PERGAMON

Computers & Education 40 (2003) 255–269

COMPUTERS & EDUCATION

www.elsevier.com/locate/compedu

Redefining the learning companion: the past, present, and future of educational agents

Chih-Yueh Chou[a,*], Tak-Wai Chan[b], Chi-Jen Lin[c]

[a]Learning Technology Research Center, National Central University, Chungli 32054, Taiwan
[b]Graduate Institute of Network Learning Technology, National Central University, Chungli, Taiwan
[c]Department of Computer Science and Information Engineering, National Central University, Chungli, Taiwan

Received 17 June 2002; accepted 18 September 2002

Abstract

The development of intelligent tutoring systems has long been the focus of applying artificial intelligence and cognitive science in education. A new breed of intelligent learning environments called *learning companion systems* was developed over a decade ago. In contrast to an intelligent tutoring system, in which a computer mimics an intelligent tutor, the learning companion system assumes two roles, one as an intelligent tutor and another as a learning companion. Motivated by recent interest in agent research and other technologies, this learning companion field has received increasing attention. This study addresses issues that arise from different perspectives on this research effort. With a view to future networked learning environments, the learning companion is redefined for application to a wide spectrum of educational agent research. Accordingly, several subjects that relate to educational agents, and hence learning companions, are identified.

Keywords: Intelligent tutoring systems; Collaborative learning; Simulations; Computer-mediated communication; Learning communities

* Corresponding author. Tel.: +886-3-4227151; fax: +886-3-4261931.
E-mail addresses: yueh@src.ncu.edu.tw (C.-Yueh Chou), chan@src.ncu.edu.tw (T.-Wai Chan), zen@src.ncu.edu.tw (C.-Jen Lin).

0360-1315/03/$ - see front matter
PII: S0360-1315(02)00130-6

1. Background

Despite the recent fall in interest in intelligent agent[1] research, the vision of employing a computer as an agent, called an *intelligent tutor*, to assist a student to learn has been continuously pursued since Carbonell (1970) conducted his pioneering work on simulating a Socratic tutor, using a semantic network technique, in the early 1970s. This kind of learning system, in which a computer simulates a tutor, is known as an intelligent tutoring system (ITS), and has been investigated by a number of researchers (Sleeman & Brown, 1982; Wenger, 1987). A rationale for this effort is that one-on-one human tutoring is seen as an ideal learning scenario that supports individual learning. A human tutor, although not as patient as a computer, offers immediate feedback that is tailored to the student's particular needs. In fact, private tutoring has been shown to be four times as effective as classroom teaching, according to a study conducted by Bloom (1984).

Unlike a computer treated as a provider of content (mostly text at that time) to students, or as a learning tool or toy, a computer that models a tutor must mimic tutoring interactions such as looking over a student's shoulder, and offering just-in-time and adaptive feedback. These interactions are sometimes mixed initiatives, posing a challenge in human–computer interactions in which the student's input was mainly in the form of line editing at this time. Thus, an intelligent tutor must understand the student and be equipped with a strategy for interacting with the student. When this field started to grow in the 1980s, various artificial intelligence techniques were developed and people had high expectations of developing such an intelligent tutor.

ITSs are generally composed of four modules—domain expert module, student model, pedagogical module, and interface (Wenger, 1987). Firstly, ITSs usually possess expert domain knowledge that can be presented to the student and used to evaluate the student's performance. The expert domain knowledge also explicitly represents the goal of teaching. Knowledge is normally represented in ITSs as a set of facts or rules. Some ITSs even have a complex problem solver, which can solve a specific range of problems. Secondly, ITSs must "model" the student, detecting the student's beliefs and misconceptions directly from the student's answers to diagnostic tests or through tracing the student's actions. Such modeling is essential to providing adaptive feedback. Various approaches for modeling students have been proposed (Greer & McCalla, 1994, Mizoguchi, 1995). One of the well-known approaches for modeling a student is the overlay model, which treats and presents the knowledge of a student as a subset of that of an expert (Carr & Goldstein, 1977). Many ITSs also collect a set of bugs and misconceptions to diagnose the student (Brown & Burton, 1978). Thirdly, ITSs include pedagogical strategies and instructions to guide, coach, or tutor the student. The pedagogical strategies determine when and how to instruct the student. The instruction may be to directly point out an error and give the correct answer or indirectly to provide hints to lead the student to find out the correct answer. Fourthly, ITSs contain interfaces with which to communicate with the student, including receiving input and presenting information. The interface may be composed of buttons, menus, text,

[1] "Agent" has different meanings and definitions in different domains and communities (Les, Cumming, & Finch, 1999). In this study, agent refers to a computer-simulated character, which presents users with human-like characteristics, such as domain competence, emotions, and other personal characteristics. The characteristics can be expressed or displayed in text, graph, icons, voice, animation, multimedia, or virtual reality.

graph, voice, animation, multimedia, virtual reality, or other advanced techniques. An interface usually processes natural language to facilitate communication between ITSs and the student.

The effort of creating such an intelligent tutor in a computer continues today. However, in the mid 1980s, some researchers challenged the vision of modeling an intelligent tutor, for reasons concerning the feasibility of implementing an intelligent tutor and the role it might take (Gilmore & Self, 1988; Self, 1988). Another path was opened up, substantially broadening the view of the intelligent tutor, with respect to the roles, multiplicity, distribution, and the social context of educational agents. The paradigm of this field shifted, as described below.

2. Learning companion and educational agent

2.1. Learning companion—its origin and extension

The development of learning companion systems contributes to the shift mentioned above. The origin of learning companions is reviewed and related issues are identified below. In the environment of first learning companion system (LCS), three characters are involved—namely, the human student, the computer-simulated learning companion, and the computer-simulated teacher (Chan & Baskin, 1988, 1990). As implied by its name, a learning companion acts as a companion to the student during learning activities. To this end, the learning companion learns from the teacher, and collaborates with or competes against the human student. Chan and Baskin presented three learning protocols between the student and the learning companion in an LCS—working independently and then comparing work (competition), working collaboratively with one's working and one's watching (suggestion), and working collaboratively and sharing responsibility (collaboration). Moreover, Chan and Baskin proposed an approach in which the student can teach the learning companion and thus ''*learn how to learn by teaching*''. This last approach aims to encourage the student to provide the learning companion with knowledge and examples, to observe how the learning companion solves the problems, and to explain why the solution of the learning companion is correct or incorrect. The approach is called ''learning by teaching'' and has been examined further by a number of researchers (Palthepu, Greer, & McCalla, 1991; Nichols, 1994; Ur & VanLehn, 1995; Ramirez Uresti, 2000). The approach is also useful for training human tutors.

During the last decade, several LCS-related studies have considered different protocols for various social learning activities. A social learning activity enables a student to interact with other people, such as a teacher or a fellow student, during the learning activity. These people are either computer-simulated or real human beings. Furthermore, a social learning environment is an environment promoting students to engage in social learning activities. ''Distributed West'' is a distributed LCS, which consists of two connected computers, with which students can learn collaboratively and/or competitively at different locations (Chan, Chung, Ho, Hou, & Lin, 1992). Instead of a real student, the system also provides a learning companion as the opponent of the student. Seven hundred and sixty-eight possible distributed learning models have been enumerated, including different numbers and roles of agents. The ''People Power'' system contains a micro-world and a computer simulated co-learner (Dillenbourg & Self, 1992). The student can create an electoral system and simulate elections in the micro-world. Meanwhile, the student collaborates with a co-leaner to learn from their experiences in the micro-world. The co-learner questions the student to seek answers for itself, not to check whether the student knows the answers. Collaboration between the student and co-learner is modeled as ''socially distributed cognition''. In the RTS system, a learning companion is designed to support the learning activity of reciprocal tutoring (Chan & Chou, 1995, 1997). Reciprocal tutoring is a collaborative learning activity, in which two participants play the role of the tutor and tutee, in turn (Palincsar, 1984). PALs is another LCS that supports reciprocal tutoring. Results show that PAL tutorials are nearly as effective as individual tutoring by experienced tutors, but require much less instructor time (Scott & Reif, 1999).

Other researchers have focused on different levels of knowledge, and different numbers and roles of learning companions. Hietala and Niemirepo (1998) designed an LCS, named the EduAgents system, which provides multiple learning companions, two strong and two weak. The strong learning companions have expert domain knowledge and do not make mistakes. The weak learning companions have rather poor knowledge and often make mistakes. Researchers found that different students prefer different learning companions. Introverts and more capable students prefer the strong companions. Extraverts and less capable students prefer the weak companions. These researchers have suggested that ''*a group of heterogeneous companion agents at the learner's disposal will increase his/her motivation to collaborate with agents*''. Aimeur and Frasson (1996) proposed a simulated student that plays the role of a troublemaker and sometimes provides the student with erroneous opinions. The troublemaker is used to trigger a cognitive dissonance—a difference between the student's knowledge and the troublemaker's suggestions—to motivate the student to seek new information to reduce this dissonance (Aimeur, Dufort, Leibu, & Frasson, 1997). Goodman and his colleagues designed an LCS in which the learning companion encourages the student to reflect on, and articulate his/her past actions (Goodman, Sollor, Linton, & Gaimari, 1998). Chang and his colleagues proposed four animal companions that played different roles—collaborator, troublemaker, peer tutor, and tutee (Chang, Wang, Hsu, & Chan, 1999).

With a view to future networked learning environments, the learning companion is redefined to be applied in a wide spectrum of educational agent research. Some subjects concerning educational agents and hence learning companions are identified. In an extensive definition, **a learning companion is a computer-simulated character, which has human-like characteristics and plays a non-authoritative role in a social learning environment.** Moreover, a learning companion is a kind of educational agent. The relationship between the educational agent and the learning companion is discussed below.

2.2. Educational agent and multiplicity of roles

Educational agents are pieces of educational software with human characteristics that facilitate social learning. The characteristics of the agent can be expressed to students in text, graph, icons, voice, animation, multimedia, or virtual reality. The social context catalyzes the cultivation of, and motivation for knowledge. Systems that support social learning by including learning environments that involve multiple people, are emerging for operation on a single computer or across connected machines (Chan, 1995). These people are either computer-simulated (educational agents) or real humans, taking various roles according to different protocols of social learning activity.

Educational agents can be classified according to their role and function (Table 1). Pedagogical agents are designed to be involved in social learning activities for a specific pedagogical purpose. Two typical applications of pedagogical agents are ITSs and LCSs. ITSs employ one-to-one tutoring activities, thus pedagogical agents in ITSs are authoritative teachers. ITSs provide a simulated teacher, who is a domain expert, and tutors, coaches, or guides the student. The simulated teacher can provide adaptive instruction appropriate to the individual student, according to the learning process of the student (Sleeman & Brown, 1982; Wenger 1987).

‘‘Learning companion’’, ‘‘co-learner’’, ‘‘simulated student’’, and ‘‘artificial student’’ are names used by different researchers to denote a non-authoritative pedagogical agent, simulated by a computer. The name ‘‘learning companion’’ is used herein. To the student, the learning companion is not a domain expert and may make mistakes. Collaborative or competitive learning activities are adopted in LCSs as alternatives to one-to-one tutoring. Moreover, some LCSs contain both a simulated teacher and learning companion(s). Therefore, LCSs may include many various learning activities. The learning companion can act as a peer tutor (Chan & Chou, 1997), tutee or teachable student (Biswas, Katzlberger, Brandford, Schwartz, & TAG-V, 2001; Brophy, Biswas, Katzlberger, Bransford, & Schwartz, 1999; Nichols, 1994; Ramirez Uresti, 2000; Scott & Reif, 1999; Ur & Vanlehn, 1995), collaborator (Chan & Baskin, 1990; Dillenbour & Self 1992; Goodman et al., 1998; Hietala & Niemirepo, 1998), competitor (Chan & Baskin, 1990; Chan et al., 1992), troublemaker (Aimeur & Frasson, 1996), critic, or clone (Chang et al., 1999). Therefore, as extensively defined, a learning companion is a computer-simulated character, which has human characteristics and plays a non-authoritative role in a social learning environment.

A personal assistant provides each user (a teacher or a student) with individual and useful information that pertains to learning activities, but does not become directly involved in those learning activities. A teacher’s assistant can provide the teacher with the learning portfolio of the students, including learning performance, misunderstandings, levels of effort, and motivations. The portfolio can help the teacher to understand the students and respond appropriately to them. The students’ assistant can help students to collect information to perform learning activities, such as searching for and arranging instructors, tutors, or classmates. For example, in the PhelpS system, a personal assistant helps a user to record the subtasks that have been completed and to

Table 1
Classification of educational agents

Educational agent		
Pedagogical agent		Personal assistant
Authoritative teacher	Learning companion or co-learner or simulated student	Teacher’s assistant Student’s assistant
Tutor Coach Guide	Competitor Collaborator Tutee Peer tutor Troublemaker Critic Clone	

obtain help from peers when required (Greer, McCalla, Collin, Kumar, Meagher, & Vassileva, 1998; McCalla et al., 1997). The rest of this paper focuses primarily on the pedagogical agent.

3. Design issues of the educational agent: computational support for social learning

Bernard and Sandberg (1993) proposed that a student in a learning environment should be placed within the context of surrounding entities that facilitate the student's access to learning resources, learning by various means, and participating in a range of learning activities. This idea has been extended to a situation in which educational agents are presented in a social learning environment. A student has access to many learning resources, which can be classified into three categories—content, community, and computational support (Chan, Hue, Chou, & Tzeng, 2001; see Fig. 1). The content includes the student's learning materials, such as books, libraries, museums, and databases. The student can participate in communities and communicate knowledge with fellow students, teachers, volunteers, and parents. Additionally, the student is empowered by computational support, such as calculators, note-pads, and different kinds of computer software. Accordingly, **an educational agent is a kind of computational support, which enriches the social context in a social learning environment either by providing virtual participants to enhance the member multiplicity of communities or by supporting facilities to foster communication among real participants.**

The positive impact of research on educational agents lies in its ability to strengthen the social learning environment. The research issues can be divided into design issues and implementation issues. Design issues concern two main elements of defining a particular social learning model—the structure of the model and the protocols of learning activities. Structure relates to the members of the learning environment, including the numbers, roles, and characteristics of the educational agents. The protocol is a set of rules that governs the learning activity, including communication among participants. The existing social learning model or theory, which is successful for real participants, is usually adopted in designing educational agents. A widely applied theory is Vygotsky's hypothesis, which regards cognitive development as the gradual internalization and personalization of what was originally a social activity (Vygotsky, 1978). Vygotsky also

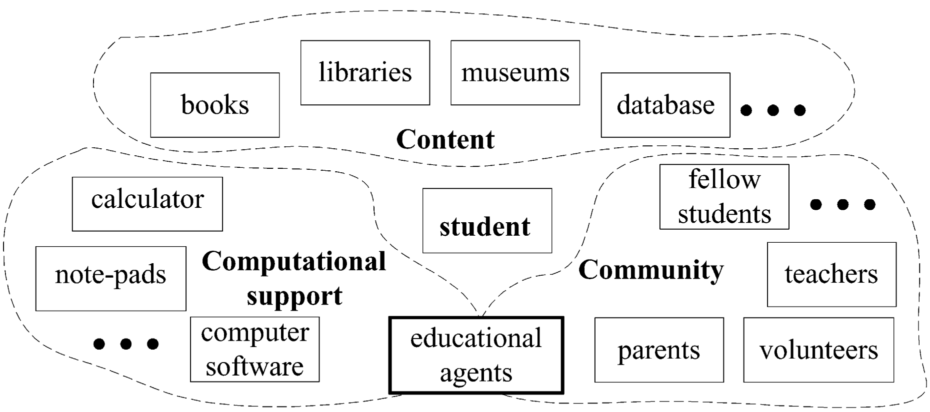

Fig. 1. Role of educational agents in a social learning environment.

proposed a hypothetical distance, ''zone of proximal development'', between development through solving problems independently and potential development through solving problems under adult guidance or in collaboration with more capable peers. Other social learning theories have been recently explored and applied to educational agents, such as responsibility sharing (Chan & Baskin, 1990), socially distributed cognition (Dillenbourg & Self, 1992), reflection and articulation (Goodman et al., 1998), reciprocal tutoring (Chan & Chou, 1997; Scott & Reif, 1999), and learning by teaching (Chan & Baskin, 1990; Nichols, 1994; Ramirez Uresti, 2000). Through different communication media, face-to-face communication among real participants and communication between a student and an educational agent (computer software) have different benefits and constraints (Collins, Neville, & Bielaczyc, 2000). Face-to-face communication is easy to produce, has a high bandwidth and is highly interactive, but cannot be reproduced, distributed, or modified. Meanwhile, the educational agent, as a kind of computer software, is easily reproduced, distributed, and modified, but is difficultly produced. It allows computer interaction, and variable bandwidth of communication. Thus, educational agents and real participants provide a student with alternative social contexts to foster learning. Various combinations of educational agents and real participants enrich a social learning environment (Chan, 1995). Furthermore, an educational agent can not only directly impose its perspectives on the student, but also stimulate learning and collaboration among real participants (Greer et al., 1998; McCalla et al., 1997; Thibodeau, Belanger, & Frasson, 2000).

Educational agents with controllable characteristics facilitate particular learning models or learning environments. For example, Aimeur and Frasson proposed a strategy of learning by disturbing, in which an educational agent deliberately gives misleading advice at times (Aimeur & Frasson, 1996; Aimeur et al., 1997). If the system detects that the student's self-confidence is high, the agent provides an erroneous suggestion and waits for the student to explain his or her decision or provide a correct solution. Otherwise, the agent provides correct suggestions to reinforce the student's beliefs. The strategy ''*strengthens the attention of the student, increases the perception of details and provides the means for arguing and memorizing*'' (Aimeur & Frasson, 1996). Chang and his colleagues proposed a learning environment in which a student is surrounded by four kinds of companions—collaborator, peer tutor, troublemaker, and tutee (Chang et al., 1999). These companions have different levels of knowledge, adjusted according to the knowledge of the student. Thus, during learning, the knowledge levels of these companions are adapted to fit the student's expectations of the companion and to support the planned pedagogical strategy.

4. Implementation issues of the educational agent

4.1. Realization of a social learning model using particular technologies

Implementation involves realizing a social learning model using particular technologies. An intelligent, individual, and adaptive environment is sought by using artificial intelligence technologies, such as system architecture, data and knowledge representation, and communication mechanisms. Many studies have addressed issues that pertain to implementing authoritative pedagogical agents, as in building an ITS (Sleeman & Brown, 1982; Wenger, 1987). The crucial issues related to ITSs concern the intelligent communication of knowledge, while knowledge communication is defined as ''*the ability to cause and/or support the acquisition of one's knowledge by someone else, via a restricted set of communication operations*'' (Wenger, 1987). These studies deal with four elements of knowledge communication—the object of communication, the recipient of communication, the skill of communication, and the form of communication. These elements are usually governed by four main ITS components—domain module, student model, tutoring module, and interface, respectively (Self, 1974; Sleeman & Brown, 1982).

Constructing learning companions and personal assistants often involves or modifies techniques developed for ITSs while maintaining a focus on communicating knowledge. However, the communication skills of LCSs differ from those of ITSs. A personal assistant prefers to promote communication among participants by understanding each participant and matching appropriate community, rather than directly communicates with participants. The knowledge and capacity for learning of the learning companion can be implemented by a simulation or machine-learning approach (Chan & Baskin, 1990). Most LCSs adopt the simulation approach, which selectively uses domain knowledge. Such LCS can be implemented by modifying the ITSs. However, educational agents can also be implemented by integrating machine-learning and knowledge acquisition (Tecuci & Keeling, 1999).

Several researchers have explored implementation techniques to better realize a social learning model. Some examples are described below. Some researchers have proposed techniques through which educational agents can exhibit emotive behavior and expressions, such as happiness, sorrow, pride, enjoyment and others (Abou-jaoude & Frasson, 1999; Lester, Towns, & FitzGerald, 1999). The emotions enhance believability of educational agents and increase the bandwidth of communication between educational agents and the student. Several implementation techniques have been developed to provide multiple educational agents in a single system. For example, Wang and Chan (2000) designed an agent-oriented programming language, CAROL5, to implement social learning systems for a complete course. CAROL5 provides a hierarchical knowledge-sharing mechanism for building a curriculum-tree structure to organize learning activities. After designing the curriculum-tree, CAROL5 simultaneously supports prototype-based programming, rule-based programming, and event-driven programming to create multiple educational agents to meet the needs of learning activities in the curriculum-tree. Frasson et al. suggested an actor-based architecture to support multiple agents with different pedagogical roles (Frasson, Mengelle, Aimeur, & Gouarderes, 1996). The actor is a specific intelligent agent, which is reactive, instructable, adaptive, and cognitive. Furthermore, the actor can dynamically interact according to various strategies and, in so doing, improve itself.

Several investigators have suggested the multiple ''agent'' implementation approach to deal with the complicated tasks and various perspectives of educational agents (Canut, Gouarderes, & Sanchis, 1999; Leman, Marcenac, & Giroux, 1996; Vassileva, Greer, McCalla, & Deters, 1999). Unlike the agents discussed in this study, these ''agents'' are pieces of software, which autonomously deal with some tasks, interact with other software ''agents'', and hide inside the system. The approach divides the task of implementing educational agents into several subtasks and handles these subtasks by employing several software ''agents''. For example, the mechanism for building a student model can be implemented by constructing spy ''agents'' and task ''agents'' (Leman, Giroux, & Marcenac, 1995). The spy ''agents'' observe and identify the student's actions; the task ''agents'' analyze the action and construct the model.

4.2. Architectural view: example of architecture for various pedagogical agents

Architecture for implementing various pedagogical agents using the simulation approach is presented here to support further discussion of some implementation issues concerning pedagogical agents, addressed from an architectural view. The architecture, named GCM, is an extension of typical ITS architecture (Chou, Chan, & Lin, 2002). GCM is designed for constructing LCSs. However, this architecture also provides a means of constructing various pedagogical agents and comparing ITSs with LCSs (Fig. 2). The GCM architecture consists of five main components—pedagogical module, domain module, student model, learning companion pattern, and interface. The pedagogical and domain modules are substantial generalizations of the tutoring and expert modules in ITSs, respectively. Each learning companion is designed to support a specific pedagogical strategy. The pedagogical module governs the communication between the learning companion and the student, following the guidelines of the pedagogical strategy. Therefore, the pedagogical module has a function similar to that of the tutoring module in ITSs; that is, it simulates the pedagogical behavior of the agent. The difference is that the agent's role in ITSs is that of a tutor, while in LCSs, it is as a learning companion, such as a peer tutor, a peer tutee, a collaborator, a competitor, a critic, or a troublemaker. The expert module in a typical ITS, which simulates a domain expert, theoretically knows all domain knowledge, potential mistakes, and problem-solving processes. The simulation can be modified to enable a domain module in GCM to manage both the knowledge and gaps in knowledge of the learning companion, and to simulate its mistakes and method of learning.

The content of the student model in GCM differs from that of the student model in ITSs because the former model is constructed from a different perspective—that of a learning companion rather than that of a tutor. Notably, in typical ITSs, the student model is the best representation of the student that is available in the system. This happens to be the same model that the system requires if it is acting as a tutor. In contrast, a learning companion might think that the student does not understand a concept although the system has deduced that the student does understand it: this situation may occur if the learning companion is a troublemaker. The learning companion pattern stores data concerning the characteristics of the learning companion. The data supports the domain and pedagogical modules to simulate the level of knowledge and the behavior of the learning companion. Since GCM was originally designed for LCSs, GCM uses the name of the learning companion pattern to denote the component that stores the characteristics of

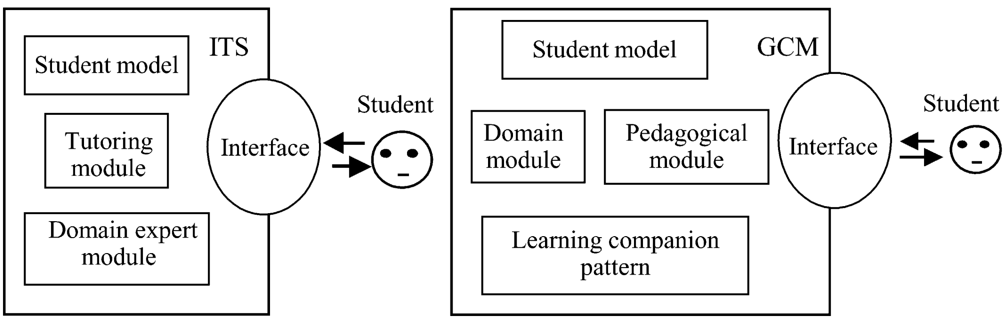

Fig. 2. Comparison of typical ITS and GCM architecture.

the educational agent. The component is generally considered to be an agent model (Yin, El-Nasr, Yang, & Yen, 1998). The function and implementation of the interface component in GCM is similar to that of typical ITSs. A more detailed discussion of the interface is beyond the scope of this paper. Table 2 compares the components of typical ITS architecture with those of GCM.

GCM architecture is a superset of typical ITS architecture. Therefore, GCM is useful for building pedagogical agents as well as learning companions. GCM works like typical ITS architecture when the characteristics and role of the learning companion are set to those of an expert tutor. GCM architecture separates characteristics from simulation mechanisms. The mechanisms are performed according to the characteristics. Therefore, a system can simulate many learning companions with various characteristics by manipulating multiple learning companion patterns. The system can also change the characteristics of a learning companion during learning.

5. Difficulties in development, advances in technology and future directions

Although many studies have addressed educational agents, most involve model-based systems, built for demonstration purposes and used over a limited period. One bottleneck of educational agent development is that implementing a social learning system that includes educational agents is expensive and difficult (Mizoguchi & Bourdeau, 2000; Murray, 1999). The bottleneck arises from the multiplicity and complexity of the educational agents. Many researchers have developed shells and authoring tools to support the implementation of the system and thus release the bottleneck (Murray, 1999). A shell is a generalized framework, including software architecture and code libraries, that enables programmers to increase the efficiency of system implementation. The

Table 2
Comparison of components of typical ITS and GCM architectures

Components	Similarity	Difference
Student model (ITS)	Helps to understand the state of the student	Built from the viewpoint of a tutor
Student model (GCM)		Built from the viewpoint of a tutor or a companion
Tutoring module (ITS)	Determines the pedagogical behavior of the agent	The agent acts as a tutor
Pedagogical module (GCM)		The agent may act as a competitor, peer tutor, tutee, critic, troublemaker, or collaborator
Domain expert module (ITS)	Determines the domain competence of the agent	The tutor has competence of an expert
Domain module (GCM)		The companion may have the competence of an excellent student, an average student, or a novice
Learning companion pattern (GCM)		Stores characteristics of the companions

authoring tool is a shell with a user interface that enables non-programmers, such as domain experts and teachers, to formalize their knowledge. In brief, shells and authoring tools reduce the effort and skill required to build educational agents.

The bottleneck can also be overcome by taking advantage of human intelligence. Traditionally, a system is designed to use machine intelligence to help students. However, a system could provide students with help from peers, rather than computers (Chan & Chou, 1997; Greer et al., 1998). Such a system, however, may perform less well with respect to its knowledge base, reasoning, and pedagogical dialogue. However, machine intelligence is required to support human collaboration. For example, a student model could be used to match peers or students appropriately (Greer et al., 1998). An automatic scaffolding tool supports the collaboration of students (Chan & Chou, 1997). Furthermore, the system may collect and reuse instruction portfolios of these peers or teachers, integrating human and machine intelligence (Ackerman, 1998; Heinrich & Maurer, 2000; Burke, Hammond, Kulyukin, Lytinen, Tomuro, & Schoenberg, 1997). The idea is to create an automatic FAQ system. Whenever a teacher answers a student's question, the system records the student's question and the teacher's answer in a database. When another student has a question, the system searches the database for the question and relevant answer. If the student's question matches a previous question, the system provides the answer; otherwise, the system invites a teacher to answer the question and records the answer. Human intelligence is also drawn upon to enable a student, a peer, or a teacher to model the student to supplement the student model created by the system (Bull, 1998; Bull, Brna, Critchley, Davie, & Holzherr, 1999). The collaborative student modeling process not only improves the accuracy of the student model, but also promotes student reflection and peer interaction. In summary, machine intelligence and human intelligence should complement each other in social learning systems.

Another bottleneck is that the work involved in implementing the systems is rarely shareable or reusable (Mizoguchi & Bourdeau, 2000). Authoring tools and shells support the sharing and reuse of such work. However, sharing the work among different authoring tools and shells is difficult because no relevant standard exists. Mizoguchi & Bourdeau (2000) proposed that ontology-based architecture, with appropriate ontologies, could overcome the bottleneck. According to their definition, "*An ontology consists of task ontology, which specifies the problem solving architecture of knowledge-based systems, and domain ontology, which specifies the domain knowledge*", ontology allows computers to meditate the sharing of human knowledge and extend the sharing from humans to computers. Recently, standardizing the development of e-learning has received considerable attention, such as by SCORM (sharable content object reference model, http://www.adlnet.org/) and ULF (universal learning format, http://www.saba.com/standards/ulf/). SCORM is intended to be durable, interoperable, accessible, and reusable. ULF is an XML-based format for exchanging various types of e-learning data. These standardizations provide an approach to sharing the work of system implementation.

Advances in technology facilitate the development of educational agents. A significant advance is the development of animated pedagogical agents (Johnson, Rickel, & Lester, 2000). Animated interface agents increase the bandwidth of human–computer communication, such that it approaches that of face-to-face communication. Not being restricted to textual or simple graphical communication, animated pedagogical agents engage and motivate students in a more lifelike way.

Network represents another significant advance in technology that promotes the development of educational agents. With a view to the social learning environment, as illustrated in Fig. 1,

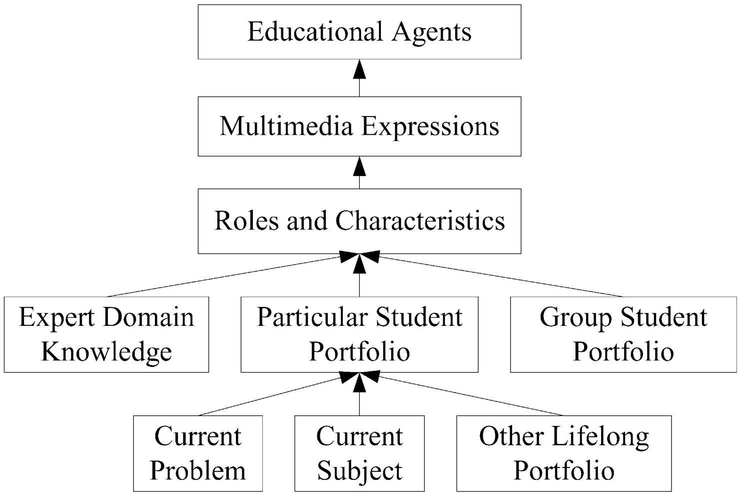

Fig. 3. Uses of a student portfolio.

network increases the convenience of connectivity within a social learning environment. Students can access distant content, operate experimental equipment at different locations, and collaborate with other students all over the world. Network also accelerates research and the development feedback loop. Network increases the convenience of disseminating, sharing, and modifying research, products, and developments.

Educational agents are currently limited to one subject or domain. One aim is the creation of a lifelong learning companion. Such a learning companion is like a friend that stays with the student from childhood to adulthood. The companion stores the student's lifelong portfolio. Educational agents constructed in many domains can be combined into a lifelong learning companion, or the student's lifelong portfolio can at least be exchanged among the educational agents.

A network-based system collects data about students in student portfolios more conveniently than a standalone system. The data can be analyzed and abstracted to build student models, which are abstract representations of students. A particular student portfolio stores data concerning the current problem, current subject, and other lifelong portfolio. The system also uses expert domain knowledge and a portfolio of a group of students to evaluate the student and determine an agent's role, characteristics, and multimedia expressions (Fig. 3). In summary, with respect to implementation, various kinds of educational agents simply represent different ways of using the student's portfolio. Such educational agents are given different names, such as tutor, tutee, collaborator, competitor, troublemaker, or assistant.

6. Conclusion

This study has reviewed the origin of and recent developments in the field of learning companions. Furthermore, the learning companion has been redefined to be applied in a wider spectrum of educational agent research. According to an extensive definition, a learning companion is a kind of educational agent, which plays a non-authoritative role in a social learning environment. On this definition, multiplicity and classification, design issues, implementation issues, development difficulties, technological advances, and the future direction of educational agents have been considered. This study supports the understanding of past, present and future developments in the field of educational agents.

Acknowledgements

The authors would like to thank all the members involved in the Learning Technology project, as well as the staffs of the National Science Council for their long-term support in the past. The LT project is supported by a grant from the Ministry of Education, No. 89-H-FA07-1-4.

References

Abou-jaoude, S., & Frasson, C. (1999). Integrating a believable layer into traditional ITS. In *Proceedings of AI-ED 99 World Conference on Artificial Intelligence in Education*, Le Mans, France, 315–324.

Ackerman, M. S. (1998). Augmenting organization memory: a field study of answer garden. *ACM Transactions on Information Systems*, *16*(3), 203–224.

Aimeur, E., & Frasson, C. (1996). Analyzing a new learning strategy according to different knowledge levels. *Computer and Education*, *27*(2), 115–127.

Aimeur, E., Dufort, H., Leibu, D., & Frasson, C. (1997). Some justifications for the learning by disturbing strategy. In *Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education*, Kobe, Japan, 119–126.

Bernard, Y., & Sandberg, J. (1993). Open learning environment: what support should they offer? *Proceeding of International Conference on Computers in Education*, Taipei, Taiwan, 156–161.

Biswas, G., Katzlberger, T., Brandford, J. Schwartz, X., & TAG-V. (2001). Extending intelligent learning environments with teachable agents to enhance learning. *Artificial Intelligence in Education*, J.D. Moore et al. (Eds.) IOS Press, 389–397.

Bloom, B. S. (1984). The 2 Sigma Problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*, 4–16.

Brophy, S., Biswas, G., Katzlberger, T., Bransford, J., & Schwartz, D. (1999). Teachable agents: combing insights from learning theory and computer science. In *Proceedings of AI-ED 99 World Conference on Artificial Intelligence in Education*, Le Mans, France, 21–28.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, *2*, 155–191.

Bull, S. (1998). 'Do It Yourself' student models for collaborative student modelling and peer interaction. In *Proceedings of 4th International Conference, ITS'98*, San Antonio, Texas, USA, 177–185.

Bull, S., Brna, P., Critchley, S., Davie, K., & Holzherr C. (1999). The Missing Peer, Artificial Peers and the enhancement of human–Human collaborative Student Modelling. In *Proceedings of the 9th International Conference on Artificial Intelligence in Education (AI-ED 99)*, Le Mans, France, 1999, 285–293.

Burke, R., Hammond, K., Kulyukin, V., Lytinen, S., Tomuro, N., & Schoenberg, S. (1997). Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System. *AI Magazine*, *18*(2), 57–66.

Canut, M. F., Gouarderes, G., & Sanchis, E. (1999). The Systemion: A New Agent Model to Design Intelligent Tutoring System. In *Proceedings of the 9th International Conference on Artificial Intelligence in Education (AI-ED 99)*, Le Mans, France, 1999, 54–63.

Carr, B., & Goldstein, I. P. (1977). *Overlays. A theory of modeling for computer-aided instruction*. Cambridge, Massachusetts: AI Lab Meno 406, MIT.

Carbonell, J. R. (1970). AI in CAI: an artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, *11*(4), 190–202.

Chan, T. W. (1995). A Tutorial on Social Learning Systems. In *Emerging Technologies in Education*, T. W. Chan, & J. Self (Eds.), AACE.

Chan, T. W., & Baskin, A. B. (1988). Studying with the prince: The Computer as a Learning Companion. *The Proceedings of International Conference of Intelligent Tutoring Systems*, ITS'88, June, Montreal, Canada, 194–200.

Chan, T. W., & Baskin, A. B. (1990). Learning companion systems. In C. Frasson, & G. Gauthier (Eds.), *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education, Chapter 1*. New Jersey: Ablex Publishing Corporation.

Chan, T. W., & Chou, C. Y. (1995). Simulating a Learning Companion in Reciprocal Tutoring System. *The Proceedings of the First International Conference on Computer-Supported Collaborative Learning*, 49–56.

Chan, T. W., & Chou, C. Y. (1997). Exploring the Design of Computer Supports for Reciprocal Tutoring. *International Journal of Artificial Intelligence in Education*, *8*, 1–29.

Chan, T. W., Chung, Y. L., Ho, R. G., Hou, W. J., & Lin, G. L. (1992). Distributed learning companion systems—WEST revisited. The 2nd International Conference of Intelligent Tutoring Systems, C. Frasson, G. Gauthier, & G. McCalla (Eds.). *Lecture Notes in Computer Science*, 608, Springer-Verlag, 643–650.

Chan, T. W., Hue, C. W., Chou, C. Y., & Tzeng, O. J. L. (2001). Four spaces of network learning models. *Computers & Education*, *37*, 141–161.

Chang, L. J., Wang, J. C., Hsu, B. Y., & Chan, T. W. (1999). Four applications of student modeling—my animal companions, *Proceedings of The Third Global Chinese Conference on Computers in Education*, Macau, 366–370.

Chou, C. Y., Chan, T. W., & Lin, C. J. (2002). An Approach of Implementing General Learning Companions for Problem Solving, *IEEE Transactions on Knowledge and Data Engineering* (in preparation).

Collins, A., Neville, P., & Bielaczyc, K. (2000). The role of different media in designing learning environments. *International Journal of Artificial Intelligence in Education*, *11*, 144–162.

Dillenbourg, P., & Self, J. (1992). People power: a human–computer collaborative learning system. C. Frasson, G. Gauthier, & G. McCalla (Eds.). The 2nd International Conference of Intelligent Tutoring Systems, *Lecture Notes in Computer Science*, 608, Springer-Verlag, 651–660.

Frasson, C., Mengelle, T. Aimeur, E., & Gouarderes, G. (1996). An Actor-Based Architecture for Intelligent Tutoring Systems. In *Proceedings of ITS'96 Conference*, Springer-Verlag, 55–65.

Gilmore, X., & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self (Ed.), *Artificial Intelligence and human learning, intelligent computer-aided instruction* (pp. 179–196). New York: Chapman and Hall.

Goodman, B., Soller, A., Linton, F., & Gaimari, R. (1998). Encouraging Student Reflection and Articulation using a Learning Companion. *International Journal of Artificial Intelligence in Education*, *9*, 237–255.

Greer, J. E., & McCalla, G. I. (1994). *Student Modelling: The Key to Individualized Knowledge-Based Instruction*. Berlin Heidelberg: Springer-Verlag.

Greer, J. E., McCalla, G. I., Collins, J., Kumar, V., Meagher, P., & Vassileva, J. (1998). Supporting peer help and collaboration in distributed workplace environments. *International Journal of Artificial Intelligence in Education*, *9*, 159–177.

Heinrich, E., & Maurer, H. (2000). Active Documents: Concept, Implementation and Applications. *Journal of Universal Computer Science*, *6*(12), 1197–1202.

Hietala, P., & Niemirepo, T. (1998). The Competence of Learning Companion Agents. *International Journal of Artificial Intelligence in Education*, *9*, 178–192.

Johnson, W. L., Rickel, J. W., & Lester, J. C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, *11*, 47–78.

Leman, S., Giroux, S., & Marcenac, P. (1995). A Multi-Agent Approach to Model Student Reasoning Process. In *Proceedings of AI-ED 95 World Conference on Artificial Intelligence in Education*, Washington, USA, 258–265.

Leman, S., Marcenac, P., & Giroux, S. (1996). A Generic Architecture for ITS Based on a Multi-agent Approach. *Intelligent Tutoring Systems,* Proceedings of Third International Conference, ITS '96, Montrél, Canada, Lecture Notes in Computer Science, Vol. 1086, Springer, 75–83.

Les, J., Cumming, G., & Finch, S. (1999). Agent systems for diversity in human learning. In *Proceedings of AI-ED 99 World Conference on Artificial Intelligence in Education*, Le Mans, France, 13–20.

Lester, J. C., Towns, S. G., & FitzGerald, P. J. (1999). Achieving Affective Impact: Visual Emotive Communication in Lifelike Pedagogical Agents. *International Journal of Artificial Intelligence in Education*, *10*, 278–291.

McCalla, G. I., Greer, J. E., Kumar, V. S., Meagher, P., Collins, J. A., Tkatch, R., & Parkinson, B. (1997). A Peer Help System for Workplace Training. In *Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education*, Kobe, Japan, 183–190.

Mizoguchi, R. (1995). Student Modeling in ITS. In *Emerging Technologies in Education*, T. W. Chan, & J. Self (Eds.), AACE, 35–48.

Mizoguchi, R., & Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems. *International Journal of Artificial Intelligence in Education*, *11*, 107–121.

Murray, T. (1999). Authoring intelligent tutoring systems: an analysis of state of the art. *International Journal of Artificial Intelligence in Education*, *10*, 98–129.

Nichols, D. (1994). Issues in designing learning by teaching systems. *In Proceedings of the East-West International Conference on Computer Technologies in Education (EW-ED'94)*, 176–181.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. *Cognition and Instruction*, *1*, 117–175.

Palthepu, S., Greer, J., & McCalla, G. (1991). Learning by teaching. *The Proceedings of the International Conference on the Learning Sciences*, AACE, 357–363.

Ramirez Uresti, J. A. (2000). Should I Teach my Computer Peer? Some Issues in Teaching a Learning Companion. In G. Gauthier, C. Frasson, & K. Vanlehn (Eds.). *Intelligent tutoring Systems. Fifth International Conference, ITS'2000*, Vol. 1839 of Lectures Notes of Computer Science, Springer-Verlag, 103–112.

Scott, L. A., & Reif. F. (1999). Teaching Scientific Thinking Skills: Students and Computers Coaching Each Other. In *Proceedings of AI-ED 99 World Conference on Artificial Intelligence in Education*, Le Mans, France, 285–293.

Self, J. A. (1974). Student models in computer-aided instruction. *International Journal of Man–Machine Studies*, *6*, 261–276.

Self, J. A. (1988). Bypassing the intractable problem of student modeling. *International Conference of Intelligent Tutoring Systems, ITS'88*, Montreal, Canada, 18–24.

Sleeman, D., & Brown, J. (1982). *Intelligent Tutoring Systems*. Academic Press.

Tecuci, G., & Keeling, H. (1999). Developing an Intelligent Educational Agent with Disciple. *International Journal of Artificial Intelligence in Education*, *10*, 221–237.

Thibodeau, M. A., Belanger, S., & Frasson, C. (2000). WHITE RABBIT—MatchMaking of User Profiles Based on Discussion Analysis Using Intelligent Agents.*Intelligent tutoring Systems. Fifth International Conference, ITS'2000*, Vol. 1839 of Lectures Notes of Computer Science, Springer-Verlag, 113–122.

Ur, S., & VanLehn, K. (1995). STEPS: A Simulated, Tutorable Physics Student. *Journal of Artificial Intelligence in Education*, *6*(4), 405–435.

Vassileva, J., Greer, J., McCalla, G., & Deters, R. (1999). A Multi-Agent Design of a Peer-Help Environment. In *Proceedings of AI-ED 99 World Conference on Artificial Intelligence in Education*, Le Mans, France, 38–47.

Vygotsky, L. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.

Wang, W. C., & Chan, T. W. (2000). CAROL5: An Agent-Oriented Programming Language for Developing Social Learning Systems. *International Journal of Artificial Intelligence in Education*, *11*, 1–32.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.

Yin, J., El-Nasr, M. S., Yang, L., & Yen, J. (1998). Incorporating Personality into a Multi-Agent Intelligent System for Training Teachers. *Intelligent tutoring Systems. 4th International Conference, ITS'98*, Vol. 1452 of Lectures Notes of Computer Science, Springer-Verlag, 485–493.