Integrating JADE Agents into Moodle

Adrian Scutelnicu[a], Fuhua Lin[b], Kinshuk[b], Tzu-Chien Liu[c], Sabine Graf[d], Rory McGreal[b]

[a]Ford Motor Company of Canada, Canada
[b]Athabasca University, Canada
[c]National Central University, Taiwan
[d]Vienna University of Technology, Austria
oscarl@athabascau.ca

ABSTRACT

This paper presents an approach to integrating the JADE-based multi-agent system (MAS) with the Modular Object-Oriented Dynamic Learning Environment (Moodle) - an open source learning management system. This approach tries to overcome some shortcomings of Moodle that limit the benefits of Web-based distance learning, from various perspectives such as collaboration and course administration. A prototype of JADE agents for automated forum monitoring demonstrates this approach and shows the feasibility of implementing MAS-supported learning management systems.

Keywords

Learning management systems, multi-agent systems, system architecture, system integration

INTRODUCTION

Web-based learning management systems (LMSs), such as Moodle, WebCT, and Blackboard, are commonly and successfully used in distance learning. They provide a variety of features to support instructors and course developers to create and manage their online courses. However, currently such environments provide very little, if any, intelligent support for instructors and students. For example, in our university, although the current version of Moodle is used by instructors to build online learning communities, it offers passive services only. As a result, instructors end up spending too much of their time simply monitoring students' progress and participation by visiting many Web pages and using different Moodle tools. This includes, for example, monitoring the message board activities log to verify student participation. Software agent technologies can be used to make the above-mentioned tasks much easier for instructors [3] [12] [5]. Software agents have been defined as programs that act on behalf of a human being by locating and accessing information, resolving inconsistencies, filtering irrelevant or unwanted information, integrating information from several sources, and adapting to a human's needs [10].

In this research, the aim has been to develop a methodology for incorporating intelligent agents into currently used learning management systems (LMSs) to provide a better educational environment. Specifically, this paper focuses on the integration of software agents with Moodle (http://moodle.org/) to improve its capabilities by creating a monitoring agent based tool for its forum and by adding new functionality to the Web-based forum interface in Moodle.

The agents presented in this paper perform various tasks, relieving the instructor from manual monitoring and management of course activities and contents.

The rest of the paper is structured as follows. Section 1 presents the literature review on intelligent agents in learning systems. Section 2 describes the architectural design and a description of how the agents work. Furthermore, the implementation of the architecture as well as experimental results are discussed. The paper concludes with summary of the presented work and suggestions for future research.

1. LITERATURE REVIEW

Agent technology can be used in different ways to extend LMSs [4] [8]. Some examples for the effective use of agent technology are given by Thaiupathump et al. (1999) [11] and Ayala (2002) [1]. Furthermore, Moodie & Kunz (2003) [7] proposed an Intelligent Learning Management System (iLMS), which includes agents for helping the teachers finding appropriate learning objects and analyzing learners' understanding of the key concepts of the course and helping them by suggesting them to further study components based on each learner's behavior. Suh & Lee (2006) [9] developed an extensible collaborative learning agent that was used to promote interaction among learners.

However, developing agent-supported learning environments are non-trivial [8]. What is needed is a methodology for the design, development, integration, implementation, and maintenance of agent-supported LMS.

The fundamentals of multiagent system (MAS) have been found to be very appropriate for designing intelligent LMS. The MAS-based architecture allows great flexibility and scalability in the integration of components and provides a simple yet extensible and powerful software layer to develop further online learning environments.

A number of agent or MAS toolkits are available in the literature. One of them, JADE (Jave Agent Development Environment, http://jade.cselt.it), embodies extensive experience in the implementation of large agent systems. JADE is a robust and efficient environment for distributed agent systems. It was

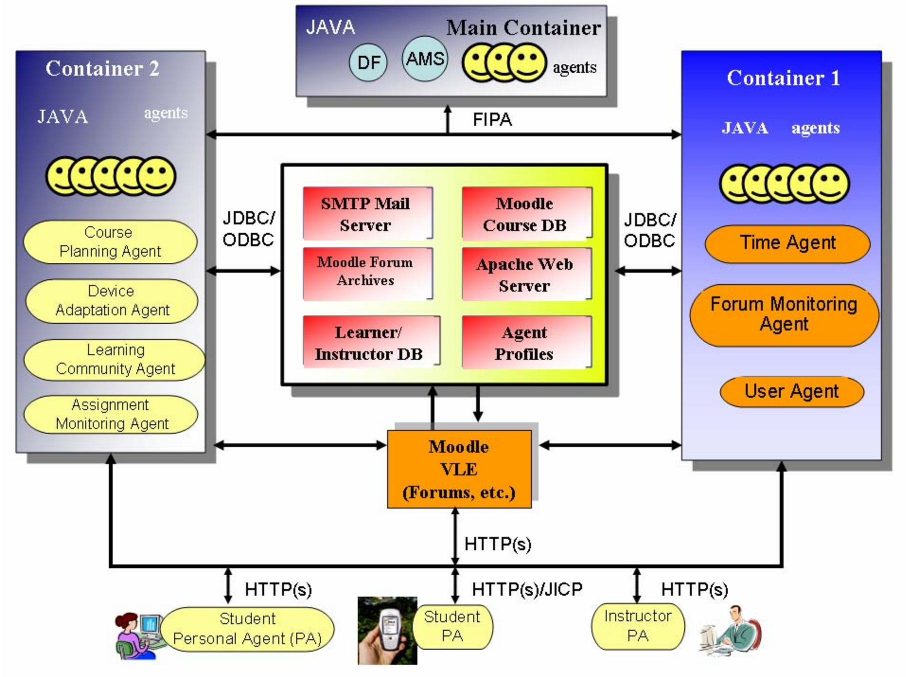


Figure 1: The proposed system architecture.

developed in Italy jointly by Telecomm Italia Lab in conjunction with the Computer Engineering Group of the University of Parma. JADE follows the Foundation for Intelligent Physical Agents (FIPA) standards.

We used JADE in our proposed methodology because [2]: (1) it has all the agent features that we needed, including scalability and flexibility; (2) communication between agents running on various workstations on the network is trivial to do; (3) it is efficient and tolerant of faulty programming; it followed FIPA standards; (4) the user group is very active; and (5) it is free. Furthermore, Java-based JADE can interact relatively easily with Java implementations of Prolog or Expert systems (JESS) to add “intelligence” into software agents.

Our work explores the methodology for the design and development of an agent-supported LMS.

2. ARCHITECTURE DESIGN

The system architecture is described in Figure 1 and it has the following components:

Web client: Consisting of a browser interacting with the Moodle Virtual Learning Environment (VLE) or receiving email or text messages by PDA or cell phones.

Moodle Virtual Learning Environment (VLE): Providing a communication interface where users interact with the virtual learning environment. For example, through the Moodle forums the participants can discuss by posting messages. In the proposed architecture, there are some *adaptive user interfaces* with the Moodle VLE for users to configure their agents and receive messages from the agents. For example, *Moodle forum database* is used to store the forum posts by the users.

Servers and databases: Including toolkits and databases (such as SMTP mail server, Moodle databases, Forum archives, Apache Web server, Agent profiles, user databases, and so on) supporting Moodle VLE and JADE agents.

JADE Agent Platforms: A JADE platform is composed of agent containers that can be distributed over the network. Agents live in containers which are Java processes that provide the JADE run-time and all the services needed for hosting and executing agents. There is a special container, called the main container, which represents the bootstrap point of a platform:

3. AN EXAMPLE: AGENTS FOR AUTOMATED FORUM MONITORING

3.1 Types of agents

Following agents have been developed for automated forum monitoring:

Time Agent: This agent creates and starts the instances of the Forum Client Agent and instances of the Forum Monitor Agent based on interval frequencies set up in the configuration of the forum. It also checks the completion status of the assignment and the timeframe elapsed from the previous execution and it does incremental readings of the workspace database.

Forum Client Agent: It handles the calls to the Forum Monitor agent.

Forum Monitor Agent: The Forum Monitor Agent is responsible for monitoring the activities of the students. This agent acts as an assistant and is invoked by an instance of the Time Agent based on a predefined timeframe. The agent performs data aggregation by querying the Moodle forum database. The agent makes decisions, merges the results, and sends customized reports to the students and instructors, and also posts messages to the forum based on the students' activity. Figure 2 depicts the workflow of the Forum Monitor agent. The Forum Monitor agent performs the following activities:

(1) It builds and sends a report showing the number of posts per student.

(2) It sends the students a "Managing Conflicts" status report based on the posts which fall under the "argue" subcategory of the Collaborative Learning Skills Taxonomy structure [6]. For example, a student is determined to have a possible conflict if the number of disagreement-type "argue" posts subcategory (such as, "I disagree because", "Alternatively", "So', "If", "But", "I'm not sure") within the same thread is greater than three.

(3) It groups all the posts by sentence openers and by students and sends each student in the course an email showing the number of posts grouped by the learning skills defined in [6]. The indicator for the type of learning skill is provided by the sentence opener part of the post. Each post has a sentence opener part as a mandatory field where the student selects one of the skill and sub-skill before sending the posts. It retrieves a list of students who have registered for the course and have not posted any comments to the forum. Then, it sends the instructor an email listing all the students with no posts. It counts the number of learning skills posted, grouped by the learning skills.

3.2 Interface integration and agent delegation

The JADE agents perform without human interaction once they are delegated by the instructor through setting their execution time intervals in Moodle (see Figure 3). This agent delegation can be viewed as a way of Agents and Moodle integration. The "active/inactive" status of the agents and the time interval at which the data is retrieved from the Moodle database can be setup up by the instructor.

3.3 Inter-agent Interaction

There are three types of agents in the experimental system: *Time Agent*, *Forum Client Agent*, and *Forum Monitor Agent*. They communicate through exchanging FIPA-ACL (Agent Communication Language) messages. An instance of the Time Agent acts as a container and is responsible for creating and starting the instances of Forum Client Agent and the instances of Forum Monitor Agent. After the Forum Client agents and Forum Monitor agents accomplish their tasks, they terminate their lifespan. After the Forum Client agents and Forum Monitor agents finish running, the Time Agent terminates as well. The Forum Client Agent handles the requests (message posts) submitted by the Time Agent and it sends a request to the Forum Monitor Agent, which performs data aggregation on the Moodle database, and builds and sends reports by email to corresponding students based on the predefined set of rules.

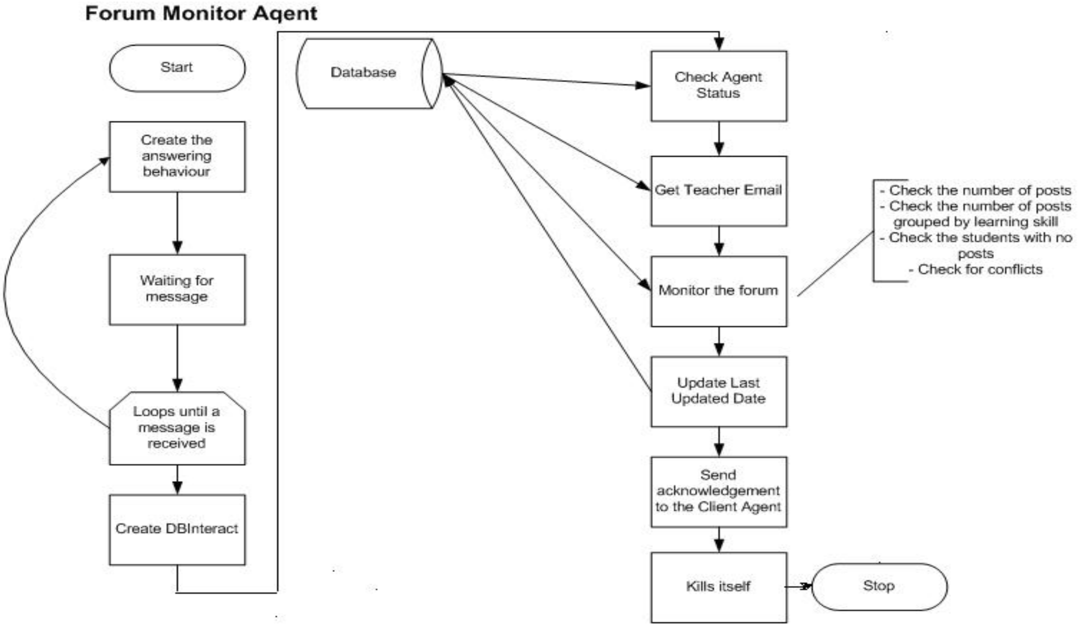
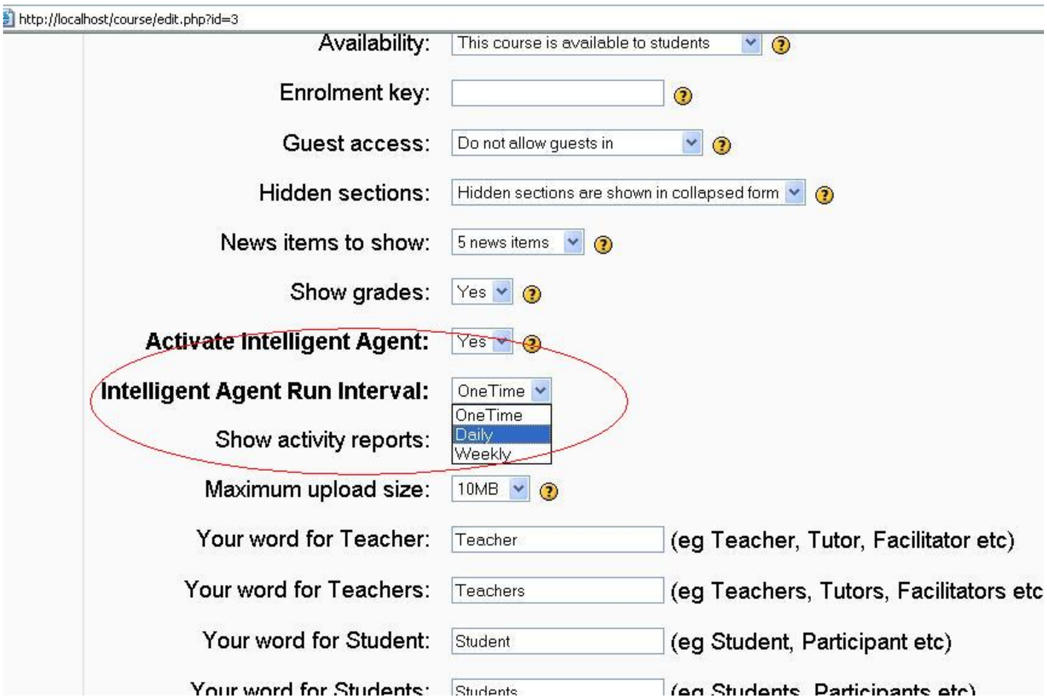

Forum Monitor Agent
Start
Create the answering behaviour
Waiting for message
Loops until a message is received
Create DBInteract
Database
Check Agent Status
Get Teacher Email
Monitor the forum
Update Last Updated Date
Send acknowledgement to the Client Agent
Kills itself
Stop
- Check the number of posts
- Check the number of posts grouped by learning skill
- Check the students with no posts
- Check for conflicts

Figure 2: Flowchart for the Forum Monitor Agent.



http://localhost/course/edit.php?id=3

Availability: This course is available to students

Enrolment key:

Guest access: Do not allow guests in

Hidden sections: Hidden sections are shown in collapsed form

News items to show: 5 news items

Show grades: Yes

Activate Intelligent Agent: Yes

Intelligent Agent Run Interval: OneTime

Show activity reports:

Maximum upload size: 10MB

Your word for Teacher: Teacher (eg Teacher, Tutor, Facilitator etc)

Your word for Teachers: Teachers (eg Teachers, Tutors, Facilitators etc

Your word for Student: Student (eg Student, Participant etc)

Your word for Students: Students (eg Students, Participants etc)

Figure 3: Agent delegation in the interface for the Instructor.

As soon as the Time agent is started, it reads the time interval for which the agent should check the data. Then, it creates and starts the Forum Client Agent and the Forum Monitor Agent based on the interval frequency set up on the Moodle forum. For example, if the interval is set in the Moodle to “One time”, it will retrieve the entire data stored in the database of the forums from the beginning of the course. If the interval is “Daily” or “Weekly”, it will read the data only for the time frame specified so it will check whether a day/week has passed since the last check. If the agent runs at a date greater than a day/week since the last read, the agent will retrieve the data between the last checked date and the run date, and it will set a CHECKED flag in database with last checked date for the next time it will run. The Forum Client Agent handles various ACL requests to the other agents, such as Forum Monitor agent. For example, it can send the Forum Monitor agent a “MONITOR” message to poll the database in Moodle. Once the Forum Monitor agent has received the “MONITOR” request, it will first check the agent status and if it is “Inactive” it will not proceed. If the agent is “Active” it will perform the data aggregation by querying the Moodle database.

3.4 Data Aggregation

Data aggregation is performed by the Forum Monitor Agent through using Java Database Connectivity (JDBC) APIs to connect to the Moodle MySQL databases. The agent performs the following operations against the Moodle databases for a selected time interval, which may be daily, weekly, or the entire course period:

- It calculates the number of posts performed by each student;
- It checks if any students have zero posts;
- It checks whether some students may be in conflict by checking whether the disagreement-type “Argue” posts for a student is greater than three for the same thread;
- It calculates the number of posts grouped by learning skills;
- It checks and summarizes all the students’ posts based on the learning skills; and
- It inserts new posts into the Moodle forum database regarding the inactivity of the students on the forum.

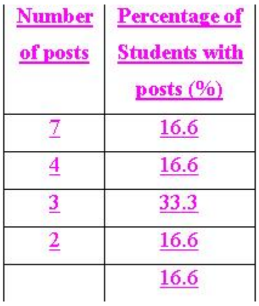
3.5 Experimental results

We have developed the following five Java classes:

- *DBInteract.class*,
- *ForumEmail.class*,
- *ForumAgent.class*
- *TimeAgent.class*, and
- *ForumMonitor.class*,

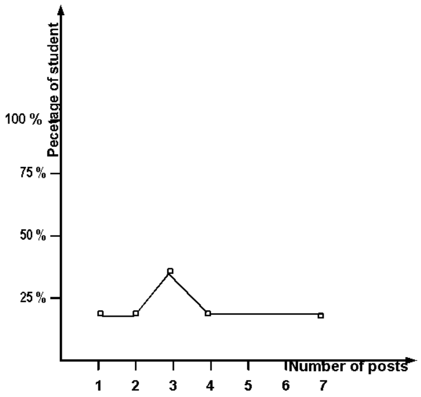
which implement the JADE middleware technology. The Moodle Web application forum was developed in PHP, which runs under the Apache Web server.

The results are based on a set of simulated sample data, and the following statistics were calculated: the percentage of students with no posts: 30%; the percentage of students who sent messages which fall under the “Argue” learning skill category: 11.11%; the number of posts per percentage of students with posts;

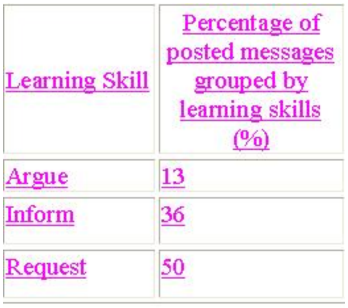


Number of posts	Percentage of Students with posts (%)
7	16.6
4	16.6
3	33.3
2	16.6
	16.6

(a)



(b)



Learning Skill	Percentage of posted messages grouped by learning skills (%)
Argue	13
Inform	36
Request	50

(c)

Figure 4: The experimental results.

and the percentage of learning skills grouped by learning skill types (please see Figure 4 (a), (b), and (c)).

Based on the above sets of results, the following statistics can be inferred during the entire course or per interval: the average number of posts for each student, the average type of opening sentences, and the average number of inactive/very active students. The progress of the students' activity in time can also be determined from which the teachers can draw conclusions regarding their interest in the course, forum, and posted subjects. Also, the list of opening sentences based on the learning skills can be enhanced by studying the mostly used/unused learning skills. The effectiveness of the warning messages posted for students with no posts can be measured by analyzing the activity of the students in time after such messages are posed. At last, the interest of the students in different subjects and courses can be measured by comparing the results for multiple courses.

4. CONCLUSION AND FUTURE WORK

We have presented a novel methodology for incorporating software agents into an LMS - Moodle. The agent-supported Moodle LMS can provide both students and teachers with the results of the activity of the students by performing data aggregation on the Moodle database. We expect that the system will increase the quality of learning by:

(1) providing a tool to measure and monitor the students' activity on collaborative learning. It will help increase the students' interest in the learning process and will enhance the critical thinking among the learners by brainstorming, making decisions, debating, setting priorities on a project, formulating tasks, taking actions, and solving problems. It will also encourage passive members to participate more in discussions by posting messages to the forum explaining that there are students with no posts;

(2) making the teachers' work easier in estimating the students' activities related to a course by studying the active participation of the students; and

(3) motivating students to use the Moodle LMS by keeping them up-to-date with the statistical information and their activity on their collaboration work for a course.

Our ultimate goal is to overcome some shortcomings of the current LMSs such as the lack of intelligent support and the lack of reporting.

The advantages of the architecture are two-folds. First, the system is extensible. For example, the user agent in this architecture sends a "MONITOR" to the Forum Monitor agent responsible for managing the forum. The user agent may be extended to send other requests for other types of analysis. Second, the system is platform-independent. As it supports different operating systems, it can be integrated with other systems. The agents are written in Java using JADE framework, so the code is easy to maintain because JADE middleware provides easy-to-use API methods for agent communication. The JADE framework was fundamental in the design and development of the architecture, which provides interoperability, uniformity and portability, ease of use, and asynchronous agent communication.

We plan to evaluate the proposed integration of agents and investigate if it helps students and instructors. We will enhance the agents through: (1) Adding a bigger variety of posts and replies by the agents on the forum; (2) Creating an interface where some questions can be created, which can be posted by the agents at predefined times and where some milestones can be set as extra activity on the course. Also, new components can be added, which could perform other monitoring tasks for different Moodle functions such as, assignment module, resource module, survey module, and workshop module. Furthermore, the system can be easily enhanced to also produce all of the above statistics grouped by the main thread by listing the thread name and body. This way it would be easy to detect the most active/inactive subjects posted.

ACKNOWLEDGMENTS

We thank NSERC (The Natural Sciences and Engineering Research Council of Canada) and Athabasca University of Canada for supporting this project. We also thank Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) for supporting parts of the research (under grant 31.963/46-VII/9/2002). We thank two anonymous reviewers for their constructive comments on earlier drafts of the paper.

REFERENCES

[1] Ayala, G. (2002). Intelligent agents for a lifelong learning environment in information technologies, *International Journal of Continuing Engineering Education and Lifelong Learning*, Vol. 12, No.1/2/3/4, pp. 189-200

[2] Bellifemine, F., G. Caire, & D. Greenwood, (2007). Developing Multi-agent Systems with JADE, WILEY.

[3] Chou, C. T. Chan, C. Lin (2003). Redefining the Learning Companion: The Past, Present, and Future of Educational Agents, *Computers & Education*, vol. 40, 2003, pp. 255-269.

[4] Dowling, C. (2000), Intelligent pedagogical agents in online learning environments**,** Educational Uses of Communication and Information Technologies, Beijing, China, Aug. 21-25.

[5] Lin, F., Esmahi, L., & Poon, L. (2004). Integrating Agent Technology and Web Services into Distributed Learning Environments. In F. Lin (Ed.), Designing Distributed Learning Environments with Intelligent Software Agents, Information Science Publishing, pp. 184-217.

[6] McManus, M. & Aiken, R. (1995), Monitoring computer-based problem solving. *Journal of Artificial Intelligence in Education* 6(4), pp. 307-336.

[7] Moodie, P. & P. Kunz, (2003). AIED 2003 workshop proceedings, Towards Intelligent Learning Management Systems**,** July 20.

[8] Jafari, A. (2002). Conceptualizing intelligent agents for teaching and learning. *Educause Quarterly,* vol. 3.

[9] Suh, H-J, & S-W Lee, (2006), Collaborative Learning Agent for Promoting Group Interaction, *ETRI Journal*, Vol. 28, No. 4, August 2006, pp. 461-474.

[10] Sycara, K., K. Decker, A. Pannu, M. Williamson, & D. Zeng. (1996), Distributed intelligent agents. IEEE Expert, 11(6):36-46.

[11] Thaiupathump, C., Bourne, J., & Campbell, J. (1999). Intelligent agents for online learning. *Journal of Asynchronous Learning Networks, 3*(2). Retrieved May 17, 2004, from http://www.sloan-c.org/publications/jaln/v3n2/pdf/v3n2_choon.pdf

[12] Zhang Y., Kinshuk, Yang J., Smestad O. & Jeffery L. (2006). Using Agents for Enhancing Learning Effects in an Advanced Discussion Forum. *14th International Conference on Computers in Education*, Nov. 20-24, 2006, Beijing, China.