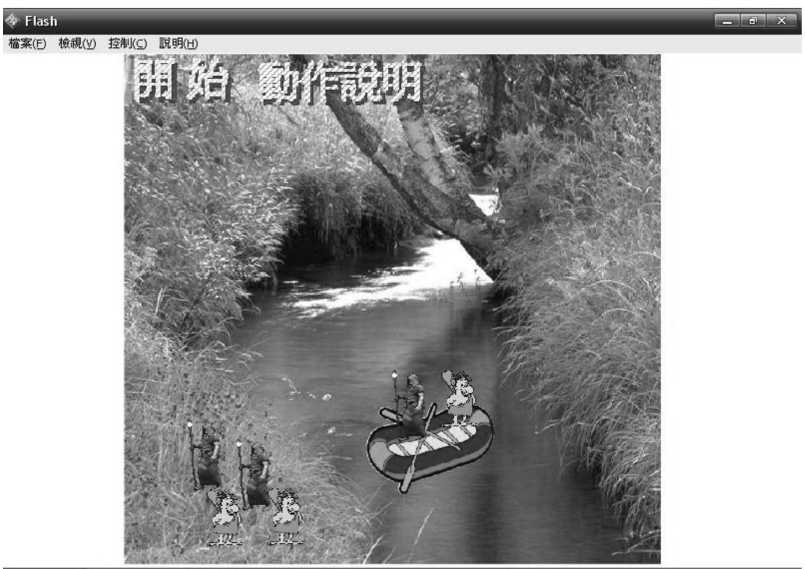

Fig. 5. Sample screen shot of the Example. (Chinese translation. Start and action descriptions; F. File; V. View; C. Control; H. Help).

rated the virtual bookshelf as the most helpful feature (62%). Eighty-two percent highly endorsed the clarity of the teaching materials. Although participating students might be technologically sophisticated, they rated numerous features of the virtual environment as realistic: student's role as an AI engineer (mean = 3.12), the company office (mean = 3.11), and virtual lab (mean = 3.07). Students rated IMATS as a high-quality multimedia product including audio quality (mean = 4.01) and graphics quality (mean = 3.97).

The analysis based on the students' log files indicated that students accomplished a number of learning objectives (median = 14). The most frequently encountered objectives were related to the heuristic search algorithm, reasoning, fuzzy logic, and multi-agent techniques. The least frequently encountered were remedial objectives regarding neural networks, robotics, and speech recognition. This outcome is fully consistent with the content expected in the AI course.

4.2 Effectiveness evaluation

1. *Sample and Design*: Students enrolled in the AI course participated in a quasi-experimental study to examine the impact of IMATS on learning (N = 56). The AI course contained two sections: 28 students in one section comprised the control group and 28 students in the other section obtained a curriculum in which the instructor applied IMATS. The intervention was assigned to match sections randomly.
2. *Data Analyses and Results*: In order to compare the outcome of traditional instruction with that of the proposed IMATS framework, we select one junior class of 56 students and divide it into two groups, namely, a control group (A) and an experimental group (B) [1, 5, 10]. Each group contains 28 students.
3. The control group (A) was taught by traditional instruction method and the experimental group (B) was taught by the proposed IMATS framework. Starting from the second week of this semester, we are intensively teaching Artificial Intelligence in the four contiguous weeks using the two different instructional methods. The experimental results are summarized below.

4.2.1 The statistical scores analysis for Group A and Group B

Based on the tests, the related scores are shown in Table 1. The number of attendees is 56 (N) in total. For group A test, the highest score is 80, the lowest score 17, and the average 51.22. For group B test, the highest score is 95 and the lowest score 46, the average 62.89.

4.2.2 t-test for Group A and Group B scores

For research purposes, the related scores are analyzed accordingly. Group A and Group B scores are t-tested as shown in Table 2. As the t-tested results, the scores' standard deviation is up to the difference of significance. This shows that the students' academic achievements have been significantly improved if the IMATS teaching activities are used. In other words, the IMATS activities can remarkably enhance the learning performance.

In summary, for the control group, the students felt that the teaching materials were tedious and boring; learning speed is slow, and retention is ambiguous. For the experimental group, the students felt that the teaching materials were interesting and attractive, learning speed is fast, and retention is obvious.

4.3 Satisfaction level evaluation

The results for the satisfaction level from the questionnaire are presented below.

1. The IMATS sessions have improved my under-

Table 1. Group A and Group B tests' statistic scores

	No. of students	Lowest	Highest	Average	Standard deviation
Group A	28	17	80	51.22	14.21
Group B	28	46	95	62.89	15.12
N	56				

Table 2. t-test summary for Group A and Group B scores

	No. of students	Average	Standard deviation	t-values	Significance
Group A	28	51.22	14.21	2.729	0.025 *
Group B	28	62.89	15.12		

* $p < 0.05$.

standing of the lectures provided within this module. (85%)
2. The IMATS sessions have helped my understanding of the theoretical AI algorithm design process. (80%)
3. The IMATS sessions have improved my understanding of the practical aspects of AI algorithm design. (73%)
4. Having participated in the IMATS sessions, my confidence and ability to undertake a real AI algorithm design has been enhanced. (75%)
5. The IMATS sessions were realistic and reflected typical real practical situations. (69%)
6. The IMATS sessions have helped my ability to work in groups. (75%)
7. The IMATS sessions were well organized and effective. (78%)
8. The IMATS sessions should be kept as part of this module. (85%)
9. How motivating do you feel the IMATS system was? (Very motivating, 83%)
10. How much harder did you work than usual? (Much harder, 72%)
11. How well did you feel learned the course matter? (Very well, 87%)
12. How was your learning experience? (Fun, 86%)

According to the above results, if the satisfaction level 75% is set to be a successful criterion, then those items except (3), (5), (10) have achieved the successful criteria.

4.4 Discussion

The IMATS framework presents the following benefits and results:

1. providing the Department of Computer Science and Information Engineering, National Taipei University, Taiwan, with an example software system on Artificial Intelligence;
2. providing teachers with a computer platform to setup, save, analyze, deduce, and organize the teaching materials and files;
3. assisting students in visualizing the behavioral performance of those complex AI algorithms;
4. enhancing students' learning interests and efficiency;
5. accumulating teachers' annual teaching experiences in order to improve the quality of AI-based instruction.

5. Conclusions

The IMATS framework aims to assist students in clearly understanding the theory and applications of Artificial Intelligence. It obviously performs better and becomes a successful undertaking. Furthermore, by using the IMATS one the students' problem solving capabilities and creativity still need to be cultivated. Finally, the student must be assigned some AI-related projects and do more exercises in order to design some creative algorithms.

Applying the IMATS framework, one can control the student's progress non-intrusively and efficiently modify or tailor lesson delivery to help the student with learning. It also allows for incremental knowledge updates, thus providing the current knowledge bases.

Acknowledgments—The authors are very grateful to the 56 undergraduate students who took the AI course and provided the authors with numerous valuable comments on how to enhance the preliminary version of the IMATS. Also, this work was supported by the National Science Council of Taiwan, R.O.C., under grant NSC 99- 2221- E- 305- 013.

References

1. W. L. Johnson, Using agent technology to improve the quality of web-based education, *Proceedings of the International Conference on Artificial Intelligence in Education,* June 2002, pp. 1–26.
2. B. P. Butz, M. Duarte, and S. M. Miller, An intelligent tutoring system for circuits analysis, *IEEE Trans. on Educ.*, **49**(2), 2006, pp. 216–223.
3. S. Piramuthu, Knowledge-based web-enabled agents and intelligent tutoring systems, *IEEE Trans. on Educ.*, **48**(4), 2005, pp. 750–756.
4. J. Bergasa-Suso, D. A. Sanders, and G. E. Tewkesbury, Intelligent browser-based systems to assist internet users, *IEEE Trans. on Educ.*, **48**(4), 2005, pp. 580–585.
5. A. C. Graesser, P. Chipman, B. C. Haynes and A. Olney, AutoTutor: an intelligent tutoring system with mixed-initiative dialogue, *IEEE Trans. on Educ.*, **48**(4), 2005, pp. 612–618.
6. G. P. Baxter and R. J. Shavelson, Science performance assessments: benchmarks and surrogates, *International Journal of Educational Research*, **21**(3), 1994, pp. 279–316.
7. V.M. Roth and A. Roychoudhury, The development of science process skills in authentic contexts, *Journal of Research in Science Teaching*, **30**(2), 1993, pp. 127–156.
8. A. Dasgupta, R. A. Matthes and C. G. Takoudis, Web-based instruction on the fundamentals and design of micro- and nanoelectronic processes: innovations, challenges and benefits, *Int. J. Eng. Educ.,* **18**(5), 2002, pp. 539–549.
9. T. Chang, P. Jaroonsiriphan and X. Sun, Integrating nanotechnology into undergraduate experience: a web-based approach, *Int. J. Eng. Educ.,* **18**(5), 2002, pp. 557–565.
10. R. Li, *A Theory of Conceptual Intelligence: Thinking, Learn, Creativity, and Giftedness*, Praeger, Westport, CT, 1996.
11. M. Pantic, R. Zwitserloot, and R. J. Grootjans, Teaching introductory artificial intelligence using a simple agent framework, *IEEE Trans. on Educ.*, **48**(3), Aug. 2005, pp. 382–390.
12. J. W. Rickel, Intelligent computer-aided instruction: A survey organized around system components, *IEEE Trans. Syst., Man, Cybern.*, **19**(1), 1989, pp. 40–57.
13. J. Weizenbaum, ELIZA- a computer program for the study of natural language communication between man and machine, *Commun. ACM*, **9**(1), 1965, pp. 36–45.
14. M. Wooldridge and N. R. Jennings, Eds., *Intelligent Agents*, Sringer-Verlag, Berlin, Germany, 1995.
15. A. Jafari, Conceptualizing intelligent agents for teaching and learning, *Educause Quart.*, **3**, 2002, pp. 28–34.
16. P. A. Jaques, J. L. Jung, A. F. de Andrade, R. H. Bordini, and R. M. Vicari, Using pedagogical agents to support collaborative distance learning, *Proc. Computer Supported Collaborative Learning (CSCL)*, G. Stahl, (Ed.), 2002, pp. 546–547.

17. AgentLink. Agents in the Press-European co-ordination action for agent-based computing [Online]. Available: http://www.agentlink.org/press , 2007.
18. AgentLink. Agent Software-European co-ordination action for agent-based computing. [Online]. Available: http://www.agentlink.org/resources/agent-software.php/ , 2008.
19. B. Coppin, *Artificial Intelligence Illuminated*, Jones and Bartlett Publishers, Inc., 2004.
20. G. F. Luger and W. A. Stubblefield, *Artificial Intelligence*, 3nd edn, Addison-Wesley, 2000.
21. S. Russell and P. Norvig, *Artificial Intelligence— A Modern Approach*, Prentice-Hall, 2003.
22. K. W. Tracy and P. Bouthoorn, *Object-Oriented Artificial Intelligence Using C++*, Computer Science Press, 1997.
23. M. H. Lee, *Becoming More Creative- How to Invent and Innovate*, Unalis Corp. September 1995.

Victor R. L. Shen, received his B.S. and M.S. degrees in Electronic Engineering of Industrial Education from the National Taiwan Normal University, Taiwan, in 1975 and 1980, respectively. Also, he earned his M.S. degree in Electrical Engineering and Computer Science from the University of Illinois at Chicago, USA, in 1987. And he received his Ph.D. degree in Electrical Engineering from National Taiwan University in 1997. From July 1980 he has worked as a lecturer, professor and Chairman in Taiwan universities. From August 2007 to July 2009, he was Chairman at the Graduate Institute of Electrical Engineering in National Taipei University. From August 2009 to the present, he has been Dean of the College of Electrical Engineering and Computer Science in the National Taipei University. His current research interests include applications of Petri nets and artificial intelligence (AI) techniques to knowledge-based systems, specification and formal verification of digital systems, high-level synthesis of VLSI designs, information security, and e-Learning. Dr. Shen is a Fellow of IET, a senior member of IEEE, and member of ACM, IEICE, Taiwanese Association for Artificial Intelligence (TAAI), Taiwanese Association for Consumer Electronics (TACE), the Institute of Information and Computing Machinery (IICM), the Chinese Institute of Engineering, and the Chinese Institute of Industrial Education. He received the distinguished guidance awards in computer programming contests held by the Ministry of Education in Taiwan six times in 1985, 1990, 1991, 1992, 1995, and 1996. He was a winner of the 1998 Outstanding Research Award of the National Science Council in Taiwan. He was also included in Marquis Who's Who in the World, New Providence, NJ, U.S.A., in 2006.

Cheng-Ying Yang received his M.S. degree in Electronic Engineering from Monmouth University, New Jersey, in 1991, and his Ph.D. from The University of Toledo, Ohio, in 1999. He is a member of the IEEE Satellite & Space Communication Society. Currently, he is employed as an Associate Professor at Taipei Municipal University of Education, Taiwan. His research interests are performance analysis of digital communication systems, signal processing, computer security, and e-Learning systems.