

Software Agents

to Assist in Distance Learning Environments

*Software agents
can act as
teaching assistants
for distance
learning courses
by monitoring and
managing course
activities*

By **Sheung-On Choy, Sin-Chun Ng, and Yiu-Chung Tsang**

The Open University of Hong Kong (OUHK) is a distance education university with about 22,500 students. In fulfilling its mission, the university has adopted various Web-based and electronic means to support distance learning. For instance, OUHK uses a Web-based course management system (CMS) to provide students with a flexible way to obtain course materials. Apart from traditional print-based course materials and textbooks, the university staff and instructors use the Web to replace surface mail to deliver assignment files, supplementary study materials, and other correspondence with students. Online discussion forums provide students with a convenient place to ask questions and receive answers from the teaching staff.

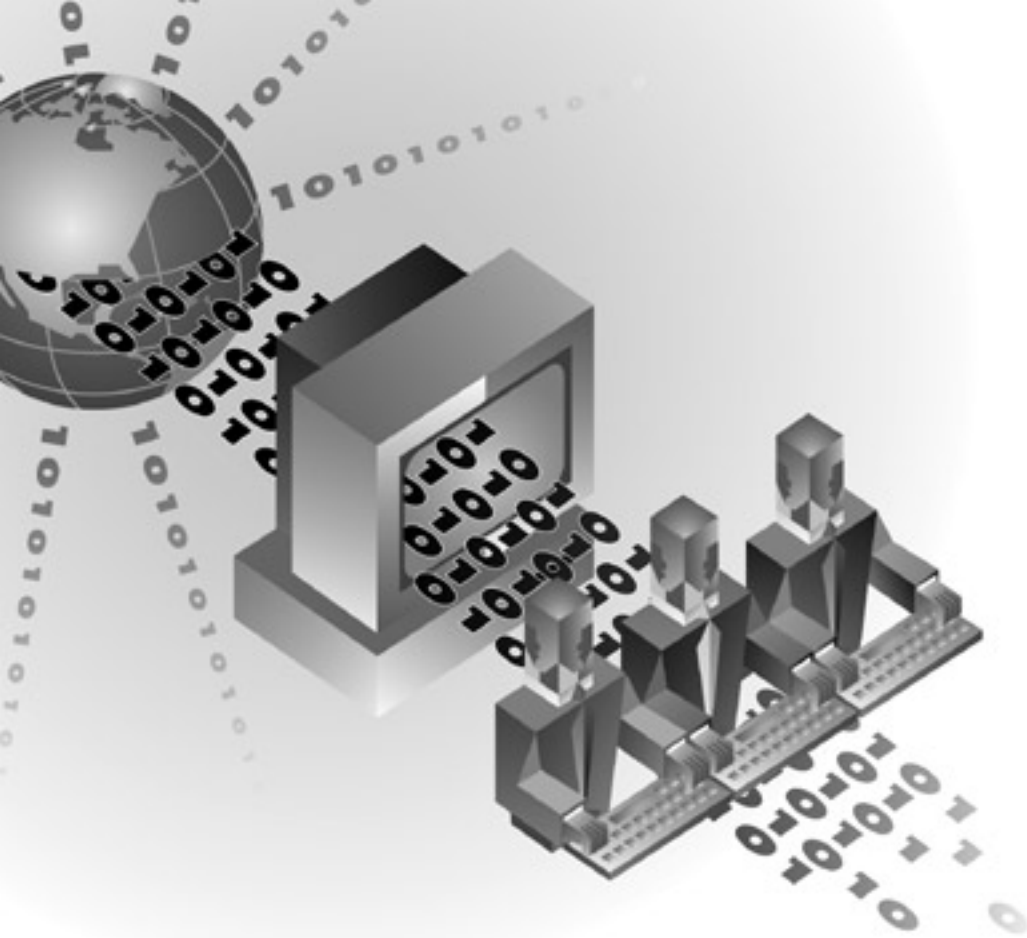
An electronic assignment-handling system administers student assignments. Assignments, which are marked up and commented on, are considered an important element in the distance learning process. The assignment-handling system enables students to submit assignments electronically, apply for an extension, inquire about the status of their submitted assignments, and receive timely feedback on assignments

from the teaching staff.

Clearly the Web-based CMS provides many benefits in the service of distance education. However, it is also fair to say that the current ways that we use and interact with the CMS are not ideal. Employing a human teaching assistant to relieve academic staff from manual monitoring and management of course activities is one way to improve the process, but not effectively and efficiently. We investigated employing a software agent to act as a teaching assistant to the course coordinator instead—the idea behind this article.

The Process Explained

The key players in the OUHK education process are students, tutors, and academic staff. Students normally spend much of their time studying alone. To complement the distance learning process, face-to-face sessions are provided, including tutorial, day school, and laboratory sessions. Part-time tutors conduct the face-to-face sessions. Students in a course are required to submit homework assignments to their tutors (usually four to five times during the course). Tutors then mark the assignments and provide feedback to students.



Every course at the OUHK has an academic staff member as the course coordinator. The coordinator's major responsibility is to supervise the teaching and learning process for that course. This includes making assignments and scheduling examinations, delivering the course materials and supplementary teaching notes, and monitoring students' learning progress, tutors' teaching performance, and the assignment-marking process.

The use of the Internet and Web-based technologies to support large-scale distance learning has led to a significant change in the working pattern of OUHK academic staff. In order to ensure smooth course presentation, improve adherence to procedures, and enhance students' comfort levels and retention, course coordinators expend a lot of time and effort in interacting directly with various Web-based systems and communicating with students and tutors over the Internet. The following list summarizes tasks that an OUHK academic staff member would usually do for a distance-learning course:

- Post course material and assignment files.

- Post course schedules and details of upcoming face-to-face sessions.
- Post announcements about the course, such as changes to the course material.
- Remind students of assignment submission and due dates.
- After the assignment due date, contact those students who did not complete their assignments.
- Monitor tutor progress in marking assignments to ensure that submitted assignments are marked and commented upon within an established period of time.
- Identify inactive students and contact them for proactive consultation.
- Assist those students who are having problems completing the course.

Figure 1 illustrates the interactions of distance-learning students, part-time tutors, and the academic staff in an OUHK course that uses a Web site as a common interface for course presentation, communication, and student support. The server hosting the course Web site contains a lot of data about students' learning progress and their behavior when interacting with the course Web site. The information can be useful in

providing timely and meaningful feedback on student progress and aids in early detection of problems in both the teaching and learning processes.

Unfortunately, acquiring information from the course Web server requires initiating all information retrieval tasks and interacting directly with the replies from the server. In other words, to manage the course and acquire useful information, the user must interact directly with the course Web server, initiate all tasks explicitly, and monitor and process all the server's responses. This direct-manipulation method of interaction¹ is labor-intensive, consuming a lot of time for unintelligent information retrieval and filtering. To check students' progress, for example, the user has to visit many Web pages regularly and use different tools within the server to verify student progress and participation. This includes regularly checking the course activity log to monitor students' online activities, visiting the assignment submission records to see if students have submitted their assignments, and reviewing the assignment marking records to identify students with problems.

Employing a human teaching assistant to interact with the CMS is one option, of course, but with most of the same problems as involved in having academic staff perform these functions. We propose creating a software agent to act as a teaching assistant to the course coordinator instead.

Software Agents

The term software agents originated from the field of artificial intelligence, back in the 1950s. Research on software agents started to proliferate in the mid-1990s, after several key agent-related papers appeared in the popular comput-

ing press, followed by several books. (See the sidebar for research publications on software agents.)

Software agents can be defined in a number of ways depending on the functions and context. A general and widely accepted definition is that soft-

ware agents are computational autonomous entities capable of sensing (such as acquiring information) and acting (such as producing and sending information) in an environment (such as a Web server) to accomplish a set of designated goals.

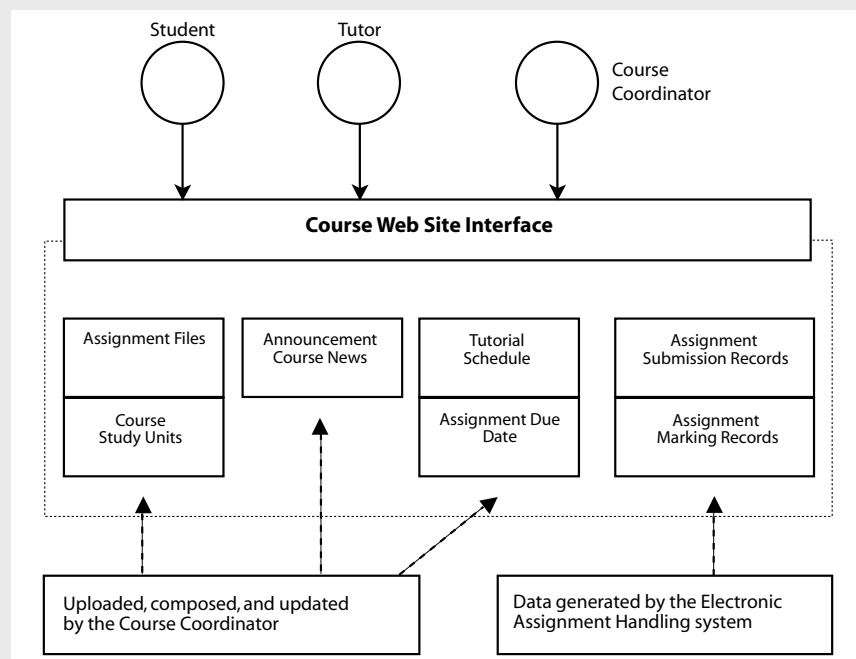
The domain of software agents is divided into two types: autonomous *interface/information agents* (interface agents for short) and *multi-agent systems*.² The concept of interface agents involves the provision of agents that enable the user to advance from direct manipulation of systems to indirect human-computer interaction by delegation to autonomous software agents. Multi-agent systems are more complex. The goal is to create a system that interconnects separately developed agents and to enable the ensemble to function beyond the capabilities of any single agent in the system.³

Interface agents demonstrate great potential, as reflected in multiple developments in various application domains. Existing use includes auctions and electronic markets, entertainment, e-mail filtering, news filtering, expert assistance, matchmaking, recommendation systems, and Web domain management.⁴

In the context of this article, the term software agents refers to personal interface agents. We are interested in the application domain specific to teaching a distance-learning course within the OUHK setting. This article reports on the implementation and realization of software agents as personal teaching assistants in the distance-learning environment. The software agents work on behalf of academic staff and have the authority and autonomy to interact with the course Web server.

Figure 1

Interactions in the Course Web Site



Research Publications on Software Agents

Bradshaw, J. M., ed., *Software Agents* (Boston: MIT Press, 1997).

Maes, P., "Agents that Reduce Work and Information Overload," *Communications of the ACM*, Vol. 37, No. 7, 1994, pp. 31–40, <<http://www.acm.org/>>.

Middleton, S. E., "Interface Agents: A Review of the Field," Technical Report Number ECSTR-IAM01-001, University of Southampton, August 2001.

Norman, D., "How Might People Interact with Agents?" *Communications of the ACM*, Vol. 37, No. 7, 1994, pp. 68–76.

Nwana, H., "Software Agents: An Overview," *The Knowledge Engineering Review*, Vol. 11, No. 3, 1996, pp. 205–244, <<http://journals.cambridge.org/bin/bladerunner?30REQEVENT=&REQAUTH=0&116000REQSUB==&REQSTR1=KER>>.

Nwana, H. S., and D. T. Ndumu, "A Perspective on Software Agents Research," *The Knowledge Engineering Review*, Vol. 14, No. 2, 1999, pp. 1–18, <<http://journals.cambridge.org/bin/bladerunner?30REQEVENT=&REQAUTH=0&116000REQSUB==&REQSTR1=KER>>.

Ovum Report, *Intelligent Agents: the New Revolution in Software* (London: Ovum Publications, 1994).

Reinhardt, A., "The Network with Smarts," *Byte*, October 1994, pp. 51–64.

Wooldridge, M., and N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review*, Vol. 10, No. 2, 1995, pp. 115–152, <<http://journals.cambridge.org/bin/bladerunner?30REQEVENT=&REQAUTH=0&116000REQSUB==&REQSTR1=KER>>.

Related Work

A number of researchers have proposed the development of software agents in teaching and learning situations. Jafari⁵ conceptualized three types of software agents to assist teachers and students:

- Digital Teaching Assistant—assists the human teacher in various teaching functions
- Digital Tutor—helps students with specific learning needs
- Digital Secretary—acts as a secretary to assist students and teachers with various logistical and administrative needs

Our work involves the first type of software agent.

Razek, Frasson, and Kaltenbach⁶ proposed the application of software agents to provide distance learning students with timely and useful information in a group discussion. The software agent they built can observe conversations among a distance learning community, interpret the community's input, organize the information, and then present information useful to the students in their current activities.

Our work also proposes using software agents to make the communication in a distance learning community more effective. However, we focus on the communication between teachers and students. The software agent we built is designed to work on behalf of teachers, assisting them in communicating more effectively and closely with students.

Suzuki and Yamamoto⁷ proposed an agent-based distance learning system for effective delivery of distance learning courses. Their system provides a personalization service to each set of courseware, allowing it to customize its content and present materials in a context-sensitive way.

Distance learning considered from a learner perspective basically involves four types of interactions:

- Learner-to-interfaces
- Learner-to-content
- Learner-to-learner
- Learner-to-instructor⁸

Suzuki and Yamamoto's work⁹ focused on using software agents to enhance the first two types of interactions for a distance-education course. Razek, Frasson, and Kaltenbach's work¹⁰ contributed to the third type of interactivity factor. Our work proposes applying a software agent to enhance the fourth type of interactivity factor in a distance education environment—learner-to-instructor interaction.

Software Agents as Teaching Assistants

As mentioned earlier, the main actors in the education process of an OUHK course are students, part-time tutors, and an academic staff member as the course coordinator. Most of the time, these three actors are geographically

and temporally remote from each other, and they use the course Web site as the online environment for course delivery, assignment management, and communication. Therefore, the course Web server contains a lot of data about students' learning progress and their study behavior.

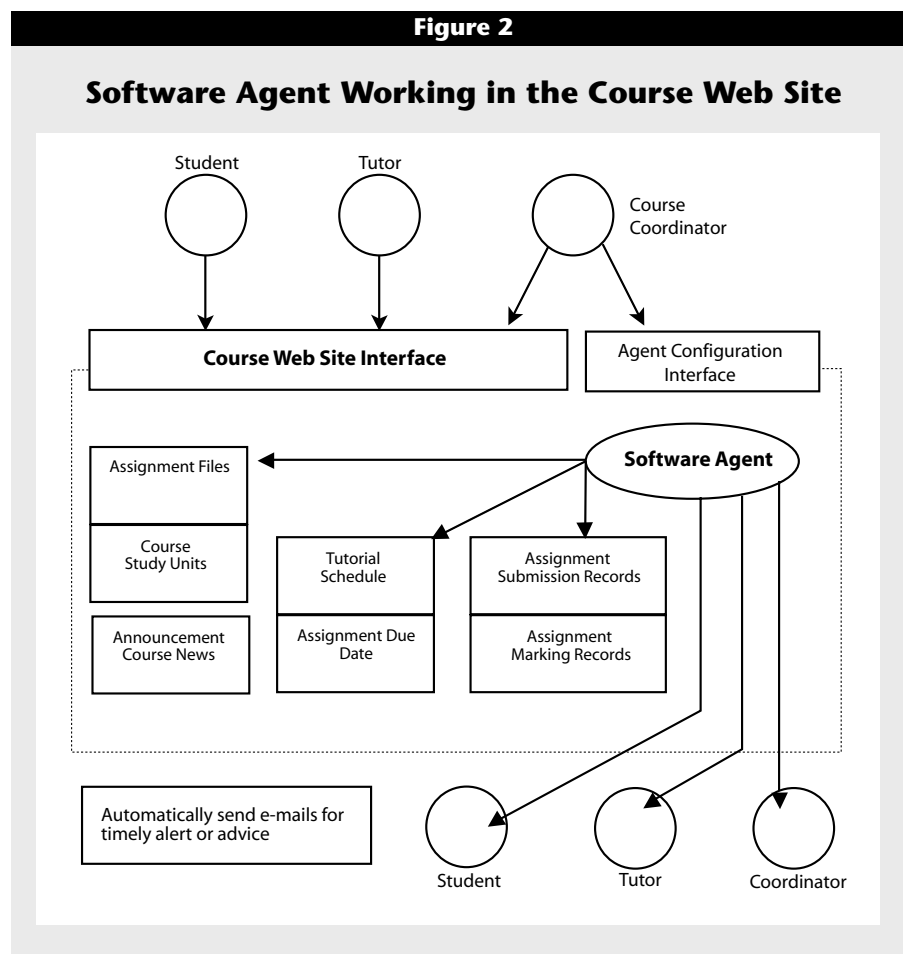
We propose a software agent working in the course Web environment to assist the course coordinator in performing the tasks requiring tedious and direct interactions with the course Web server. Figure 2 depicts how the proposed software agent would work. The agent can be configured through an agent configuration interface to perform its jobs on behalf of the course coordinator. It can retrieve information such as that about students' learning progress and study behavior, aggregate the information, and send e-mails to the coordinator to report status. When appropriate or necessary, the agent can represent the course coordinator in sending e-mails

to tutors or students for timely alerts or reminders.

Based on the course coordinator's usual responsibilities (summarized earlier in the task list), we wanted the software agent to perform the following duties. Note that none of these tasks was new because of the availability of the software agent.

- Send alert e-mails to inactive students (those who have not accessed the course Web site for a long period of time). The course coordinator decides the length of the inactive period and instructs the software agent to send the alert e-mails based on the established time.
- Send e-mails to inform tutors about inactive students and advise tutors to have proactive consultation with those students.
- Send e-mail alerts to those students who have not downloaded a particular piece of course material or who have not read an important piece of

Figure 2



course news since it was uploaded to the file server. This helps prevent students from missing information or forgetting to download an important item, such as an assignment file.

- Help the course coordinator keep track of students' progress, and send e-mails to the coordinator and tutors about those students whose performance is at a marginal level.
- Retrieve information from the course timetable and send reminder e-mails to students. For example, it might send a reminder to students five days before an assignment due date and one day before a face-to-face session.
- During the period when assignments are submitted, the agent will monitor the assignment submission status and send e-mails after the due date to those students who have not submitted the assignment. It will also inform the course coordinator and tutors about those students.

We did not follow a rigorous procedure in coming up with this list of tasks—we simply selected them from those of the course coordinators' list of tasks that can be clearly expressed without the need for further clarification or judgment. They are tasks that can be easily delegated to an autonomous software agent. In the absence of software agents, these tasks consumed massive human effort for unintelligent information retrieval and processing.

Development and Implementation

We first installed Red Hat Enterprise Linux (<http://www.redhat.com/>) as the operating system on a server connected directly to the Internet. This machine is used as the software agent's execution environment. We also installed an Apache Jakarta Tomcat (<http://jakarta.apache.org/tomcat/>) server in the machine as the Web server. The Tomcat server is a Web application container that can run Java Servlets and JavaServer Pages in Web applications. We chose it because our implementation of the software agent was based on these two Java technologies, and the Tomcat is an open-source, stable, Java-based Web

server with good performance.

The software agent has to access course presentation information including the schedule of face-to-face sessions and assignment due dates. The agent also needs to access course news and announcements posted by the course coordinator. For this purpose, we developed a course Web interface for the coordinator to perform the following tasks:

- Enter or update course presentation information.
- Post course news and announcements.
- Upload assignment files and supplementary study material.

All this information is stored in the server's database, which the software agent can access to perform its duties. In particular, the agent can send e-mails to students and tutors to:

- Remind them about upcoming face-to-face sessions
- Remind students to submit their assignments before the due date
- Inform students that a new course item (such as a course announcement or an assignment file) has been posted on the course Web site for them to read or download

We also developed a Web interface

for students to use in subscribing to the course reminder service. Students can also choose the frequency and quantity of those reminder e-mails. Figure 3 shows the interface we built for this service.

The software agent can access information relating to assignment submission, mark-up progress, and students' assignment scores. The agent needs this information to perform its analysis and reach some conclusions that may be useful for early detection of problems in either the teaching or the learning process. For this purpose, we developed the agent to the extent that it can log into the university's electronic assignment submission and recording system (ASRS), interact with the system to obtain the necessary information, filter the information, and send e-mails to the course coordinator to report its findings. The agent needs the coordinator's authorization (user name and password) before logging into the ASRS. For example, one task the software agent can do on behalf of the course coordinator is locate—after the due date—those students who have not submitted an assignment. The agent reports this to the coordinator by e-mail.

Next we describe implementation of

Figure 3

Reminder Service Interface

Application for Course Reminder Email Service HELP ?

Please choose the desired service and no. of days notice before an event happens.

☒ News	
☒ Tutorial	1 day ☒ 3 days ☐ 5 days ☐ 7 days ☐
☒ Surgery	1 day ☒ 3 days ☐ 5 days ☐ 7 days ☐
☒ TMA Cutoff	1 day ☒ 3 days ☐ 5 days ☐ 7 days ☒

Please enter an email address to receive system email reminder.

Email

To unsubscribe from the entire service, please tick the box below.

☐ Unsubscribe

SUBMIT

the software agent at a more technical level. Figure 4 shows the main elements, which we implemented using Java servlets and the related technology. To start the process, we put the “ScheduleLife servlet” in the server to accommodate the agent. The servlet will start and initialize execution of the software agent when the server starts and stop the agent when the server shuts down.

To facilitate the software agent’s performance in the server, we implemented several Java classes:

- *Schedule Control Class* is the program code to control the software agent’s life and behavior in carrying out various tasks.
- *Job Listener Class*, which listens to Schedule Control Class, generates an error notification e-mail to report to the system administrator when exceptional events occur.
- *Data Retrieval Class* retrieves all required information at once for the Job Central Processing Class. The information includes students, tutors, the course coordinator, the schedule of face-to-face sessions, assignments, and course news.
- *Job Central Processing Class*, which is invoked by Schedule Control Class, accepts information provided by Data Retrieval Class. This class hosts the criteria to generate e-mail alerts to participants, composing e-mails and sending them to the appropriate recipients when the criteria are met.

The software agent is scheduled to run every day. It sleeps until the current server time meets the scheduled time defined in its configuration file. After it “wakes up,” the agent retrieves all necessary data from the system database or from outside sources. In particular, it obtains information about the participants’ (including the course coordinator, tutors, and students) e-mail addresses and their communication preference (when and what types of information). After all information has been prepared, the software agent composes e-mails and sends them to the corresponding participants using the system’s mail server.

Figure 5 shows the software agent’s operational flow. Monitoring Agent

Figure 4

Agent Architecture

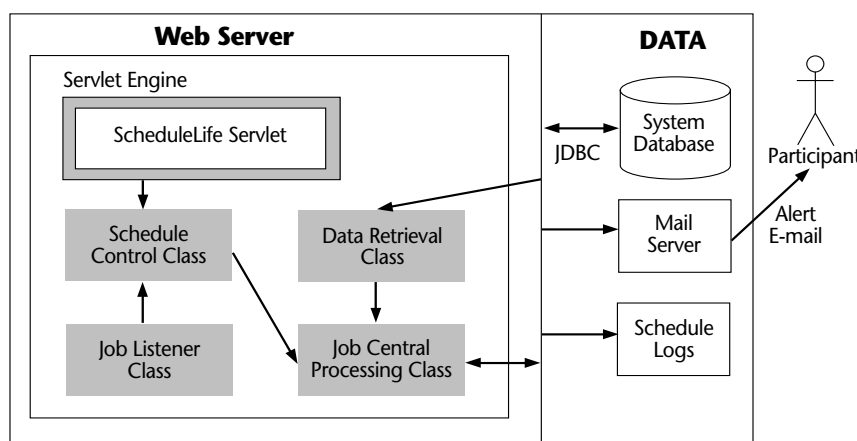
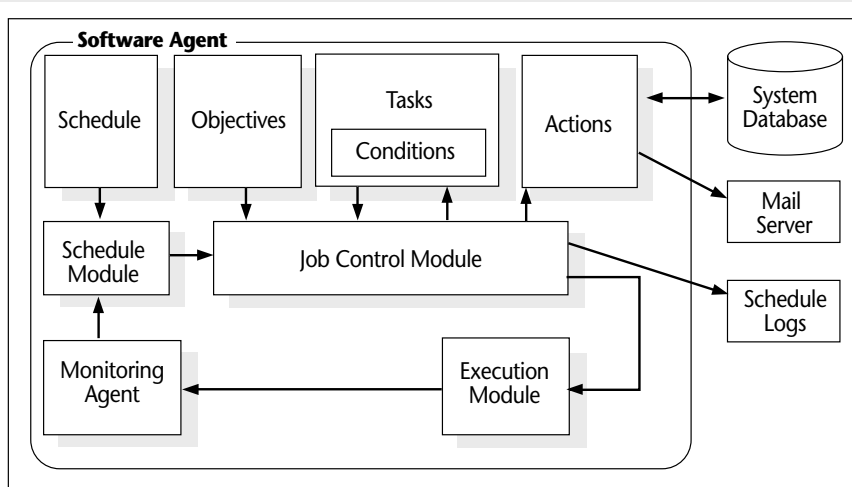


Figure 5

Software Agent’s Operational Flow



monitors the software agent’s scheduled time. When the scheduled time arrives, the Scheduler Module invokes the Job Control Module, which looks into the Objectives defined previously. According to the Objectives defined, different tasks require handling that are governed by different conditions. When all the required conditions for a specific task have been fulfilled, the control module takes corresponding action(s) by using previously retrieved

information to compose an e-mail alert and automatically send it to recipients on behalf of the course coordinator. At the same time, the Job Control Module updates the system database according to the information sent to recipients. After all tasks have been performed, the module records the total number of reminder e-mails sent out to all participants in each course. Then the Job Control Module returns control to the Execution Module, which invokes the

Monitoring Agent again to monitor the next scheduled time.

Discussion

We have piloted use of the software agent as a teaching assistant in four OUHK distance learning courses, of which one commenced in April 2004 and three in October 2004. Choy and Ng are the course coordinators and the sole users to interact with the software agent to date. As mentioned before, the ultimate function of the implemented agent was to take over routine jobs handled by the course coordinators. Because the coordinators handle those routine jobs in the absence of the software agent, its existence was not noticed by students and tutors. When the agent sent reminder e-mails to students or tutors, it wrote on the coordinator's behalf. It also sent a duplicate e-mail to the coordinator for recording purposes. During the pilot period, all the software agent's operations, such as accessing course information, processing information, and sending e-mails to the course coordinators, students, or tutors, functioned properly.

Choy and Ng agreed that the software agent was helpful and concluded that course coordinators could save a lot of time by delegating those routine jobs, which involve direct interactions with the course Web site and tedious information processing, to the software agent. It is fair to say that students experienced no immediate impact, since, after all, the software agent worked as an administrative assistant to the course coordinators. Nevertheless, from time to time the coordinators received positive feedback. Students commented that their teacher was enthusiastic about their studies and aware of their learning progress and performance in the course. They appreciated the close communication between coordinators and students—maintained by the software agent. This communication, in turn, helped create a good learning atmosphere—the indirect benefit of the software agent to students.

We developed the software agent to work as a software robot to assist the teaching staff in the OUHK distance



During the pilot period, all the software agent's operations, such as accessing course information, processing information, and sending e-mails to the course coordinators, students, or tutors, functioned properly.

learning environment. We can further increase the agent's capabilities in advancing the project. For example, the software agent could perform a number of useful tasks with the information stored in the assignment-management system. It would also be valuable to develop the agent to the extent that it can read the discussion messages among students and tutors in the online discussion forum, analyze them, and report its findings to the teaching staff.

At the time we wrote this article, the OUHK software agent had only been used by its advocates—us—in our courses. We are convinced, however, that software agents offer good support of distance education. Still, the developer of a tool no doubt finds it helpful and easy to manage when others might not enjoy the same experiences, so we know the potential difficulties facing us in promoting the use of software agents in our university. The institution-wide adoption of software agents to assist distance learning teaching requires institutional policy to emerge during the evolution of this new technology. We hope that further development of the software agent's capabilities will prompt the development of supportive institutional policies and increased adoption of this technology at OUHK. *e*

Acknowledgment

The project described in this article was funded by Earmarked Research Grant Ref: 9004/03H. We would like to thank the Open University of Hong Kong and the Education and Manpower Bureau for funding that allowed the system to be developed.

Endnotes

1. P. Maes, "Agents that Reduce Work and Information Overload," *Communications of the ACM*, Vol. 37, No. 7, 1994, pp. 31–40, <<http://www.acm.org/>>.
2. S. E. Middleton, "Interface Agents: A Review of the Field," Technical Report Number ECSTR-IAM01-001, University of Southampton, August 2001; and H. S. Nwana and D. T. Ndumu, "A Perspective on Software Agents Research," *The Knowledge Engineering Review*, Vol. 14, No. 2, 1999, pp. 1–18.
3. Nwana and Ndumu, op. cit.
4. Middleton, op. cit.
5. A. Jafari, "Conceptualizing Intelligent Agents for Teaching and Learning," *EDUCAUSE Quarterly*, Vol. 25, No. 3, 2002, pp. 28–34, <<http://www.educause.edu/ir/library/pdf/eqm0235.pdf>>.
6. M. A. Razeq, C. Frasson, and M. Kaltenbach, "A Context-Based Information Agent for Supporting Intelligent Distance Learning Environments," *Proceedings of the Twelfth International World Wide Web Conference*, held May 20–24, 2003, in Budapest, Hungary, <<http://www2003.org/cdrom/papers/alternate/P344/WWW2003-Razeq-344/Razeq-344.htm>>.
7. J. Suzuki and Y. Yamamoto, "Building a Next-Generation Infrastructure for Agent-Based Distance Learning," *International Journal of Continuous Engineering Education and Life-Long Learning*, Vol. 12, Nos. 1–4, 2002, pp. 299–317, <<http://www.cs.umb.edu/~jxs/pub/ijceell.pdf>>.
8. D. B. Ehrlich, "Establishing Connections: Interactivity Factors for a Distance Education Course," *Journal of Educational Technology & Society*, Vol. 5, No. 1, 2002, pp. 35–47, <http://www.ifets.info/journals/5_1/ehrlch.html>.
9. Suzuki and Yamamoto, op. cit.
10. Razeq, Frasson, and Kaltenbach, op. cit.

Sheung-On Choy (sochoy@ouhk.edu.hk) is a lecturer of Applied Computing in the School of Science and Technology at the Open University of Hong Kong. Sin-Chun Ng is an assistant professor, and Yiu-Chung Tsang is a technical assistant, at the same institution.