2002

A Plugable Web Based Intelligent Tutoring System

Ang Yang
Massey University, angy_nz@yahoo.com

Kinshuk
Massey University, kinshuk@massey.ac.nz

Ashok Patel
De Monfort University, apatel@dmu.ac.uk

A PLUG-ABLE WEB-BASED INTELLIGENT TUTORING SYSTEM

Ang Yang
Department of Information Systems
Massey University, Palmerston North
New Zealand
Tel: +64 6 350 5799 ext 7024
Fax: +64 6 350 5725
angy_nz@yahoo.com

Kinshuk
Department of Information Systems
Massey University, Palmerston North
New Zealand
Tel: +64 6 350 5799 ext 2090
Fax: +64 6 350 5725
kinshuk@massey.ac.nz

Ashok Patel
De Montfort University
Director of CAL Research
Bosworth House, Leicester
United Kingdom
Tel: +44 116 257 7193
apatel@dmu.ac.uk

ABSTRACT

With the development of WWW technology, web-based ITSs are becoming mainstream area of research and development. The major benefit of web-based ITS is that, the ITS installed and supported in one place can be used by thousands of learners all over the world. Although many web-based educational systems appeared recently, most of them emerged from their predecessor legacy stand-alone systems. Therefore, they not only restrict themselves in functionality, but also fail to take advantage of distributed nature of Internet. This paper describes an open architecture based adaptable web-based intelligent tutoring system with pluggable domain modules. The system is based client/server architecture and has distinct and separable domain modules and a generic module. Such architecture not only provides salability in the Internet environment but also allows the same architecture to be used for multiple subject domains.

1. INTRODUCTION

The Internet is now becoming more and more important and is being integrated into all kinds of education and training. A large number of learners with all age and background are using computer networks for different interests and motivations. Internet has generally been successful in keeping up with these motivations and interests by providing useful information and a greater sense of enjoyment through new ways of cognition. In the current educational environment the available resources to the traditional education sectors are constrained. But the demand on educational opportunities is increasing. This makes alternative approaches such as web-based education even more attractive for academic institutions.

Leigh (1996) observed that the Internet can provide an instantaneous dissemination of information to a wider audience and education at a distance can meet the financial needs of both the institution and the learner often proving a very effective alternative to traditional instruction or training. However, the students may not prefer to learn at a distance since the traditional distance learning methods lack interaction and attraction. In this scenario, the Internet may even prove more beneficial than the traditional distance learning methods.

With the development of WWW technology, web-based ITSs are becoming mainstream area of research and development. The web-based ITSs installed and supported in one place can be used by thousands of learners all over the world (Brusilovsky et al., 1996a). Although many web-based educational systems appeared recently, most of them emerged from their predecessor legacy stand-alone systems. Therefore, they not only restrict themselves in functionality, but also fail to take advantage of distributed nature of Internet. These systems are usually dedicated to a restricted knowledge base and have closed architecture with little possibility of modifications once released.

This paper describes an open architecture based adaptable web-based intelligent tutoring system with pluggable domain modules. The system has distinct and separable domain modules and a generic module. The domain modules contain the specifications of domain knowledge, and other domain dependent parameters. The generic module includes three components: user interface, overlay student model and inference mechanism. The dynamic instances of these generic components are created on the basis of domain specific information from available domain modules. In this way, different knowledge domain may be plugged in the system by using different domain modules.

2. CLIENT/SERVER ARCHITECTURE

Recently a lot of web-based ITS systems appeared on the internet. Example include C-Book (Kay et al., 1994), RAPITS (Woods et al., 1995), WEST-KBNS (Eklund et al., 1996), ELM-ART (Brusilovsky et al., 1996), ELM-ART-II (Weber et al., 1997), CALAT (Nakabayashi et al., 1997), InterBook (Brusilovsky et al., 1997a), AST (Specht et al., 1997), DCG (Vassileva, 1997), InterBook (Brusilovsky et al., 1997), 2L670 (Calvi et al., 1997), Adele(Shaw et al., 1997), PAT online (Brusilovsky et al., 1997b), ADIS (Warendorf et al., 1997), AlgeBrain (Alpert et al., 1999) and VC PROLOG (Peylo et al. 2000). Most of these systems used a client/server architecture that places shared resources and functionality on servers, and uses internet to deliver student interfaces on a wide variety of client platforms at any location with internet access. This architecture is aimed to improve the deployability and interoperability of knowledge-based educational software without sacrificing advanced functionality.

The main characteristic of a client/server application is that a part of the application (Client) requests the execution of some job, while the other part of the application (Server) executes this job. All the functions of the application must be split into the two programs: the client program, which is executed on the user's machine and the server program, which is executed on the host machine.

The client program provides an interaction with a user, the input of the data, and the presentation of the results. It is able to communicate with the server program. It sends the requests and receives the results (Schank, 1994). On the other hand the server program, which is the heart of a client/server application, receives requests from the client, executes them and sends the results back to the client. All the main and computationally expensive operations are implemented in the server program.

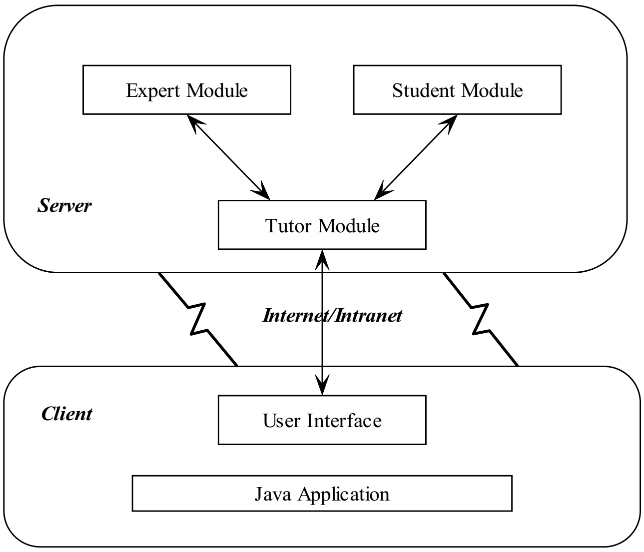


Figure 1. Architecture of system

Our system adopts this client/server architecture and the components of a classical ITS architecture are distributed between client and server (figure 1). Its applied inference engine is based on the common overlay architecture suggested by various researchers (Wenger, 1987; Burns & Capps, 1988; Mandl & Lesgold, 1988) and consists of four basic components, namely, expert knowledge module, student model, tutoring module and user interface module. The prototype system is based on inter-relationships of variables in the Marginal Costing domain and creates an interactive approach for problem setting and problem solving.

3. PROTOTYPE SYSTEM

3.1 Knowledge Domain

The system consists two main modes: the traditional learning mode (figure 2), which facilitates the learning of concepts and examples of the topic; and the interactive mode (figure 3), which allows the students to practice concepts.

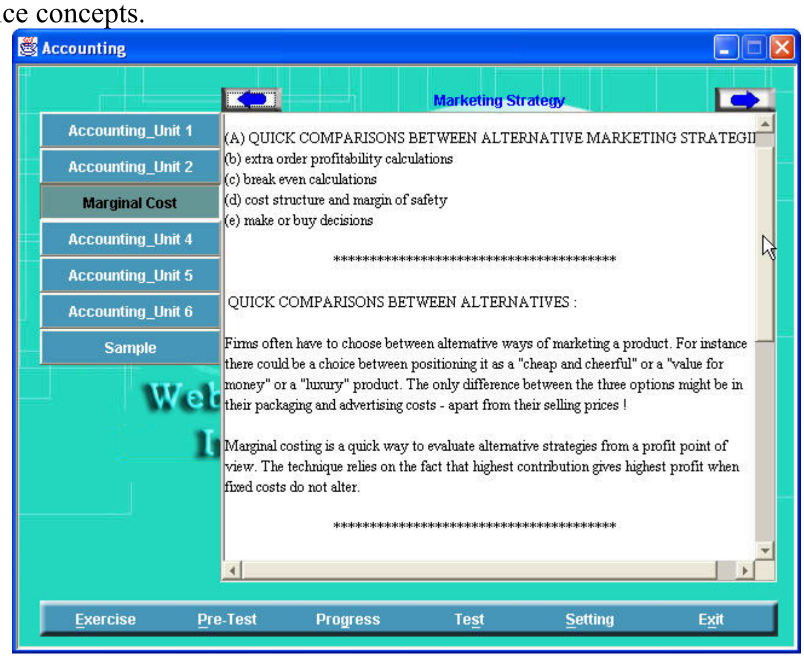


Figure 2. Traditional Learning Mode

Figure 3. Active Learning Mode

In our program, knowledge base is stored in a MySQL database on the server. The human tutors may configure knowledge base at the server. They may delete topics or add topics. Each topic usually consists of several units and each unit may contain several chapters and variables. After logging in, the students may select the topic they want to learn (figure 4).

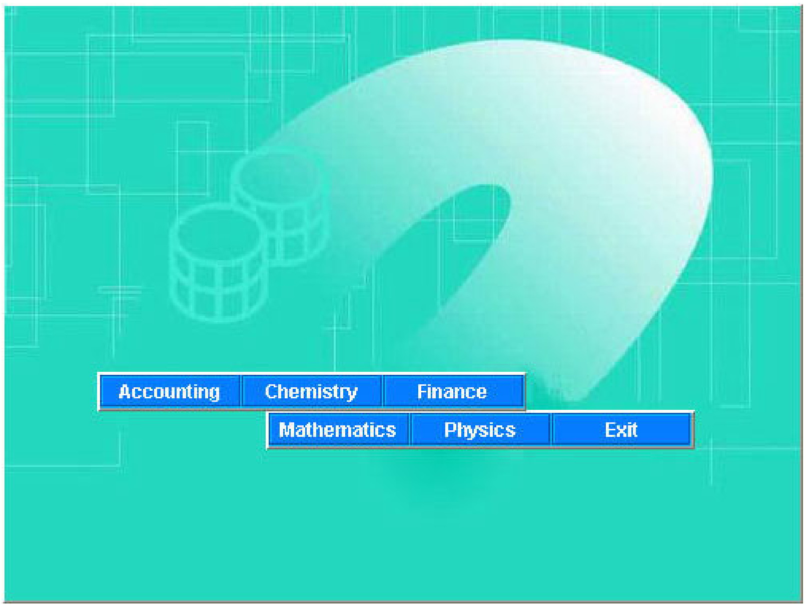
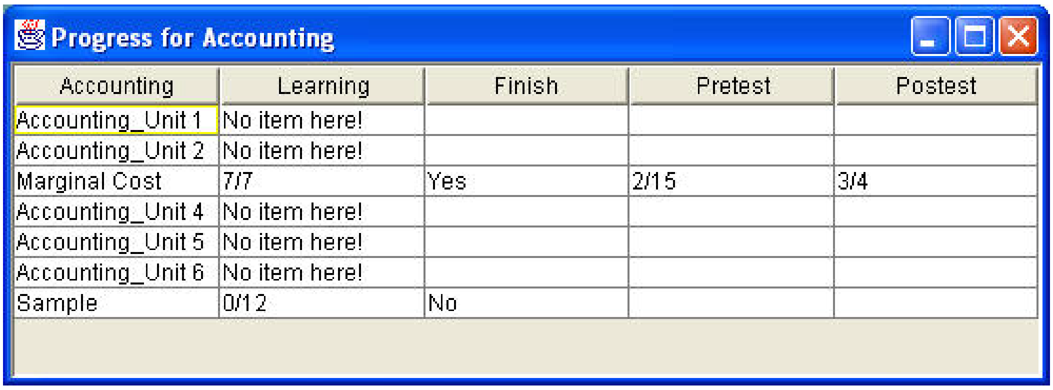


Figure 4. Topic Selection

3.2 Student Model

Overlay student model is used in our system, where the student's knowledge is assumed to be a subset of the expert's knowledge and the goal of tutoring is to enlarge this subset. When students learn the topic and navigate within the learning material, the system stores the progress in the database. Next time, when students log in, system starts where the previous session was stopped. If students finish one unit, system informs them that the unit is finished and post-test is available. While students take tests, system's inference mechanism calculates the knowledge acquired by the student in that particular unit and stores the result in the database. Both human tutor and students may check the learning progress (figure 5).



Progress for Accounting

Accounting	Learning	Finish	Pretest	Postest
Accounting_Unit 1	No item here!			
Accounting_Unit 2	No item here!			
Marginal Cost	7/7	Yes	2/15	3/4
Accounting_Unit 4	No item here!			
Accounting_Unit 5	No item here!			
Accounting_Unit 6	No item here!			
Sample	0/12	No		

Figure 5. Learning Progress

3.3 Tutoring Model

System uses tutoring model to guide students' learning. According to the terminology used by Self (1988), corrective and elaborative aspect of student model are used for tutoring in our system. The corrective behavior of the software rejects a wrong value and conveys that it was incorrect. If there is second consecutive mistake, then the elaborative behavior comes into action, suggesting an appropriate relationship by which the value can be derived. The suggestion depends on what student has done so far. Diagnostic, predictive and strategic aspects of student model are not considered in the design. The learning process is broken down to very small steps through suitable interfaces and adoption of this strategy coupled with immediate dynamic feedback eliminates the need to consider these aspects.

In an interactive knowledge acquisition process, there are two types of feedback. The first type is based on the "Road to London" paradigm (Patel, 1995) and with its focus on the immediate goal (often a sub-goal on the path to an overall goal), the feedback is concerned only with the correct way of attaining the immediate goal. A prerequisite of such interaction is continuous monitoring and immediate feedback preventing a student from proceeding along a wrong path. Thus with immediate corrective feedback the learning is concentrated on the know-how (learn by doing) aspect. This type of feedback is very suitable for novices as it allows learning in small doses. The prototype produced gives this kind of feedback for the learner. For example, when the student makes mistake, the system provides graded feedback with inserting the correct answer in the last step of the feedback process.

3.4 Inference Mechanism

The inference mechanism is based on the inter-relationships of variables in the knowledge domain. All the variables are separated into two types: primitive (independent) variables and dependent variables. The value of any primitive variable cannot be calculated from existing variables at the time of their assignment. All the values of dependent variables are derived from primitive variables. There are two types of interactive learning mode in our system. One is where the system randomly generates the values of all primitive variables. Another is where the student enters the value of the primitive variables. This second mode, coupled with teacher's expert solution enables the system to identify correct method but wrong interpretation on the part of the student, facilitating partial marking just like a human tutor would.

4. CONCLUSION AND FUTURE WORK

Classic architecture of computer-based intelligent tutoring system can be used in web-based intelligent tutoring system. It only needs to distribute the components between server and client. In our system, the student model and knowledge domain remain at the server end, and the user interface and tutoring model are transferred at the client end. Messages are transferred between client and server through Internet.

Using appropriate knowledge domain modules, which are stored in database, our web-based intelligent tutoring system has been made plug-able. That means different knowledge domain can be plugged into the system at any point of time. This feature enables different knowledge domain to be learned in the same system.

Two adaptive techniques are used: adaptive presentation and adaptive navigation. Adaptive presentation means the system presents different information to different users according to users' behaviors, competence and other attributes. In our system, such adaptation typically takes place during assessment process where the system offers the students only those tests that are ready for testing but the student has not yet completed. Using adaptive navigation, the system limits browsing space. For example, only after finishing learning a unit or successfully completing all questions in the exercises, student is offered the post-test.

This prototype till now is only implemented for the domain of Marginal Costing. Further work is required to add more units and topics into the system. Furthermore, the task of knowledge addition in the system is to be handed-over to the teacher. For that purpose, a sophisticated yet user-friendly authoring component is being developed that acquires the domain knowledge from the teacher using knowledge-by-example approach. In this approach, the teacher attempts to solve various exercises in the system and system monitors the teachers' actions to infer various relationships that are practically important for learning purposes. Such an approach ensures that human teachers remain focus on the domain knowledge they are expert in and do not require any technical knowledge to interact with the system.

The system described in this paper takes the real advantage of interaction and distributed nature of the Internet. Coupled with the authoring module, it is expected to provide an integrated solution for adaptive learning on the Internet.

5. REFERENCE

1. Alpert, S.R., Singley, M.K., & Fairweather, P.G. (1999). Deploying Intelligent Tutors on the Web: an Architecture and an Example. *International Journal of Artificial Intelligence in Education,* 10, 183-197
2. Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An Intelligent Tutoring System on World Wide Web, In: C. Frasson, G. Gauthier and A. Lesgold (Eds.), *Proc. of 3rd International Conference on Intelligent Tutoring Systems, ITS-96,* Montreal, June 12-14, 1996. Springer Verlag, Berlin, 1996, pp. 261–269.
3. Brusilovsky, P., Schwarz, E. & Weber, G. (1996a). A tool for developing Adaptive Electronic Textbooks on WWW. *WebNet 96*.
4. Brusilovsky, P. & Schwarz, E. (1997). User as student: Towards an Adaptive Interface for Advanced Web-based Applications. In: A Jameson, C. Paris and C. Tasso (eds.) *Proceedings of 6th International Conference on User Modeling*, Chia Laguna, Sardinia, Italy, Wien: Springer-Verlag, 177-188.

5. Brusilovsky, P., Eklund, J., & Schwarz, E. (1997a). Adaptive Navigation Support in Educational Hypermedia on the World Wide Web, in: *Proceedings of INTERACT97, the 6th IFIP World Conference on Human–Computer Interaction,* Sydney, Australia, 14–18 July, 1997. New York: Chapman and Hall, 1997, pp. 278–285.
6. Brusilovsky, P., Eklund, J., and Schwarz, E. (1997b). Adaptive navigation support in educational hypermedia on the World Wide Web, in: *Proceedings of INTERACT97, the 6th IFIP World Conference on Human-Computer Interaction,* Sydney, Australia, 14-18 July, 1997. New York: Chapman and Hall, 278-285.
7. Bums, H.L. & Capps C.G. (1988). Foundations of intelligent tutoring systems: an introduction. *Foundations of intelligent tutoring systems.* Poison M.C. & Richardson J. J. (eds.) Lawrence Eribaum, London, 1-19.
8. Calvi, L. and De Bra, P. (1997). Using dynamic hypertext to create multi-purpose textbooks, in: T. Muldner and T. C. Reeves (eds.), *Proceedings of ED-Media/ED-Telecom'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications,* Calgary, Canada, June 14—19, AACE, 1997. 130-135.
9. Eklund, J. & Sawers, J. (1996). Customising Web-based Course Delivery in WEST® with Navigation Support, in: *Proceedings of WebNet'96, World Conference of the Web Society*, San Francisco, CA, October 15–19, 1996, pp. 534-535.
10. Kashihara A., Kinshuk, Oppermann R., Rashev R. & Simm H. (2000). A Cognitive Load Reduction Approach to Exploratory Learning and Its Application to an Interactive Simulation-based Learning System. *Journal of Educational Multimedia and Hypermedia*, 9 (3), 253-276 (ISSN 1055-8896).
11. Kay, J. and Kummerfeld, R.J. (1994). An Individualised Course for the C Programming Language, in: *Proc. of 2nd International WWW Conference,* Chicago, IL, 17–20 October, 1994
12. Leigh, D. (1996). *The Internet and Distributed Learning: Instructional Designer's Medium and Tool*. Available: http://mailer.fsu.edu/~dleigh/superflux/words/dleigh/NetDistLearn.html.
13. Mandle, H. & Lesgold, A. (1988). *Learning Issues for Intelligent Tutoring Systems*. London: Springer-Verlag.
14. Nakabayashi, K., Maruyama, M., Kato, Y., Touhei, H., & Fukuhara, Y. (1997). Architecture of an Intelligent Tutoring System on the WWW, in: B.D. Boulay and R. Mizoguchi (Eds.), *Proceedings of AI-ED'97, World Conference on Artificial Intelligence in Education,* Kobe, Japan, 18–22 August 1997. IOS, Amsterdam, 1997, pp. 39–46.
15. Patel, A. (1995). Intelligent Tutoring Tools: An evaluation of TLTP Byzantium's approach to software design for a computer integrated learning environment in introductory accounting. *Project report,* De Montfort University, UK, September 1995.
16. Peylo, C., Thelen, T., Rollinger, C. & Gust, H. (2000). *A Web-based Intelligent Educational System for PROLOG*, URL: http://virtcampus.cl-ki.uni-osnabrueck.de/its-2000/paper/peylo/ws2-paper-9.html
17. Schank, J. D., (1994). *Novell's guide to Client - Server Applications and Architecture,* Novell Press, San Jose, 1994.
18. Self, J. (1988). Student models: what use are they? *Artificial Intelligence Tools in Education,* P. Ercoli & R. Lewis (eds.) North Holland, Amsterdam, 73-86.
19. Shaw, E., Johnson, W. L and Ganeshan, R. (1997). *Pedagogical Agents on the Web*, URL: http://www.isi.edu/isd/ADE/papers/agents99/agents99.htm, (On July 11, 2000).
20. Specht, M., Weber, G., Heitmeyer, S., & Schöch, V. (1997). AST: Adaptive WWW-Courseware for Statistics, in: P. Brusilovsky, J. Fink and J. Kay (Eds.), *Proceedings of Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling,* UM97, Chia Laguna, Sardinia, Italy, June 2, 1997, pp. 91–95, http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Specht.html
21. Vassileva, J. (1997). Dynamic course generation on the WWW, in: B.D. Boulay and R. Mizoguchi (Eds.), *Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education,* Kobe, Japan, 18-22 August 1997. IOS, Amsterdam, 498-505.
22. Warendorf, K. & Tan C. (1997). ADIS - An Animated Data Structure Intelligent Tutoring System or Putting an Interactive Tutor on the WWW. *Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web", 8th World Conference of the AIED Society*, Kobe, Japan, 18-22 August 1997

23. Weber, G. & Specht, M. (1997). ELM-ART, in: A. Jameson, C. Paris and C. Tasso (Eds.), *Proc. of 6th International Conference on User Modeling,* Chia Laguna, Sardinia, Italy, June 2–5, 1997. Springer-Verlag, Wien, 1997, pp. 289–300.
24. Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos: Morgan Kaufmann.
25. Woods, P. & Warren, J. (1995). Rapid Prototyping of an Intelligent Tutoring System, *Proceedings of Twelfth Annual Conference of the Australian Society for Computers in Learning in Tertiary Education* (ASCILITE95), Melbourne, December 1995, pp. 557-563.