

Improving Learning Skills based on Artificial Intelligent Technique

Abdullah S. Alotaibi

Faculty of Science and Humanities Shaqra University, KSA

Abstract

Artificial Intelligence (AI) and its applications in the area of learning have played an important role to impart intelligence in learning tools and techniques. In recent years, many of teachers can cooperate together to put the curriculum of one/more computer programming language(s). The students can access the system through WWW, select any language they want to learn as well as the style of presentation they prefer and they can exchange their experiences. A personal assistant agent for teachers, a personal assistant agent for students with an adaptive interface, and tutoring agent has been built. The tutoring agent resides on the server side and communicates via HTTP and IIOP. This structure allows customization of the personal assistant agent for teachers and personal assistant agent for students to the needs of the teachers and students, without putting extra burden on the server. In addition, this allows having many teacher agents attending to the needs of a single or multiple student agent(s).

Keywords: Intelligent agent, Expert system, and computer aided instruction

1. INTRODUCTION

In recent years, many of the universities and educational institutions worldwide offer online services such as for admissions, virtual (online) learning environments in order to facilitate the lifelong learning and to make this compatible with other educational management activities [16]. For example, a teacher may create a purely Web-based delivery system including online handouts in respect of student's activities, projects and lists of resources for reference. The students and other learners may access web-based material anytime from anywhere in the world, being connected through Internet. The idea behind of using Artificial Intelligence (AI) in e-learning skills is to facilitate quality education and training in a computer based system worldwide. To facilitate quality education, the identification & selection of various factors that may influence a student's academic performance is very important. Knowing these factors is important for parents & teachers working positively on these factors may improve the performance of the student. AI that may result in development of e-learning systems that may extensively benefit in providing facilities more personalised & better decision support system (DSS). By offering such approach, education system could play much better for student centric operation towards positive improvement of his performance. Cormo and Snow [11] define adaptive instruction as instructional approaches and techniques that are geared to meet the needs of individually different students in developing knowledge and skills required to learn a task. Similarly, Park and Lee

[13] state that "adaptive instruction embodies all instructional forms that accommodate the needs and abilities of different learners". In this paper, we extend the features of that system to include Artificial Intelligence, Internet-based, adaptive learning environment and multi-agent technologies. With the WWW as an educational platform, it will be feasible for the students to access the multimedia courseware with general-purpose browsers. No special tools are required to start learning. For the courseware provider, it is not necessary to worry about the distribution and maintenance of the copies of the courseware but they just take care of the original on their server [1, 2]. The proposed system consists of three agents representing a server-client relationship, tutoring agent as a "server", personal assistant agent for teachers, and personal assistant agent for students "clients". The personal assistant agent for students can communicate with the tutoring agent through the WWW to retrieve the tutoring dialog of the command(s) that a student wants to practice, and to access the experiences of other students in the blackboard module. While the personal assistant agent for teachers communicates with the tutoring agent to add/modify semantic rules of computer programming languages and to check the correctness of the contents of the blackboard database see figure 1.

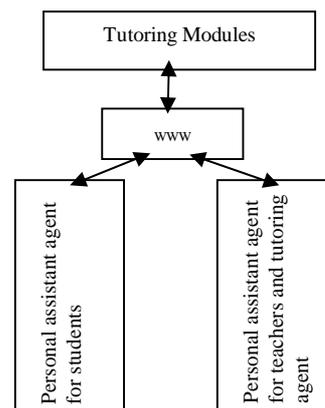


Fig. 1. Expert system

The remainder of the paper is organized as follows: Section 2. Assistant Agent. Section 3, focuses on State Adaptation. Section 4, presents a Tutoring Agent. Section 5, show the Expert System and WWW interface. Finally, section 6, show the concludes of this paper.

2. ASSISTANT AGENT

2.1 Personal assistant agent for teachers

The personal assistant agent for teachers is to standardize the components of the language under investigation which every PAA-T present the component of language as the texts, structures and variables on the world wide web, such that the tutoring agent can deal with all languages in the same way. The PAAG-T consists of three parts, expertise module, semantic rules base, and tutoring text base. The personal assistant agent for teachers helps the teacher to cope with the knowledge base of a computer programming language under investigation, to add or modify the command's structure that will be taught, and to produce a meta level language representing this computer language, which we call "semantic rules" [7].

- Expertise module: The expertise module consists of domain knowledge that the teacher intends to teach to the student. Representative AI methods used to organize the domain knowledge in the expertise module include development of semantic networks, application of production systems, procedural representations, and building of scripts-frames [3,4]. In our system, a production rule to construct modular representations of commands is used.
- Semantic rules, When a teacher wants to build the semantic rule base, the screen of building the semantic rules base will appear containing the keys guide to build the knowledge base. In this phase, the teacher puts the structure of each command (using upper case characters for reserved keywords only), e.g. IF vov THEN s Means that one of the reserved keywords of basic language is called IF, and it can take the following structure (if the condition (variable operator variable) is satisfied then do the statements). Latter, personal assistant agent for students consults this semantic rules base to check the answers of the students.
- Tutoring text: The tutoring text-base contains the text that represents the commands of computer programming languages. Text is organized in terms of a conceptual network [8] hierarchically into lessons, sections, subsections, and terminal pages. Terminal pages contain the problems to be solved for the current command under investigation before it introduces new commands.

Each teacher can contribute in this tutoring text-base: he/she provides an optimal learning path for an assumed average student. The tutoring text will be retrieved by the name of the command. At the database, the extension of the command's name represents the teacher's ID number as well as the order of this text. The ID number of the teachers and the order of the text will be used to determine which text should be present first at personal assistant agent for students, taking into consideration the teacher's specialist (since a text of one computer programming language can be used in teaching another computer programming language. When the teacher wants to send his/her contribution to the tutoring agent, the following data should be sent: a) teacher ID; b) computer language's

name; c) command name; d) text describes the command; e) structure of the command; and f) variables used.

When the tutoring agent wants to consult a teacher about the correctness of student's experience which is stored at the blackboard database, the following data should be sent: a) student ID; b) student behavior record; c) computer language; d) command name; e) problem description; and f) suggestion of solution.

2.2. Personal assistant agent for students

The personal assistant agent for students consists of three components, student model, tutoring module and interface module.

2.2.1. Student model

The student model is used to assess the student's knowledge state and to make hypotheses about his/her conceptions and reasoning strategies employed to achieve the current knowledge state. where most intelligent computer aided instruction systems represent the student's knowledge state as a subset of an expert's knowledge base, the model is constructed by comparing the student's performance to the computer-based expert's behavior on the same task. Saeki [14] presents his opinion about the new system called CSCL, which supposes that the goal of the student is not the same for the teacher's goal in all cases. We select the "overlay" technique, since our domain knowledge is well defined, and our target students have the same goal of their teachers. However, the system has the ability to support students who have another goal (i.e. not to complete the course, instead for example, they want to check the structures of some commands and compare it with other computer programming languages they already know). In this case, the system does not build student behavior records for them, and it allows them to jump through the commands without the required sequence. Personal assistant agent for students invokes the student model to check the student answer. In order to accept free format answers from a student, one of the Compiler-phases had been used, it is called "Lexical Phase" [10]. The lexical phase is concerned with breaking up the string of characters into the words they represent.

2.2.2. Tutorial module

At the student accesses the module, the system asks about the computer's language, which the student wants to learn. The system downloads the tutoring dialog of this language and waits for the student to select the command he/she wants to practice, to retrieve the associated tutoring dialog's file(s). The system presents the text that describes the command, and asks the student if he/she understands it or not. If the reply is "no", the system presents another text for another teacher. If still the answer is "no" the system converts to another style for presenting the command or consults the blackboard database. If the reply is "yes", the system asks the student to write an example statement to check it. The system can accept a free format text from the student, and then infer the student model to check this statement. If the statement is correct, the system increments two counters, one for the number of questions which had been asked to the student, and the other is the correct answer counter. Otherwise, only the

first counter will be increased. After the student completes this step, the system re-presents the first menu to allow the student to select one of the followings: (a) select another command; (b) present the score; and (c) exit.

2.2.3 Interface module

Students can be sharing the experiences through the blackboard module. The blackboard module consists of three components:

1. A global database called the blackboard (hosted at the tutoring agent-Server);
2. Independent knowledge sources (from any personal assistant agent for students);
3. A scheduler to control knowledge sources and the blackboard database.

All experience elements are recorded in a structure, global database called the blackboard [5, 6]. The blackboard structure organizes experience elements along two dimensions, computer language name and command name. Each record contains the following fields (student ID, computer language, command name, problem description, and suggestion of solution). When a student faces a problem, and he/she could not catch the required meaning from the teacher's text, he sends a message to the message center at the blackboard module. A statement of the existing problem is displayed by the message center to all students on line. When a participant feels that he can contribute to the solutions, he sends his observations and/or conclusions to the message center. The scheduler organizes the knowledge source activity as well as the blackboard database and sends the contents from time to time to teachers to check its correctness, if it is correct it is moved to the previous dialog database (as shown in Fig. 1). Each teacher observes the messages written by others and considers those messages related to his specialty. Some of those teachers (who built the knowledge base for the same computer programming language under investigation) are able to offer immediate suggestions on what to do since their knowledge applies directly to the information currently in the message center. Others (who built the knowledge base for another computer programming languages), however, are forced to wait, possibly for an extended time, before their expertise is needed and can be applied. In most of the intelligent learning environments systems, only the tutoring component is adaptive. The user interface usually looks the same for the novice and for the advanced student, while the student's knowledge changes from the beginning to the end of a course [9]. In personal assistant agent for students we use the student model for creating an adaptive interface. This is done in several ways: (1) last state adaptation; (2) visual adaptive annotation of links; and (3) function panel adaptation. By this way, the student can feel better, because when the interface is oriented towards an experienced student, the interface appears to be too complex for a novice student, and vice versa. The browser downloads the HTML page and the applet code from the tutoring agent server to the personal assistant agent for students. The applet then runs on the client's computer. First, the applet downloads the data file

containing the initial variables about how the user interface should look like. Let us suppose that the data are stored in the file called test.dat, in the same directory as the HTML page. The applet then needs to open the URL, which is accomplished in the following JAVA code:

```
URL url = new URL (getDocumentBase(test.dat));  
where url is the object of URL class created by the operator new and the method getDocumentBase return the database file. This file, "test.dat" dynamically changes according to the student's performance, and every time the above construct invokes, the suitable interface will be downloaded.
```

3. STATE ADAPTATION

State adaptation means adaptation to the last state of the user-system interaction, i.e. the system keeps the "settings" of the individual user and comes up when starting like the last time the user worked with it. Personal assistant agent for students uses an extension of the traffic lights metaphor to annotate links visually. When presenting the current command tutoring under investigation, links to the other commands was annotated corresponding to a simple traffic lights metaphor referring to the knowledge state of each student. A red font of the link indicated that the corresponding command or section was not ready to be learned because necessary prerequisites were not met. The prerequisite relationships between commands are not represented directly, but the system is able to compute them from part-or and is-a relationships using several heuristics. A green font of the link indicated that this command or section was ready and recommended to be learned and a yellow font indicated that this command should have been learned. If the student visits a tutoring part of a command and did not successfully solve the exercise, the link of this command will be marked as a green font not as a yellow one in the next invoke.

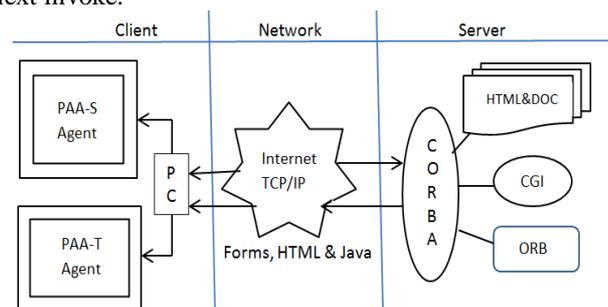


Fig. 3. The Expert system using CORBA.

3.1. Function panel adaptation

Showing all functions in the function panel can cause working memory overload and be confusing to the novice student [11] suggested the use of a conceptual overlay student model to hide all unlearned constructs from an editor's menu. PSS-A used the following three techniques in combination.

1. **Hiding:** elements of the user interface, which the student is not ready yet to use.
2. **Dimming:** using the stable layout of menus regardless of the student's level of knowledge.

3. Outlining: using different colors for the functions buttons, to outline which functions were learned, and which were not.

4. TUTORING AGENT

Tutoring agent contains tutoring modules, previous dialogs database, semantic rules base, and blackboard database. There are two main features in intelligent tutoring systems which are curriculum sequencing and interactive problem solving. These features differentiate intelligent learning systems from computer assisted instruction in that they incorporate intelligent techniques that skilled human teachers use in teaching classes [14]. Any computer language has command(s) for I/O, condition, loop etc. The way to teach those commands is almost the same for all computer languages, for instance, to teach the condition command in Basic will be done in quite the same way as the condition command in Fortran. The tutoring agent can construct the tutoring module for the new language by retrieving the tutoring dialog of similar commands in another language and adapt it. This is very useful to produce several tutoring texts from different teaching points of view to fit students needs. For example, if the tutoring agent has the tutoring dialog of the IF command in Basic, and then it consults the semantic rules of Fortran to substitute the Basic command in the tutoring dialog with the Fortran command.

5. EXPERT SYSTEM AND WWW INTERFAC

The Expert System version was reused much more of the component as possible. In the standalone version [15,17], students receive immediate feedback on every action they take. If they enter an incorrect statement, the system will present a message error and guide them to the correct answer. HTML forms do not allow this kind of tightly coupled interaction. The tutoring agent server receives information about student actions only when the student submits that information. In the Expert system, the tutoring agent, personal assistant agent for teachers and personal assistant agent for students agents can find and communicate with each other dynamically, using Common Object Request Broker Architecture (CORBA) as depicted in Fig. 3. CORBA allows agents to find each other and coordinate their behavior on a common object bus. This makes CORBA ideal for component-based applications. The advantages of using such technology in the Expert system is using those objects as a metaphor for using existing stand-alone ATCL applications as well as personalizing and task sharing between the client and server. In Internet based Expert, the JAVA ORB is used. With a JAVA ORB, an applet can invoke methods on CORBA objects using the IIOP protocol over the Internet. Consequently, there is no need to use HTTP and CGI programs that are the cause of extra overhead on the server, and also the client-side ORB enabled applets can be used in any JAVA enabled browser.

6. CONCLUSION

This paper summarizes the experience of Expert system to present teaching computer languages. Expert system

consists of three agents, tutoring agent, personal assistant agent for teachers and personal assistant agent for students. They communicate with each other as client server through internet. The system allows communicating with each other agents to exchange rules and tutoring text for different languages. The adaptive learning environment had been used in personal assistant agent for students. Moreover, the students can share their experiences through the blackboard system, i.e. the system can be dynamically enhanced by all members.

REFERENCES

- [1] Brusilovsky, P., Specht, M., & Weber, G. (1995). In F. Huber-Wa'sche, H. Schauer & P. Widmayer, Towards adaptive learning environments. GIS 95 (pp. 322–329).
- [2] Brusilovsky, P., Schwarz, E., & Weber, G. (1996). In C. Frasson, G. Gauthier & A. Lesgold, ELM-ART: an intelligent tutoring system on World Wide Web. Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS-96 (pp. 261–269). Berlin: Springer.
- [3] Bostrom, N. (2003). Ethical issues in advanced artificial intelligence. In I. Smit et al. (Eds.), Cognitive,emotive and ethical aspects of decision making in humans and in artificial intelligence (Vol. 2). Int.Institute of Advanced Studies in Systems Research and Cybernetics, pp. 12–17.
- [4] Cho, S.B.: Ensemble of structure-adaptive self-organizing maps for high performance classification. Inf. Sci. 123(1-2) (2000) 103-114.
- [5] Hartley, J. R., & Tait, K. (1986). Learner control and educational advice in computer based learning: the study station concept. Computers and Education, 10 (2), 259–265.
- [6] Hung, C., Wermter, S.: A dynamic adaptive self-organizing hybrid model for text clustering. In: Proceedings of the Third IEEE International Conference on Data Mining. (2003) 75-82.
- [7] Ian H. Witten and Eibe Frank (2011). Data Mining: Practical machine learning tools and techniques Morgan Kaufmann, 664pp., ISBN 978-0123748560.
- [8] Tohee, Z. (1999). Agent-based computer tutorial system—an experimental for teaching computer languages (ATCL). Special issue for intelligent agents for computer-based educational systems of the Journal of Interactive Learning Research. In press.
- [9] Lewis II, P. M., Rosenkrantz, D. J., & Stearns, R. E. (1978). Compiler design theory. Reading, MA: Addison-Wesley.
- [10] Lu, Haiping; Plataniotis, K.N.; Venetsanopoulos, A.N. (2012). "A Survey of Multilinear Subspace Learning for Tensor Data". Pattern Recognition 44 (7): 1540–1551.
- [11] L. Corno, R. E. Snow, "Adapting teaching to individual differences among learners," Handbook of Research on Teaching, pp. 605-629, 1986.
- [12] Nakabayashi, K., Maruyama, M, Koike, Y, Kato, Y., Touhei, H., & Fukuhara, Y. (1997). Architecture of

an intelligent tutoring system on the WWW. Proceedings of the Eighth World Conference of the AIED Society, Kobe, Japan.

- [13] O. Park, J. Lee, "Adaptive instructional systems," The Handbook of research on educational communications and technology, Issue 1911, Publisher: The Association for Educational Communications and Technology (AECT), pp. 651-684, 2003.
- [14] Saeki. (1999). The expectations and problems with CSCL: invited talk 25th CSCU meeting, Saitama University, Japan.
- [15] Pal, S., Dasgupta, B., Mitra, P.: Rough self-organizing map. Applied Intelligence 21(3) (2004) 289-299.
- [16] Rana, H., Rajiv, Prof. Lal, M., (2014). Rough set based system for effective e-learning. Proceedings of the 8th INDIACom-2014. 2014International Conference on Computing for sustainable global development (pp. 224-228). New Delhi.
- [17] Weber, G., & Specht, M. (1997). In A. Jameson, C. Paris & C. Tasso, User modeling and adaptive navigation support in WWW-based tutoring systems. User Modeling: Proceedings of the Sixth International Conference, UM97. New York: Springer.