Agents in m-Learning Systems Based on Intelligent Tutoring

Vlado Glavinic[1], Marko Rosic[2], and Marija Zelic[2]

[1] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000 Zagreb, Croatia
[2] Faculty of Natural Sciences, Mathematics and Kinesiology, University of Split, Teslina 12, HR-21000 Split, Croatia
vlado.glavinic@fer.hr,
{marko.rosic,marija.zelic}@pmfst.hr

Abstract. Intelligent tutoring systems (ITSs) represent a particular kind of e-learning systems, which base their operation on the simulation of a human teacher in the learning and teaching process. With the advent of the mobile computing paradigm, m-learning systems, as the "portable and personal" fashion of e-learning, paved the way to the introduction of mobile intelligent tutoring. Mobile intelligent tutoring systems (MITSs) are targeted to fit into a mobile learner's daily routine without disrupting her/his other activities, but conversely enhancing the efficiency and effectiveness of learning in the context of handheld terminals of restricted capabilities. As in the non-portable ITS counterparts, MITSs' tasks are taken over by agents, making them agent-based systems. In this paper we discuss the mobile intelligent tutoring paradigm, as well as the agent types to be used in the m-learning environment along with the presently affordable agent infrastructure enabling MITS implementation, and corroborate this with the description of a mobile intelligent tutoring model we are developing.

Keywords: agents, intelligent tutoring systems, m-learning, agent-based systems, mobile intelligent tutoring systems.

1 Introduction

Information and communication technology (ICT) has become an integral part of the educational system for either assisting a teacher in realizing a traditional lecture or in completely substituting the teacher. *E-learning* took place between ICT on the one hand and education on the other. The American Society for Training and Development defines e-learning as "instructional content or learning experience delivered or enabled by electronic technology" [1]. A subclass of e-learning – *mobile learning* (*m-learning*) – is nowadays attracting a lot of attention as the next kind of learning, the respective implementation issues however still being the object of more detailed further elaboration. Obviously, the development of this learning paradigm should be considered mostly through developments in a (possibly high-bandwidth) wireless infrastructure represented by technologies such as GSM, UMTS or WiMax. By using the above infrastructure, it is possible to perform networking of mobile devices

C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 578–587, 2007.

having a number of characteristics interesting to the learning process: (i) small size and high portability, (ii) flexibility in supporting a wide range of learning activities and (iii) a relatively cheap cost of technology. In fact, we are now at the break of a new world of ubiquitous computing that will bridge the gap between our largely sedentary lives and new forms of nomadic life. Hence developments in both wireless data communication and mobile devices facilitate the emergence of a trend of nomadic computing, and consequently of a new fashion in exercising the learning and teaching process. Within the educational context these new nomads will be impersonated by students, teachers, employees, businessmen and customers, who will be traversing both digital and physical landscapes to gather resources, enlarge their knowledge and test their skills, conduct business, and will overall experience both state-of-the-art technologies and services geared to new nomadic lifestyles.

Mobile learners usually work on their own, without external expert support, and possess disparate learning backgrounds. With little time on hand for educational activities and no human expert available for immediate help, these learners require the least learning "overhead" possible, in order to be able to efficiently focus on their actual learning tasks. This in turn requires m-learning systems capable to adapt their behavior to learners' individual requirements [2]. Within such a framework the opportunity to use *intelligent tutoring systems* (ITSs) is highly welcome. As it is already known, ITSs are e-learning systems which base their work on the simulation of a human teacher in the learning and teaching process [3]. Hence, *mobile intelligent tutoring systems* (MITSs) are the e-learning systems used in the m-learning context [4], which provide intelligent support to the learners. Such systems are intended to fit in the mobile learner's daily routine without disrupting her/his other activities, she/he being able to learn while on a bus, while waiting for a friend in a restaurant or some other appointment, between lectures and the like. Advantages of using MITSs include adaptation to both learner's knowledge and abilities as in conventional ITSs, as well as to her/his location and needs in that location [5].

Another question to be considered when adopting the m-learning paradigm is the adaptability of data services, which should provide for such issues as the display of high-quality images on PDAs. This obviously calls for a new technology, which will be based upon a *middleware* that will support both development and runtime execution of *distributed agent-based systems* to perform adaptability of data services. This middleware should anyway enable the systems to work both in the wired and wireless environment.

According to the previous exposition, MITSs are agent-based systems. Agents met within MITSs conceptually belong to the personal agents' type. According to Case [6] *personal agents* are agents intended to satisfy users' demands and possessing characteristics of proactiveness and adaptability, with the main purpose to help a user by performing actions on her/his behalf. Whether a personal agent will adequately decide in which moment to perform a certain action depends on the knowledge it possesses of the user. Hence the way a personal agent registers information on its user is of key importance. It initially has no knowledge on both user's needs and habits, but starts with a predefined user's profile created for an average user, and only afterwards acquires this knowledge by using some of the following methods [7], [8]: (i) registering the way a user performs actions, (ii) directly from the user's instructions, (iii) from the user's examples and (iv) from the other personal agents.

The paper is structured as follows. Section 2 contains an overview of intelligent tutoring systems; Section 3 presents agent tasks in mobile intelligent tutoring, an agent-based mobile intelligent tutoring model as well as the respective development infrastructure, while Section 4 concludes the paper.

2 Intelligent Tutoring Systems

Intelligent tutoring systems are a generation of systems intended for support and improvement of the learning and teaching process. Their work is based on simulation of a human teacher in the learning and teaching process. A question should be immediately raised about the meaning of the phrase "simulation of a human teacher in the learning and teaching process" as well as which characteristics of a human teacher a tutoring system should possess to be attributed as *intelligent*. The teaching paradigm of an intelligent tutoring system is based on the following features of a human teacher [3], [9], [10], [11]: (i) possession of domain knowledge, (ii) possession of knowledge on teaching methods, (iii) capability of diagnosing students' knowledge, (iv) capability of comparing the level of students' knowledge with the domain knowledge i.e. determining the level of students' domain knowledge acquisition and (v) capability of adjusting the teaching methods to the student.

A teacher in traditional educational institutions generally does not have the possibility to adjust her/his teaching methods to individual student's needs and mostly adjusts them to a student group. Conversely, ITSs adjust themselves to individual student's needs, and should thus contain knowledge on the subject being taught, on the way of teaching and on techniques of diagnosing students' knowledge. By adjusting itself to individual students' needs, the learning and teaching process is approaching the ideal of assigning each student her/his own teacher (1-1 relationship), resulting in significantly better results with respect to the traditional 1-n educational process. According to Major [12], the adjustment to individual students' needs is performed by an ITS in the following ways: (i) varying the difficulty level of teaching contents presented to a student, (ii) adjusting the presentation manner, (iii) providing help to the student depending on his results and behavior during system use, and (iv) adjusting the processing order of individual teaching units.

The four main interconnected modules met in the architecture of a typical ITS include (i) a domain knowledge module with the domain knowledge base, (ii) a teacher module for guiding the learning and teaching process, (iii) a student module with information specific for each individual student, and (iv) a suitable user interface module supporting interaction among the student, the teacher and the domain knowledge [3], [13], [14].

The *domain knowledge module* contains knowledge created by an expert in the domain. Separating the expert knowledge from the rest of the system is one of the prerequisites of achieving reuse of that knowledge. Reusing such knowledge is an important element that lowers costs of developing an ITS.

The *student module* creates the student's model. The student's model contains data on the student's knowledge, her/his assumed capabilities, foreknowledge, and behavior while learning as well as the teaching methods that suite the student best. Using the student's model is necessary in adjusting the system to individual students' needs.

The *teacher module* represents the control part of the system since it guides the learning and teaching process using information gained from the other modules. This module contains knowledge necessary for making decisions about the teaching strategy: it uses information on the current student's knowledge and, based on comparison with the expert knowledge, performs the necessary corrections of the teaching process. The corrections refer to (i) the method of teaching contents presentation, (ii) the adjustment to the student's characteristics of the level of difficulty of the teaching contents, as well as (iii) recommending subsequent actions to the student.

The *user interface* module contains mechanisms used both by the student and the ITS to intercommunicate, possibly including methods of adjusting the interface to a student's individual needs. As the student experiences the system exactly through this module, its implementation quality influences teaching quality. Recording the student's activity during her/his interaction with the user interface enables inferring conclusions on interaction styles that suite the student best. Based on these data it is possible to adapt the user interface to the student.

3 Mobile Intelligent Tutoring

3.1 Agents in Mobile Intelligent Tutoring

Transfer from classical intelligent tutoring to mobile intelligent one can be achieved by assigning the tasks of classical ITSs to the personal agents. Such a transfer into the m-learning environment is natural because each module of an ITS can be regarded as a personal agent. The tasks which the personal agents can take over are enumerated below, and worked out in some detail in the following:

a. taking over the role of a human teacher,
b. adapting the user interface to individual user's needs,
c. enabling cooperative learning,
d. enabling competition,
e. searching the information space,
f. help in using virtual reality systems,
g. adjusting the system to the device used for accessing the system.

3.1.1 Taking over the Role of a Human Teacher

According to King [15] an agent is a computer substitute of a person or a process that fulfills a specific purpose or activity. This entity is capable of making decisions that are similar to human ones. The agent has enough insight into personality of the person it substitutes so it can make decisions on her/his behalf. Following this definition the agent should be able to adapt itself to different educational situations like a human teacher.

3.1.2 Adapting the User Interface to Individual User's Needs

The present paradigm of using a computer is based on direct manipulation of the interface elements [16] where a user gives instructions by using the graphical user interface. Adaptation of the user interface on the simplest level enables the user to specify different interface parameters based on personal preferences. More complex adaptations call for user interface agents, which are able to learn the user's habits while using the system and subsequently to suggest automatic completion of recognized action sequences.

3.1.3 Enabling Cooperative Learning

Cooperative learning is one of the common learning paradigms present e-/m-learning systems are based upon. The cooperative learning service enables a student-student interaction aimed at solving a mutual problem. With such learning style, established groups of users get tasks to be solved by joint work. Agents within systems based on such a learning paradigm can be used in forming cooperative learning groups. Students' personal profiles provide information necessary to establish groups of users with similar capabilities and interests [17].

3.1.4 Enabling Competition

Competition is an especially important learning component in areas where the student, who has completed the learning process, is exposed to competition, e.g. buying/selling stock shares or carrying out war operations. Competition can be carried out by placing human participants on rival sides. As such type of competition often shows a negative influence on the student's independence of the system access time, agents are usually engaged as rivals to human competitors.

3.1.5 Searching the Information Space

The present information infrastructure enables access to vast volumes of data. M-learning systems can use these data for teaching a distant student. The systems enable access to different knowledge sources via a unique user interface (mostly a Web browser's one). The student having access to different knowledge sources is not spared the problems that users searching the information space generally face nowadays. The user's access to different information sources may be complicated due to distribution of data and their heterogeneousness or information overload.

It is therefore evident that application of personal agents for searching the information space would enable easier performing of complex actions which the user carries out to obtain the desired results. Having insight into the user's personal profile, the personal agent can search the information space instead of the user, selecting data it finds to be of interest to the user.

3.1.6 Help in Using Virtual Reality Systems

Virtual reality (VR) systems use simulations to familiarize students with conditions in the real environment, where they will work after acquiring required knowledge and skills. These systems are especially useful in such areas where learning and training in the real environment is either too expensive or too dangerous. Examples of such areas are flying an aircraft, mastering of surgical skills or attaining knowledge necessary for maintenance, surveillance and intervention within expensive technical systems.

In VR based educational systems, students should be provided with a "helper" that will guide them in using the system. This function can certainly be assigned to an agent, which should follow the student through the VR space and when necessary suggest her/him the appropriate activities. Furthermore the agent will be recording the student's activities in order to enable the analysis of the student's use of the system. In case the student uses a networked VR system, her/his agent should be capable of communicating with other students' agents so as to accomplish possible common tasks.

3.1.7 Adjusting the System to the Device Used for Accessing the System

By using agents it is also possible to achieve adaptation to the device used for accessing an m-learning system's resources. Such an adaptation is necessary since all devices do not have equal capabilities. This primarily means different display capabilities, amount of memory and processing power. The student's personal agent should adapt content distributed to the student taking all this into consideration. It is also necessary to take into consideration that the student will not always access the system by using the same device.

An example of the infrastructure enabling the implementation of the above described model is illustrated in the following.

3.2 Example Agent Infrastructure

A candidate agent infrastructure which should support the development of an m-learning system should obviously comply with quite a number of requirements, but the paramount one is certainly its conformance to some open standard. The open source package JADE (Java Agent DEvelopment Framework) is a middleware for development and runtime execution of distributed agent-based systems [18], which complies with FIPA (The Foundation for Intelligent Physical Agents) standardization.

JADE runtime environment instances are *containers* (since they can contain several agents), and a set of all containers forms a *platform* providing a homogenous layer above a variety of hardware, operating systems, network types etc. This means that the agent platform can be distributed across machines which do not even have to share the same operating system. JADE agents can transparently communicate across containers and even platforms.

The agent platform dynamically evolves with agents appearing and disappearing based on needs and requirements of the application. Each agent is a peer capable of initiating communication, providing services, searching and discovering other agents, entering and leaving the platform anytime.

In order to properly address the resources of mobile devices, which are limited in terms of memory and processing power, as well as the characteristics of wireless links

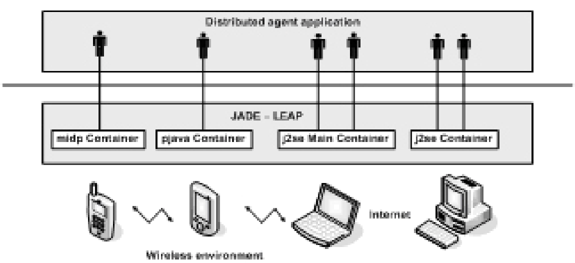

Fig. 1. JADE-LEAP runtime environment

such as high latency, low bandwidth and intermittent connectivity, the LEAP (Lightweight Extensible Agent Platform) add-on was additionally created. It replaces some parts of the JADE kernel, forming a modified runtime environment called JADE-LEAP ("JADE powered by LEAP") which enables deployment of agents on lightweight devices. Figure 1 shows an example of the JADE-LEAP runtime environment.

3.3 Agent-Based Mobile Intelligent Tutoring Model

Most ITSs are designed for use on standard desktop computers or laptops and are not customized for eminently mobile devices such as mobile phones or PDAs. In developing a framework for such systems to fit into the mobile environment with maximum efficiency and minimum effort the main intent is not to reinvent the ITS but rather to insure the integration of the existing "wireline" and the new wireless (GSM, UMTS, WLAN, various satellite systems) environment. However, even though an MITS might seem similar to a conventional ITS, implementation requirements are much more complex. The mobile computing environment is namely very different with respect to the present traditional (non-mobile, i.e. wireline) distributed systems milieu; bandwidth, delay, error rate, interference and the like, may change dramatically as a user moves from one location to another, thus changing the computing environment. Also, the variety of mobile workstations, handheld devices and smart phones is increasing rapidly as well as the range of their capabilities, spanning from low-performance mobile phones to high-performance laptops.

(M)ITS development represents an expensive process, which envisions the engagement of experts in computer science, teaching methodology as well as domain knowledge. Decreasing the respective costs can be achieved by insuring (M)ITS interoperability. An agent-based mobile intelligent tutoring model that enables interoperability of different (M)ITSs by sharing knowledge in the common information space is presented as follows.

Mobile intelligent tutoring is in this model performed by using knowledge in a common information space. The common information space is shared by classical ITSs as well as users' personal agents that take over the ITS's tasks (see Figure 2).

First of all, let us note that sharing knowledge in the common information space may be complicated due to the fact that different systems differently display and store knowledge. Furthermore, different systems use (i) different terminology for the same

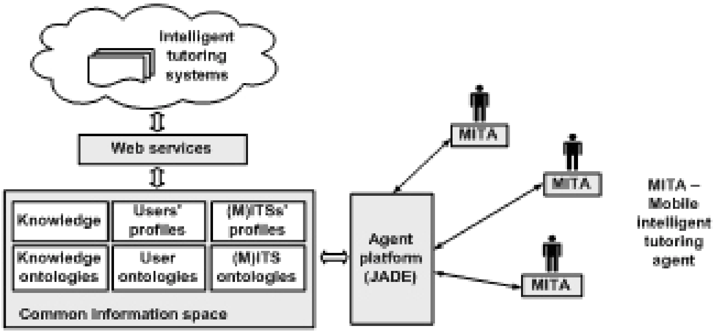

Fig. 2. Agent-based mobile intelligent tutoring model

domain knowledge elements, (ii) the same terminology for different elements, (iii) different ways of connecting these elements as well as (iv) different inference rules on the elements. The common solution to this problem is the use of *ontologies*, which define domain concepts and their attributes, relationships between these concepts as well as inference rules within the domain [19].

The common information space should include three types of ontologies, which are to enable publishing along with sharing knowledge, users' profiles as well as data within (M)ITSs.: (i) knowledge ontologies, (ii) user ontologies and (iii) (M)ITS ontologies. (M)ITSs access these ontologies in the common information space by using adequate Web services (e.g. [20]).

A *mobile intelligent tutoring agent* (MITA) takes over the tasks of classical ITSs in the new environment, see Figure 3. Therefore it should contain knowledge on the subject being taught, on the person being taught and on the way of teaching. However, MITA should additionally contain knowledge on capabilities of available terminal devices in order to adjust domain knowledge presentation styles to capabilities of the particular device used for accessing the system. Furthermore the agent should contain knowledge on available networks and their characteristics so it can adjust itself to possible changes in the environment. MITA enables communication among currently active users as well as connecting users with similar interests. This functionality is supported by sharing users' personal profiles within the common information space. MITA also enables testing of mobile users. While performing these tasks the agent is adjusting itself to users' capabilities, terminal devices and networks.

Such an approach to MITS development has the following advantages compared to MITS development from scratch: (i) teaching contents are being used, which pertain to classical ITSs, thus lowering costs of developing MITSs, (ii) by using MITSs classical ITS users can actually learn anyplace and anytime, and (iii) the group of potential intelligent tutoring users is expanding.

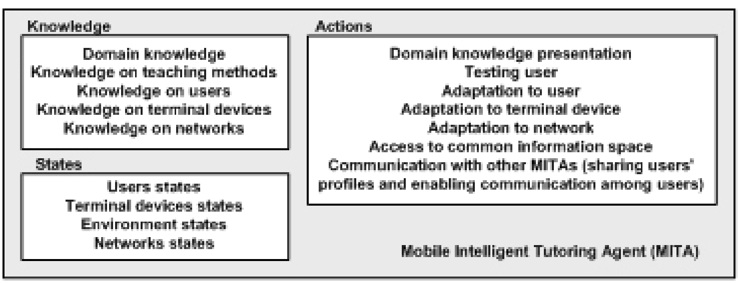

Fig. 3. Mobile intelligent tutoring agent structure

4 Conclusion

M-learning is getting a lot of attention these days as the next kind of learning. We live in an information era where people constantly demand fast and reliable ways of accessing any kind of information, and expect to get this information anytime and anywhere, by using their computers, laptops, mobile phones or whatever technology they have at hand. However, the m-learning environment represents a challenge for application developers. Firstly, a user can have more networks available at a time, each one with its own characteristics regarding bandwidth, delay, error rate, interference and so on. Secondly, the user can access information by using various mobile or stationary devices, with different processing power, amount of memory and display qualities. A promising approach to solving the above mentioned problems is the use of agent-based technology. Agents' characteristics such as autonomy, learning capabilities, proactiveness and social skills enable adaptation of m-learning systems to the mobile environment.

This paper presents an agent-based mobile intelligent tutoring model that is being currently developed. The proposed model simplifies development of MITSs by reusing knowledge within classical ITSs. The tasks of classical ITSs are taken over by agents, taking into consideration the specifics of the m-learning environment. The agents perform adaptation of the system to the network used for accessing the system as well as the terminal device.

Acknowledgements

This paper describes the results of research being carried out within the project 036-0361994-1995 *Universal Middleware Platform for e-Learning Systems* funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. American Society for Training and Development: A Vision of E-Learning for America’s Workforce. Report of the Commission on Technology and Adult Learning (2001)
2. Kinshuk: Adaptive mobile learning technologies. GlobalEducator.com (2003)
3. Stankov, S.: Isomorphic Model of the System as the Basis of Teaching Control Principles in the Intelligent Tutoring System. PhD Dissertation. Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture University of Split, Split (in Croatian) (1997)
4. Goh, T.T., Kinshuk: A discussion on mobile agent based mobile web-based ITS. In: Proc. Int’l Conf. on Computers in Education. Los Alamitos pp. 1514–1515 (2002)
5. Cui, Y., Bull, S.: Context and learner modeling for the mobile foreign language learner. System 33, 353–367 (2005)
6. Case, S., Azarmi, N., Thint, M., Ohtani, T.: Enhancing E-Communities with Agent-Based Systems. Computer, pp. 64–69 (2001)
7. Maes, P.: Agents that Reduce Work and Information Overload. Comm. ACM 37, 30–40 (1994)
8. Kuflik, T., Shoval, P.: User profile generation for intelligent information agents. In: Proc. 2nd Int’l Workshop on Agent-Oriented Information Systems. Stockholm, pp. 63–72 (2000)
9. Patel, A., Kinshuk, R.D., Oppermann, R.: Intelligent Tutoring Systems: Confluence of Information Science and Cognitive Science. In: Proc. Informing Science Conference. Krakow (2001) pp. 392–397 (2001)

10. Castro, E., Cuadra, D., Martinez, P., Iglesias, A.: Integrating Intelligent Methodological and Tutoring assistance in a CASE platform: The PANDORA experience. In: Proc. Informing Science & IT Education Conference. Cork (2002)
11. Ainsworth, S., Clarke, S.E., Gaizauskas, D.: Using edit distance algorithms to compare alternative approaches to ITS authoring. In: Cerri, A., Gouarderes, G., Paraguacu, F. (eds.) Intelligent Tutoring Systems, pp. 873–882. Springer, Heidelberg (2002)
12. Major, N.: How Generic can Authoring Shells Become? In: Proc. AI-ED-95 Workshop on Authoring Shells for Intelligent Tutoring Systems. Washington, pp. 75–82 (1995)
13. Burns, H.L., Capps, C.G.: Foundations of Intelligent Tutoring Systems: An Introduction. In: Polson, M.C., Richardson, J.J. (eds.) Foundations of Intelligent Tutoring Systems, pp. 1–18. Lawrence Erlbaum Associates Publishers, London (1988)
14. Woolf, B.P.: AI in Education. In: Shapiro, I., Stuart, C. (eds.) Artificial Intelligence - Encyclopedias, pp. 434–444. John Willy & Sons Inc. New York (1992)
15. King, J.A.: Intelligent Agents: Bringing Good Things to Life. AI Expert, pp. 17–19 (1995)
16. Schneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Publishing Company, Reading (1987)
17. Jaques, P., Andrade, A., Jung, J., Bordini, R., Vicari, R.: Using pedagogical agents to support collaborative distance learning. In: Proc. Int'l Conf. on Computer Support for Collaborative Learning. Boulder, pp. 546–547 (2002)
18. Bellifemine, F., Poggi, A., Rimassa, G.: JADE: a FIPA 2000, compliant agent development environment. In: Proc. 5th Int'l Conf. on Autonomous Agents. Montreal pp. 216–217 (2001)
19. Poli, R.: Ontology for Knowledge Organization. In: Green, R. (eds.) Knowledge organization and change. Indeks, Frankfurt, pp. 313–319 (1996)
20. Erl, T.: Service-Oriented Architecture. A Field Guide to Integrating XML and Web Services. Prentice-Hall PTR, Upper Saddle River, NJ (2004)