A Tutorial on Social Learning Systems

Tak-Wai Chan

Institute of Computer Science and Information Engineering
National Central University
Chung-Li, Taiwan 32054, R. O. C.

email: chan@sparc15.src.ncu.edu.tw

Abstract: If most knowledge can be regarded as a common consensus of a community, then learning should naturally take place in a socio-cultural environment. Furthermore, if social context is a catalyst of knowledge cultivation and motivation, then this element of learning should be emphasized in most computer-assisted learning systems. Social learning systems are emerging learning environments that involve multiple agents, working at the same computer or across connected machines. These agents are either computer simulated or real human beings, taking various roles via different protocols of learning activity. As alternatives to traditional one-to-one tutoring, these systems are sometimes called or closely related to collaborative, distributed, or distance learning systems.

BACKGROUND

Long before we had formal schools as the most prevalent educational setting, apprenticeship learning, through generations, was the means of educating the young (Collins et al., 1989). Besides the domain knowledge or skill, morals, attitude, sociability, and so on, which are aspects for the well-being of the community, were usually passed on by the master to the apprentice. Today, knowledge or skill has become so important that whether one can survive in the society depends on what and how much one can learn. For financial reasons or others, schools emerged as factories for mass production where students are inputs and graduates with acquired knowledge are outputs. Quality of learning is at stake when teachers have to face tens of students in a class. In fact, a classic study shows that one-on-one tutoring[1] is four times more effective than classroom teaching (Bloom, 1984).

Computer technology opens alternatives. An *intelligent tutoring system* intends

[1] Some music schools still adopt the appenticeship learning for their graduate student study.

to provide individual treatment by impersonating a tutor -- interacting with the student and looking over the learning process. Being different from one-to-one tutoring systems, social learning systems comprise an additional element, namely, peer interaction. In fact, to incorporate peer interaction in a traditional educational setting, group learning has been being studied extensively since the 1920s (Collis, 1993). However, with the advent of the information highway that is able to support various functions and service, instead of going to school, bringing school home by communicating with remote teacher and classmates at home is within the realm of possibility. When such technology can be accessed by most families, we shall anticipate a viable alternative to the 'economic' formal school settings.

With the growing concern of how cognition is connected to social process, the work of Piaget and Vygotsky are frequently referenced. Peers, as Piaget indicated, help children shift away from egocentric views and consider multiple perspectives, thus fostering a more comprehensive mature conception to emerge and taking a leap to a higher level of understanding (Piaget, 1965). The recent interest in viewing learning as social practice has been strongly influenced by Vygotsky. Cognitive development process, as Vygotsky put it, is the gradual internalization and personalization of what was originally a social activity (Vygotsky, 1978). Vygotsky's *zone of proximal development* is the hypothetical distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers. Experiments have shown that two children working together can successfully perform a task which cannot be performed by children of the same age working alone (Doise et al., 1975; Mugny & Doise, 1978).

Recent advancement of learning theories has further spurred this line of research. For example, constructivism and situated learning share a common view of the development of socially shared cognition. Constructivism holds that there is a real world we experience. Learners create their personal interpretation of the world based upon their past experience and their interaction with the world (Bednar, 1991; Jonassen, 1992). The emphasis of constructivism is on how we *construct* knowledge, rather than solely the receiving of canonical knowledge. On one hand, going beyond individual brain or mind, knowledge can be *jointly constructed* by people; on the other hand, social interaction in learning can trigger the *reconstruction* process of what was learned, ameliorating knowledge transfer. Situated learning speaks of domain dependent and context rich learning environment, practical knowledge, authentic activities in realistic settings, etc., (Brown et al., 1989). In particular, learning in a workplace is socially situated in a 'community of practice' and means earning 'membership of that community'. This view sees mind, cultural, history, and the social world as interrelated processes (Lave & Wenger, 1991; Clancey, 1992).

In artificial intelligence research, the dominant role of formal logic and symbol processing has been challenged for its isolation from both the physical and social world in understanding cognition (e.g. Winogard & Flores, 1986). In contrast, distributed artificial intelligence research on models such as blackboard systems studies the roles of negotiation, argument, and contradiction among individual agents that determine decisions, judgments, and problem solutions. This change in the field will press for more research into theories and models concerned with human thinking and social functioning.

Within the community of intelligent computer assisted learning research, we can briefly trace the history of the effort on social learning systems. Computer assisted instruction (CAI) systems began in the sixties with educators being the main players. Influenced by behaviorist psychology, they started with the stimulus-response approach which well matched the 'interactive' environment provided by computer. In the late seventies, computer scientists who are mainly involved in artificial intelligence (AI) research launched research into *intelligent tutoring systems* (Sleeman & Brown, 1982; Clancey, et al., 1982). Their approach is to simulate the computer as an intelligent tutor who can understand the student and provide adaptive tutoring. Research focused on cognitive diagnosis, student modeling, representation of domain knowledge, teaching strategies, and so on. Early work on intelligent tutoring systems (ITS) was heartening. During the eighties, the swiftly mounting interest in AI drew more computer scientists into ITS research. Soon, the difficulties of building ITS became apparent. For example, the student model which accounts for understanding the student's status in learning is hard to build (Self, 1988). In the mid eighties, the ITS model was being challenged, in particular, its role as an authorized teacher for transmitting certified knowledge by Self and his colleagues (Self, 1985, 1986; Gilmore & Self 1988). They suggested that the computer can *cooperate* with the student in learning, instead of teaching the student. In 1988, Chan and Baskin proposed an alternative model, *learning companion systems* (Chan & Baskin, 1988). They suggested that the computer can be simulated as two co-existing agents: a teacher and a learning companion. The learning companion learns together with the student under the guidance of the teacher. Thus, the learning companion performs the learning task at about the same level as the student, and both the student and the companion exchange ideas while being presented the same material by the teacher.

Chan and Baskin (Chan & Baskin, 1988, p.199) further suggested that the human student may 'teach' the learning companion. For example, by providing knowledge and examples for the learning companion, and observing how the companion solves the problems and improves performance, the student "learns how to learn by teaching the learning companion." This inverted model of ITS is termed as *learning by teaching* and is recently being further elaborated by other researchers (Palthepu et al., 1991; Nichols, 1993; VanLehn et al., in press). For example, besides refining or tuning the acquired knowledge, this model is also potentially useful for teacher training.

Figure 1 summarizes the different models. Simply speaking, if an agent has some expertise of a domain, then the agent can act as a teacher and the other agent as a student. If the knowledge of both agents is at a similar level, then they can be learning companions of each other. Learning by teaching, of course, assumes that the teacher agent has learned something about the domain and can benefit from the process of teaching the student agent.

THIS PAGE INTENTIONALLY BLANK

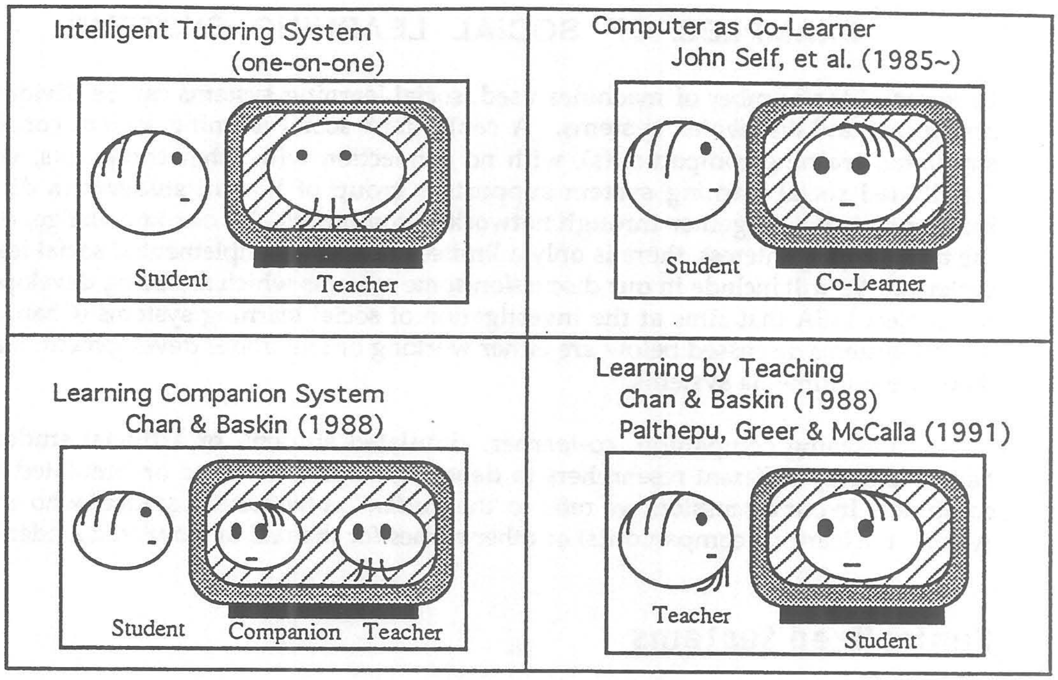


Figure 1. Initial Models of Social Learning Systems

Emerging multimedia technology provides powerful devices to support complicated peer dialogue, gestures, and so on, making complex social learning systems more viable than before. Also, besides interaction with computer simulated students, through computer or telecommunication network, distance education allows human students at different locations to learn together. For example, students solve problems across two connected machines in the Distributed West system (Chan et al., 1992, see next section) with collaboration or competition. Social learning models are manifold: the agents can be computer simulated or real human students, working at the same computer or across networked machines, and under different protocols of learning activity (cf. Chan & Baskin, 1988, p.199).

Of course, the technical problems have to be overcome. For example, like ITS, building the group student model in social learning systems is difficult and distance learning could be comparable to typical classrooms if interaction between students cannot be strengthened or appropriately controlled (Looi, 1994). As a tutorial, this chapter only intends to take a glance at the infrastructure of these emerging technology supported learning environments. The remainder of this chapter will introduce some social learning systems and give a framework to classify and design social learning systems.

EXAMPLES. OF SOCIAL LEARNING SYSTEMS

In terms of the number of machines used, social learning systems can be divided into *centralized* and *distributed* systems. A centralized social learning system consists of simulated learning companion(s), with no connection with other computers, while a distributed social learning system supports a group of human students in different locations learning together through networked computers. To our knowledge, despite the fast growing interest, there is only a limited number of implemented social learning systems. We will include in our discussion some systems which are being developed for the project LISA that aims at the investigation of social learning systems (Chan, 1993). All the systems discussed below are either working or still under development; many of them are multimedia systems.

Learning companion, co-learner, simulated student, or artificial student are names used by different researchers to denote the agent modeled or simulated by the computer. In our discussion, we refer to 'the student' as the human student who uses the system, the learning companion(s) or other names for the real or simulated student(s).

Centralized Systems

Integration-Kid

Integration-Kid is a learning companion system (student-teacher-companion model) in the domain of indefinite integration, covering a complete but short course, at the level of a first year undergraduate (Chan, 1991). Different *protocols of learning activities* in different stages of learning are designed and the role of each agent, such as collaborator or competitor, is defined. For example, in solving miscellaneous problems, the protocol for *responsibility sharing* decomposes the problem solving process of the student and the companion into negotiation, decision, and execution. One is responsible for decision making and the other is for the execution of the decision made. In the case where both the student and the companion cannot solve the problem, the teacher will interrupt to help. Figure 2 shows the interface of Integration-Kid. There are four panes where each agent is represented by a pane and the fourth pane provides a hypertext-like sub-system for the student to retrieve related information. The implementation of the learning companion follows a *simulation* approach where the growing domain knowledge of the companion can be viewed as selective use of the knowledge preset by the system together with possible misconceptions of a typical student (further discussed in the next section). Unfortunately, Integration-Kid has not been fully evaluated, with only a few teaching trials by human students. But we noticed that the students seemed to be very curious about the response of the companion agent and paid more attention to the companion than to the teacher.

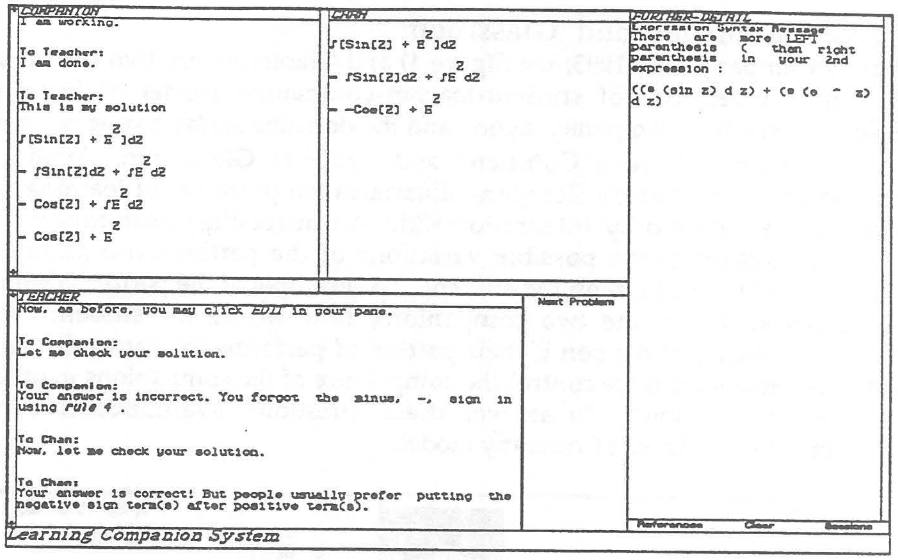


Figure 2. Interface of Integratiion-Kid

People Power

People Power (Dillenbourg & Self, 1992; see Figure 3) is a computer supported collaborative system where the co-learner acts as a collaborator of the student. The People Power system investigates how social dialogue generates structures for the reflective dialogue. Reasoning and collaborating are two instances of dialogue: reasoning is a dialogue with oneself and collaborating a dialogue with the peer. People Power contains a microworld in which the student can create an electoral system and simulate election. The co-learner stores and replays dialogue patterns in communicating with the student, resulting in a sequence of mutually refuted argument patterns. The procedure uses two arguments, the proposer and the critic, and acts as a theorem prover to prove that some change in the country map leads to a gain of seats for a particular party. It explores a tree of rules (or arguments), in a depth first search.

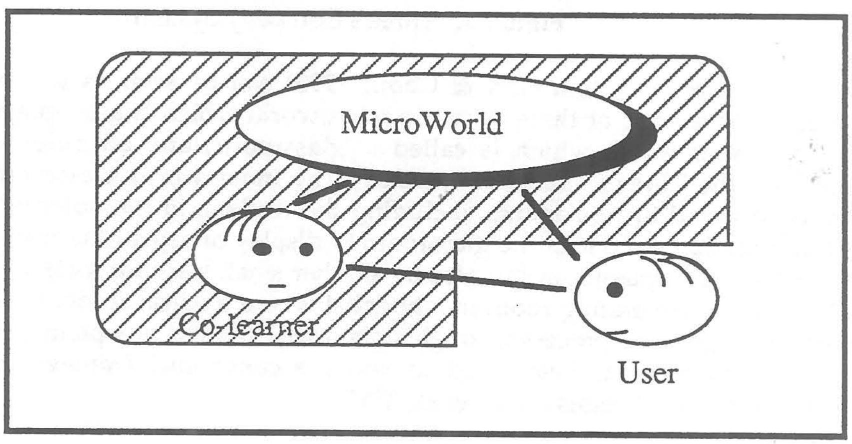


Figure 3. Environment of People Power

Three's Company and Glassroom

Three's Company (Lin, 1993; see Figure 4) and Glassroom are two similar systems: both are direct extensions of student-teacher-companion model of Integration-Kid by including one more companion agent, and the domains are both programming languages: teaching Lisp in Three's Company and ADT[2] in Glassroom. With an additional companion agent, Three's Company illustrates ten protocols of learning activities with some of them shared by Integration-Kid. An interesting issue raised by the Three's Company system is the possible variations of the performance patterns among the students and their effects on the student. For example, if the performance of the student lies between that of the two companions, how would the student's motivation be affected? What will happen if their pattern of performance varies during the learning process? How should we control the competence of the companions in order to facilitate student's motivation? To answer these questions, evaluation is currently being conducted for the Three's Company model.

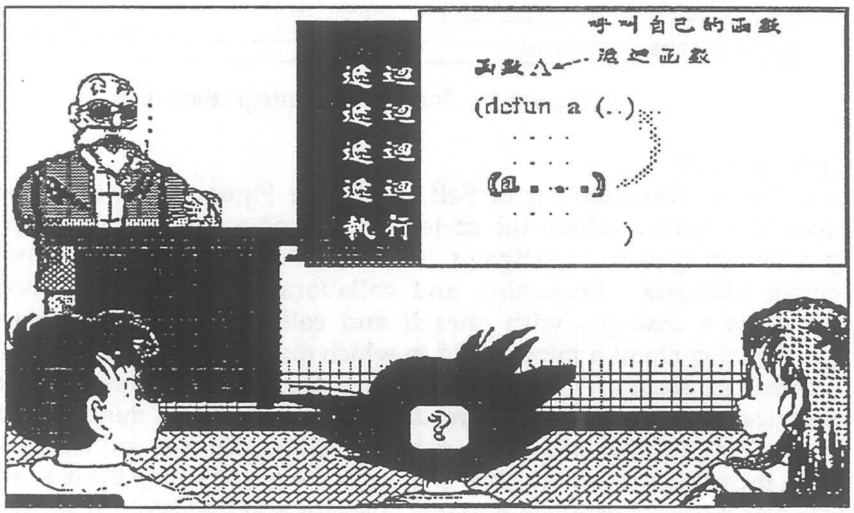


Figure 4. Three's Company System

Glassroom system (Lin & Chou, 1993) can be seen as a version of Three's Company, consisting of three parts: a microworld which is a language interpreter of ADT, an inner world which is called a 'glassroom', and an outer world which is essentially the Three's Company model. The inner world glassroom demonstrates learning material for the agents, including the student, in the outer world. Agents in the outer world can control the glassroom to display or replay the material, and thus is more active. The glassroom can also be another small learning society. This scenario is analogous to an operating room in a hospital where medical students outside the room observe the operation process through a glass window and is explained by an instructor. Glassroom system has been used to verify a conceptual framework of knowledge construction sub-processes (Chan et al., 1993).

[2] ADT is a simplified verion of CLU language designed for the purpose of teaching abstract data types (Liskov, 1977; Chan & Wang, 1992).

Rescue

Role playing games (RPG) engage the user to interact with various roles in a scene. Thus, if the purpose of an RPG is for learning, it is naturally a social learning system. Rescue (Cheng, 1993) is a RPG where the student plays the role called MacGyver whose task is to rescue the kidnapped President's daughter. MacGyver needs to meet and talk to other roles (or people) and explore different environments in order to collect information to achieve the task. When MacGyver communicates with other roles, he has to send messages to these roles by constructing the messages through a menu bar (see Figure 5). The message construction process is in effect calling a particular method of an object with a message sending to another object in a small language called Prototype (Lieberman, 1986). This language is designed for teaching object-oriented programming concepts such as object, state variable, method, delegation, and message passing. We hope that, through playing the game, students can master some concepts of object-orientation. The system has been implemented and evaluation will be conducted.

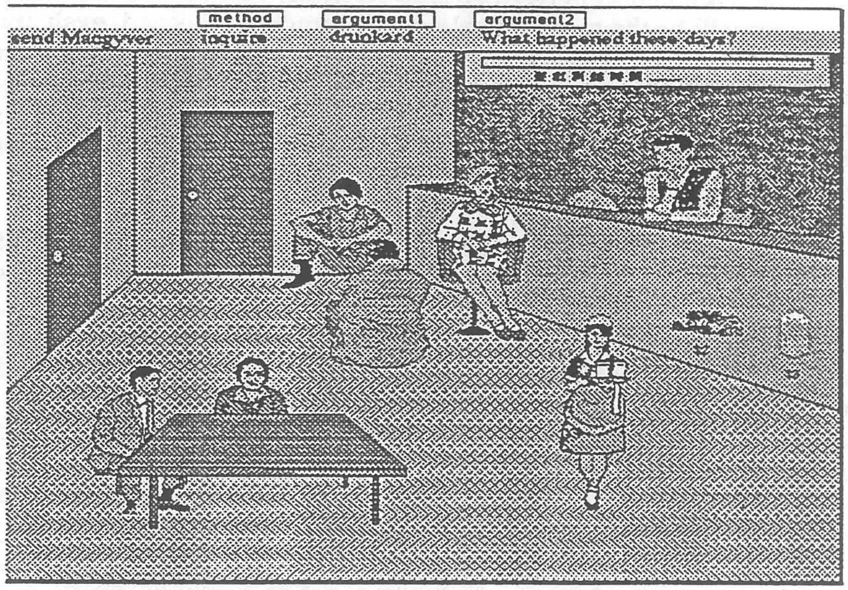


Figure 5. Rescue: A Role Playing Game for Learning

Distributed Systems

Machines in the distributed systems in our discussion are connected with local area network. The networked computers can all be located in a computer room or distributed in different parts of a building or nearby buildings. Thus, these systems can also be viewed as testbeds for distance learning in the future.

Distributed West and Contest-Kid

Distributed West and Contest-Kid are both competitive learning environments in which students play competition games through the network. Distributed West is a re-implementation of a CAI and coaching system classic, West, where a student plays against the computer opponent in trying to reach the destination before the opponent does (Burton & Brown, 1979). At each turn, an agent (the student or the computer opponent) receives three randomly generated numbers. Using this set of numbers and arithmetic operators, the student may form an arithmetic expression. The system then evaluates this expression to a number and moves the icon representing the student towards the destination, with the number of steps equal to the number obtained. The goal of the original West system is to engage and motivate students in a game environment for practicing arithmetic evaluation. In Distributed West, instead of the computer opponent, real students (usually two) at one site play against the other students at the other end of the network. Besides human competition, students at the two ends can also collaborate together against the computer opponent. Students on each side have an evaluator (a passive teacher) to judge whether the arithmetic expression can lead to a good move. Distributed West illustrates two interesting issues. First, there are a tremendous number of variations of such environments that may affect the student's motivation and performance in learning, making it difficult to decide which environment will be the most suitable one for students. Second, evaluation shows that more students prefer playing against each other in teams rather than cooperating together against the computer opponent, with a ratio 3 to 1. Students think that such competitive environment for learning is effective. This result is rather unanticipated since revealing students' strengths and weaknesses in competition is not encouraged, despite motives and striving attitude engendered by it.

Contest-Kid (Lai, 1994, see Figure 6) is a simple system designed for a group of unlimited number of students to learn simultaneously. Each student occupies a computer, not shared with other students. Activity proceeds in two phases: the learning phase and the testing phase. Competition occurs in the testing phase. In the learning phase, students watch a piece of video on their computers. Then in the testing phase, students are asked a set of questions, directly or indirectly related to the content of the video. The correct answer will be shown after the students give their answers. For the purpose of evaluation, the testing phase is divided into three parts. In the first set of questions, students do not involve competition, just answer the computer's questions. No student knows any other student's score. In the second set of questions, the students can decide whether to participate in a competition. For those who participate, their scores will be known to every participant. In the final set of questions, every student's score is known to all other students, that is, the competition is compulsory.

Our preliminary result shows that, for the extent of competition, 13% of students prefer no competition, 80% prefer optional competition, and only 7% prefer compulsory competition. For the number of students involved in competition, only 13% of students prefer pairs, the majority (61%) prefer in small group, say, seven people, and the rest (26%) prefer a large group, say, 50 people. For the types of learning companions, we have conducted small group competition (seven agents) with the following composition: three human students who are classmates (so they know each other), two *pretend* companions, and two simulated companions. We told students that the two simulated companions are simulations of their classmates and the two pretend companions are two *real* students of another department working at remote sites. In fact, it is a white lie because the two pretend companions are actually another two simulated companions. We found that two third of students pay more attention to their classmates and the rest to the pretend students. No student pays attention to the simulated companions. We did a second experiment and assigned an attribute to the simulated companions: we told the students that the two simulated students are simulations of two students coming from two local schools which they know are of different standings. This time we found that 43% of students pay more attention to their classmates, 36% to simulated companions, and 21% to the pretend companions, a significant increase of attention to the simulated students.

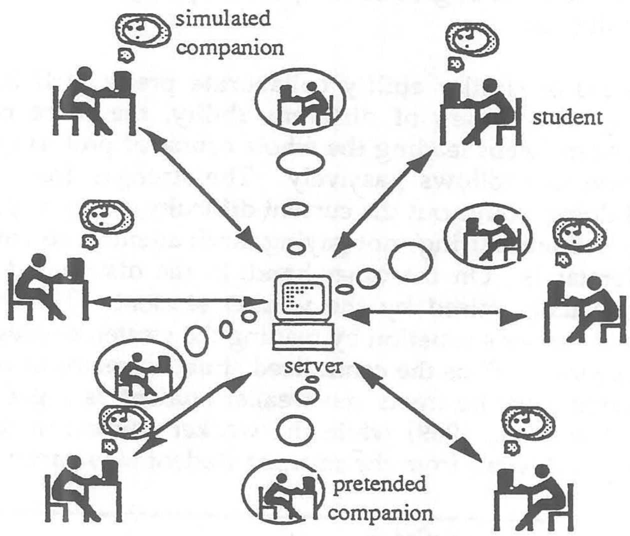


Figure 6. Contest-Kid System

TurtleGraph, Reciprocal-Kid, and Meta-Tutoring-Kid

The aim of TurtleGraph (Jehng et al., 1994) is to study meta-communication skills, such as regulation of students' individual communication behavior and their organization of effective conversation. It is a collaborative system where two students can have free discussion through the network in solving non-trivial recursive programming problems. The programming language is in Lisp syntax with LOGO subroutines to generate geometric figures. A turtle window in the interface contains a turtle for drawing and shows the geometric figure with the movement of the turtle once a program is executed (Figure 7). At anytime, the students can discuss by pressing the communication button and writing statements on a dialogue box. Besides discussion, the student may see or copy the other student's partial solution.

Preliminary evaluation has been conducted in comparing the TurtleGraph system and its centralized version where two students sit together and work at the same computer to solve problems. The result shows a number of differences and we shall mention a few here (Liang, 1994). In the distributed situation, the students try to get the problem done independently first and only at the later stage, when difficulty arises, one of them or both will initiate discussion. Thus, discussion is a free invitation. In contrast, the centralized situation seems to coerce discussion and collaboration from the beginning to the end, and utterances in the face-to-face dialogue are much more frequent than in the distributed situation. We can explain that in the distributed situation, students need to type to express their ideas as well as to coordinate the communication, for example, to inform the other student to be ready for exchange. However, when students work together at the same computer, they can express freely by speaking up in their own natural language and do not need any coordination. Thus, the main activity in face-to-face situation is dialogue. Also, in distributed situation, students illustrate some degree of competition partly because they may compare their own partial solutions.

Students of similar ability collaborate pretty well in both centralized and distributed systems; when of different ability, the more capable student in the centralized system keeps leading the whole course of problem solving activity and the weaker student just follows passively. The stronger student explains what he is thinking and doing, points out the current difficulty, suggests possible trials, and so on, in a way like 'self murmuring', not paying much attention to whether or not the weaker student understands. On the other hand, in the distributed situation, questions or requests are usually raised by the weaker student. The stronger student has to understand his partner's situation by reading the written messages of the partner before giving his responses. Thus the centralized situation seems to be more beneficial to the stronger student since he treats the weaker student as a gadget for generating self-explanation (Chi et al., 1989) while the weaker student in the distributed situation gets more help and learns from the stronger student who partly takes the role of a tutor.

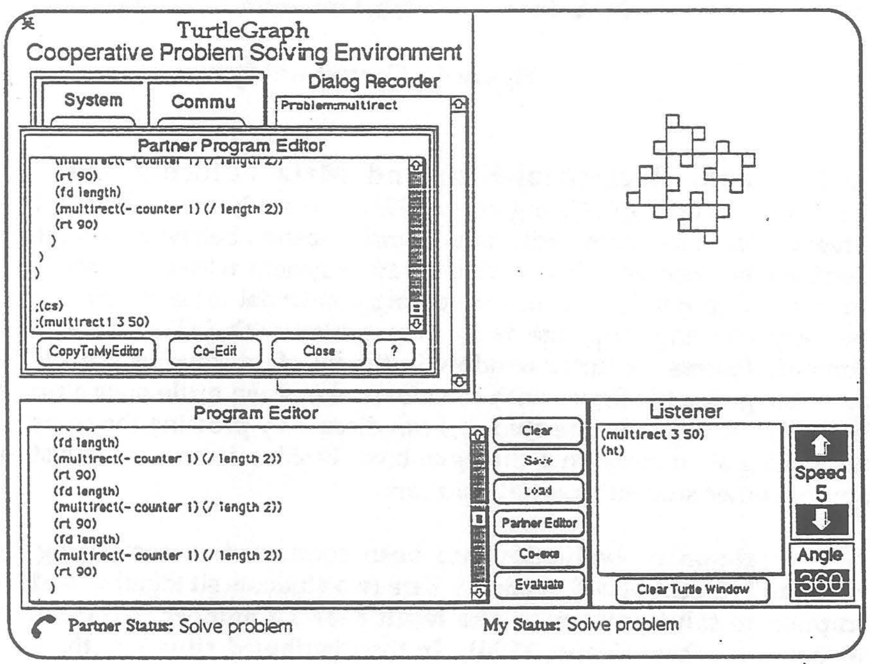


Figure 7. Interface of TurtleGraph

TurtleGraph currently does not have a computer tutor and is more like a 'tool' for collaboration. When the students reach an impasse, distributed problem solving and discussion seem to be more helpful to clarify misconceptions and validate hypotheses while in face-to-face interactions, students spend a long time in discussions and have a higher chance of being unable to conclude them. An explanation of this is that the intensive conversation in centralized situation does not allow much opportunity for individual thinking. The partial solution they obtained thus far belong to both of them, not any one of them. The cognitive conflict between the students then is not strong enough for leading to valuable consequences in the conflict resolution process.

Students' collaboration in Reciprocal-Kid is more tightly coupled than in TurtleGraph. The learning task is divided into sub-tasks and students work on these sub-tasks by taking turns. In Integration-Kid, this protocol of learning activity is termed as *responsibility sharing,* emphasizing the participants' obligation and their contribution to the accomplishment of the entire task. Such design of small group learning activity has been called *reciprocal teaching* and studied in the domain of reading (Palincsar & Brown, 1984) where the reading task is divided into questioning, summarizing, elaborating, and predicting. Reciprocal-Kid is still in the development stage.

The domain of Reciprocal-Kid is learning Lisp programming. The programming task is decomposed into planning, executing, and critiquing. Correspondingly, there are three real human agents called *planner, executor,* and *critic.* Besides these three agents, there is a *computer tutor* who may give advice and play the final judge if conflicts between different agents arise (Figure 8).

Reciprocal-Kid is built on top of a rather nicely designed and effective scaffolding system for learning Lisp programming, Petal (Bhuiyan et al. 1992). The planner first constructs natural language like pseudo code as a plan for the first step. If the planner has no idea, he or she can ask the tutor. The critic then makes comments about the plan. If the critic has no problem with the plan, then the executor turns the plan into the Lisp code. Again, the critic will judge whether the executor write the code correctly. If the critic approves, then the computer tutor will make the final judgment to see whether the current step is correct before leaving the agents to repeat the same procedure in the next step.

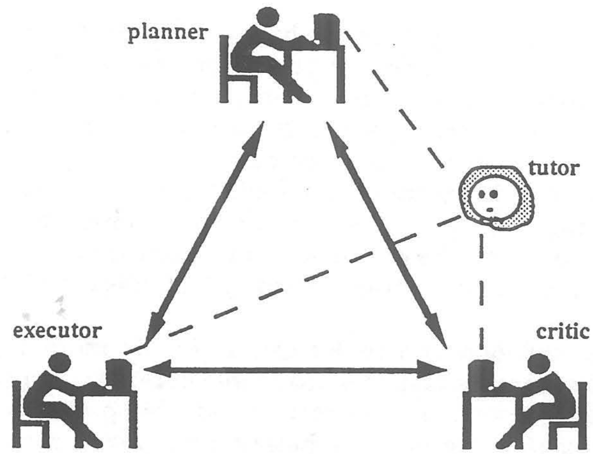


Figure 8. Reciprocal-Kid

Meta-tutoring (tutoring the tutor) is the idea advocated by Kurt VanLehn in his keynote speech in AI-ED 93. Meta-Tutoring-Kid (MTK) is a distributed system for learning by teaching. It consists of a collaborative human dyad: one plays the role of a tutee who is solving the problem and the other plays the role of a tutor who knows the answer and gives advice when the tutee needs. The dyad switch roles for alternative problems (Figure 9).

The idea of building MTK is spun off from building the Reciprocal-Kid since MTK is a simpler version of Reciprocal-Kid. Note that the tutee agent combines the roles of the planner and the executor in Reciprocal-Kid, and the tutor can at the same time take the roles of the critic and the computer tutor. We have implemented a student model of the tutee. This student model turns out to be a critical component for helping the tutor to give advice to the tutee. This is an example use of the student model to interact directly with the user to promote indirect reflection (cf Bull et al., 1993).

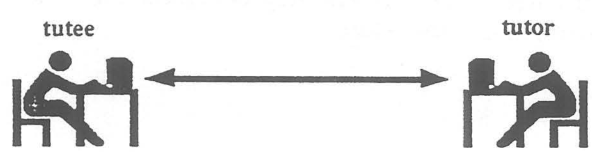


Figure 9. Meta-Tutoring-Kid

Discussion-Kid and ICLS's Group Leader Paradigm

Discussion Kid, which is still under development, is a system that will be used as a testbed for experimenting with ideas about people's verbal behavior and social manner in a distance learning environment. Hopefully, Discussion Kid can become a useful tool for training students to develop verbal and social skill in group discussion. For the moment, two users with access to computer network can use the system at the same time. Through the network, they would watch a short video on their own screens at the same time. We have chosen a video whose contents would provoke their feelings about the reliability of computer technology. When the video ends, a menu of statements will appear. A student would select a menu item to indicate his or her stand. When a choice is made, a new menu that contains responses to the choice will appear so that the other user can respond. If a user does not like any of the menu items, he or she can type in his or her own statement. The system would record this statement and include this as a new item in the menu when the system is used later. In this way, the system would expand its "dialog base" as it is used by more people. A small dialog base was collected through an initial experiment, in which several human subjects freely expressed their opinions after they watched the video. With this initial dialog base, users can go through the discussion session without making too much effort. We plan to implement the dialog base as a finite state transition network. When we want to design another dialog base on another subject matter, all we need to do is to write down the dialog base as a transition network. This network will then be taken as data by the system for the generation of the dialog menu.

The purpose of the Intelligent Collaborative Learning System (ICLS) (McManus & Aiken, 1993), is both to enable students to work cooperatively in structured groups and to teach them how to work effectively in a collaborative learning environment. The principles of the system are based on the theories of collaborative learning, intelligent tutoring systems, and CSCW (Computer Supported Collaborative Work Systems). In the ICLS, the students work collaboratively in groups in the Jigsaw method of collaborative learning (Aronson et al., 1978). Each student works at his or her own computer, in a networked environment, running a student interface program. The students work on both the discussion and the task levels as described by Cumming and Self (1990). The students work on the task level as they develop a group project in any academic domain. They also work on the discussion level as they discuss their work with each other.

The Group Leader paradigm in the ICLS provides the pedagogical expertise for the system. The Group Leader computer tutor serves as a group manager, discussion facilitator, and collaborative skills teacher. The Group Leader manages the group organization and meetings and facilitates discussions by processing the communications among the students. Student communications are messages built from a sentence opener, which represents a collaborative skill defined by Johnson and Johnson (1991), and supplementary text. The Group Leader monitors the students' discussions, tutoring the students in their use of collaborative skills to lead the discussions towards conclusion or convergence (Roschelle, 1992).

COSOFT and MITS

COSOFT and MITS are instances of distributed systems where there is a human tutor. COSOFT, which is in the development stage, intends to enrich interactions between the students and the teacher in the traditional teacher-centered classroom settings through Socratic dialogues, prepared exercises, and reflection, with instructional modes such as exposition, individual or group exercises, and open and guided discussion (Hoope et al., 1993). Central to COSOFT is a 'liveboard' which is connected to local satellite workstations for the students. Liveboard is a large computer display for multi-media presentation with multiple functions. For example, parts of the screen on the liveboard can be shared by the local workstations (in both directions) and can be manipulated by means of an electronic pen. Student exercises are supported by 'electronic worksheets' which can be conceived as interactive multimedia documents. Interactions occur in three areas: the teacher's private interface, the student's interfaces, and the liveboard screen.

The MITS (see one of the chapters in this book) system provides students on-line tutoring or various consulting services when students are working on their home assignments with a computer. Thus, students who are working on their assignments do not have to leave the computers to get help. The help can be available directly from the on-line tutor. When students initiate a remote session by sending a request to the tutor, on a first come first served basis, a shared screen and a voice channel can be established between the student and the tutor. As can be seen, because of the existence of the human tutor, both COSOFT and MITS, instead of viewing computers as intelligent, active counterparts (as in ITS), avoid the difficulties of modeling 'shared understanding' between tutor and tutee, and strike for less knowledge-dependent learning support mechanisms (Hoope et al., 1993). Such systems, in general, can be viewed as CSCW or desktop multi-media conferencing systems for educational purpose.

SPOT: A FRAMEWORK OF DESIGN ISSUES

This section gives a framework, SPOT (Structure, Protocols, Outcomes, and Techniques), of classifying and designing social learning systems. Structure deals with the physical configuration and properties of the components of the systems. Protocols refer to the protocols of learning activity, or 'rules of games', during learning. Outcome is what students have gained or changed after using the systems. Finally, techniques are concerned with the implementation.

Structure

Configuration

Configuration of a social learning system largely confines particular types of research problems to investigate. In general, as mentioned before, social learning systems can usually be categorized into centralized and distributed systems. The simplest kind of centralized social learning systems is 'one-computer-many-students', that is, a small group of students (may be two) use a single computer. A good deal of educational research has been in this form (Hooper, 1992). The experience of TurtleGraph indicates a significant difference between the outcome of the centralized and distributed versions of such systems. If the software is originally designed for single users, a human tutor may need to manage the learning activity when it is used for group learning. It is desirable that educational software for standalone computers, like some commercial entertainment software, provides versions for group players.

Systems such as Integration-Kid, Three's Company, Glassroom, and educational role playing games where multiple agents are represented in a single computer are another type of centralized social learning systems. The computer teacher, one or more simulated learning companions, together with the human student, form a small learning society.

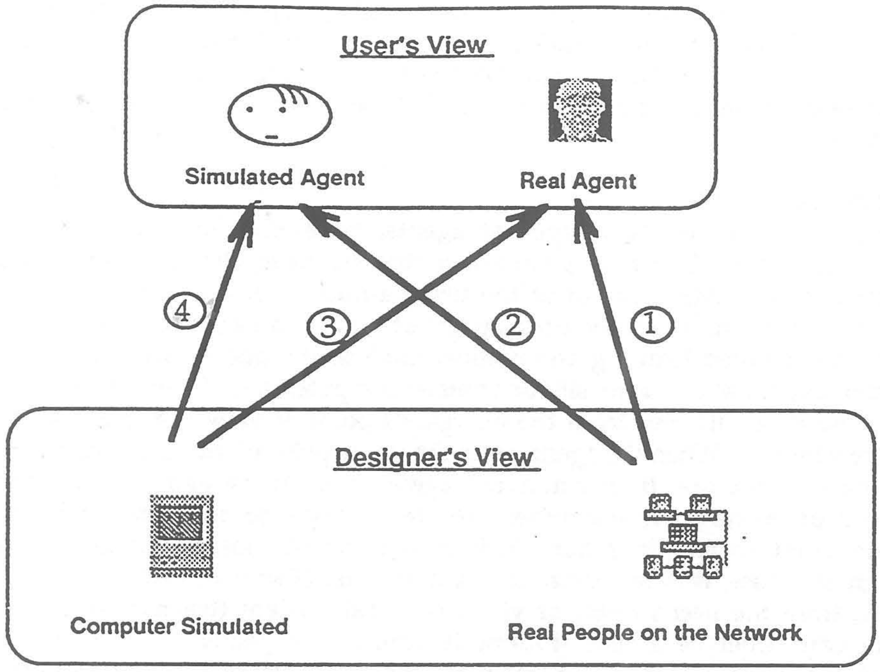
Types of Agents

In general, there are two types of agents, either real human agents or computer simulated agents. The agents are either students or teachers. If we want the agent to function as a teacher but not of the usual authority of a teacher, this agent can be a teaching assistant. From the user's point of view, if an agent is a student, this agent is either a simulated learning companion, such as the one in Integration-Kid, or a real human student at the same site or another computer site. However, the user's point of view could be different from the designer's point of view in the actual situation (see Figure 9 below). When the agent, from the user's point of view, is a learning companion, this agent can either be a simulated agent or it can be another real student, a real teacher, or the designer at another site. In the last case, the user is told that the agent is simulated, in fact, it is not. This type of agents, for pedagogical, evaluation, or design purposes, is called 'Wizard of Oz' method (Twidale, 1993). Similarly, when the agent, from the user's point of view, is a real student (teacher) at another site, this agent can either be a real student (teacher) as expected by the user or in fact a simulated learning companion (teacher), called a pretend student (teacher). See Figure 10 below.

The agents, whether students, teachers, or teaching assistants, discussed above have the same concern of the user in their activity, that is, the learning task. This is different from most role playing games where other agents may not be concerned with the task tackled by the user. For example, if the user plays the role of an employee, the company's employer may not be concerned with the particular task taken by the employee. However, a policy change, such as restructuring of the company, may indirectly affects the current task of the employee. Also, if the wife of the employee makes a phone call to him, the employee's current task may be affected too.

Composition of Agents

Social learning is a group activity. Individual differences between members, including ability, gender, acquaintance (whether the members know each other), learning styles, sight (whether they can see each other), etc., are sometimes critical to learning performance. For example, if a student does not like competition, we cannot expect good performance from the student in a competitive environment. In general, members can be homogeneous or heterogeneous, that is, have similar or dissimilar characteristics. It is an interesting issue, though not easy, to find out what is the best composition of members for different social learning environments. If this is known, for human agents, we can choose the right members to form learning groups; and for computer agents, we can adjust the characteristics of the simulated learning companions to fit the need of the group.



1 Real Site Companion
2 WO Companion
3 Pretend Companion
4 Simulated Companion

Figure 10. Different Views of Agents

Number of Agents

In distributed systems, the number of agents relates to the number of sites and thus has to be decided before building the system. Studies indicate that the number of agents should be a few, say, two or three, for collaboration (Hooper, 1992). Apparently, few members can ensure more intensive interaction during collaboration. For a larger group, instead of collaboration among individuals, we may have collaboration among sub-groups, that is, each sub-group forms a unit and is responsible for a particular sub-task. For competition, most students in Contest-Kid prefer small group competition.

Protocols

Social learning is a kind of social activity and protocols of learning activity are the rules of game to govern such activity. These different protocols of learning activity can be roughly classified as collaboration, competition, learning by teaching, and role playing game.

Collaboration

Collaboration perhaps is the most important and by far the most common form of social learning. Forms of collaboration can vary in the following dimensions: *responsibility sharing* vs. *total task; tightly coupled* vs. *closely coupled; single group* vs. *multiple groups*, etc. Responsibility sharing is one of the collaboration forms applied in Integration-Kid where one agent is responsible for making decision and the other for executing the decision, then these roles are switched for alternative problems. The following state transition like diagram (Figure 11) gives the details of the interaction protocol where the simulated learning companion decides what to do next and that the student executes the decision by the companion.

In responsibility sharing, each group member is responsible and takes turns for working on a sub-task. Alternative to responsibility sharing is the case where every member is responsible for completing the task. TurtleGraph is an example system where both students are responsible for the complete solution (total task) of the problem.

The distributed version of TurtleGraph is loosely coupled since both students try to work out the solution independently first and only communicate when they need external help. A significant part of their communication is coordination rather than negotiation about the solution of the problem. In contrast is the centralized version where communication is intensive through natural conversation and this is regarded as tightly coupled collaboration. Distributed systems can support tightly coupled collaboration if the collaboration includes responsibility sharing and is able to reduce coordination activity during communication. There are some other forms of collaboration which are loosely coupled, for example, open discussion such as Why system (Stevens, 1982), People Power, or dialog game (Pilkington et al., 1993).

Apart from collaboration between individual members in a single group, there could be collaboration between groups. For example, if the collaboration is responsibility sharing and the number of individual members is more than the number of sub-tasks decomposed, then sub-groups can be formed. The collaboration will be two level: individuals collaborate within a sub-group (intra-sub-group collaboration) while the sub-group collaborates with other sub-groups (inter-sub-group collaboration).

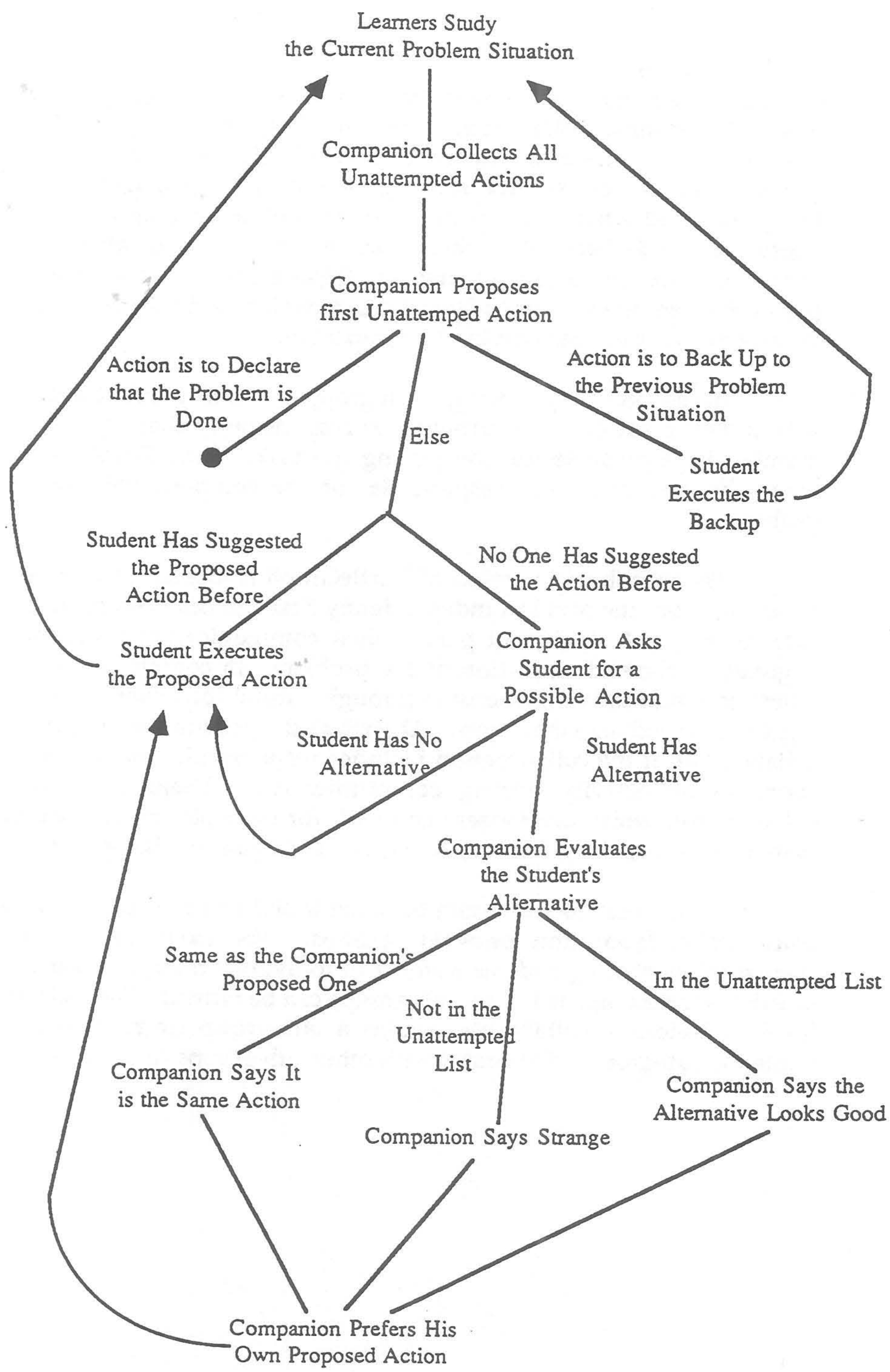


Figure 11. Computer Companion Deciding while Student Executing

Competition

Competition is a comparison process and is a strong motivator; yet, not all students like competition (see discussion on Contest-Kid above). In human competition, we may have to face the problem that there may exist a population of students who are always less competent than the rest. To lessen the negative effects due to competition, like many computer games, we may have a student who works against a set of computer companions (e.g. Three's Company) or we allow him to have more than one attempt. In general, there are some strategies that can be employed in competitive learning environments to achieve this. For example, give the students an excuse for their failure, encourage students that paying effort will assure better performance, or even fool the students (see pretend students in Contest-Kid). A centralized competitive learning system may use a score board to record performance of previous players, such indirect competition helps a student to set his or her own goal and be more ready to pay effort to reach the goal.

Collaboration and competition always emerge as a mixed form in a society. For example, members of a commercial firm will cooperate in order to beat their opponent. In many team sports, such intra-group collaboration with inter-group competition has proven to be very successful. In fact, social learning activity of such hybrid form can both achieve human affinity due to collaboration and raise motivation due to competition.

Learning by Teaching

When an agent's knowledge level of the domain is higher than that of the student, then the agent can be a tutor of the student. If the agent is computer simulated, then we have the usual ITS model. When the agent's knowledge level is about the same of the student, then they may be learning companions to each other, collaborating or competing. Conversely, when the agent's knowledge level is lower than that of the student, then the student can act as the tutor of the agent. The last model, termed as learning by teaching or peer tutoring in educational studies, is not necessarily an instance of responsibility sharing for the agent who takes the tutor role knows the answer, thus does not undertake the learning task. Learning by teaching can promote the student's retrospection, generalization, and generation of self-explanation.

Role Playing Games

Role playing games (RPG) are a socially situated learning environment. The student plays a role in the scene and interacts with the physical world and other agents. These agents take the roles that may or may not work on the same task of the student. Being a simulation of the real world situation, RPG is appropriate for many domains – simulating a workplace, RPG can be used for job training that needs decision making such as managers or hotel receptionists; simulating the environment of a university might help freshmen to adjust the new environment, simulating a historical situation might help to better understand the characters and the positions of the historical figures, and so on.

Outcomes

Different protocols of social learning activity or structure would raise different issues and lead to different learning outcomes or effects on students. For example, in responsibility sharing, the cognitive load shared by other learning companions means *scaffolding* the student. In collaborative problem solving, students encounter *cognitive conflicts* when multiple perspectives emerge. The resolution of these conflicts leads to re-construction of newly acquired knowledge and forms an integral part of the students' own knowledge. A learning companion can be viewed as a counterpart of the student while both of them possess similar knowledge. Thus watching how a companion works is a *reflection* to the student himself or herself. Social learning also seems to suit for meta-cognition. When students explain or defend their idea, the strategic knowledge of explanation makes the idea explicit. In learning by teaching, use of meta-knowledge and self-knowledge are salient because one has to know what one has and has not known as well as what one needs to know in order to be able to teach or give help.

Apart from cognition, interpersonal interactions in social learning have conspicuous effects on students' motivation, attitude, and sociability. Motivation is concerned with whether the student *will* learn, not whether he or she *can* learn. If learning is an activity that needs energy, then motivation deals with the amount of energy that one is willing to consume. Attention, confidence, and satisfaction are some motivational factors that have been identified critical in learning (Keller, 1983). Besides, social learning environments can nurture a correct attitude in students towards responsibility, error or failure, and effort. Social learning environments, if properly designed, can help students develop social skills as well as open-mindedness, such as a willingness to cooperate or to compete with others; conversely, if not properly designed, they can be harmful to students, such as hurting the students' self-esteem or discouraging them from seeking help.

However, we believe that evaluation of learning outcomes is a prominent difficulty in the study of social learning systems. For example, how should the various protocols of learning activities, in centralized or distributed environments, be evaluated and compared? More specifically, how should we evaluate the contributions from collaboration or competition, when compared to that from one-to-one tutoring? What evaluation methods are appropriate: formative evaluation, ethnographic study with protocol analysis, student-oriented summative evaluation, or combinations of these? How to evaluate the roles of the simulated learning companions in facilitating student learning? There will be no simple answer to these questions because cognition, meta-cognition, motivation, social skill, attitude, and so on, arise together and are indivisible in the process of social learning.

Another question is the bandwidth of the multimedia communication. How much of the available bandwidth is needed? Is it the case that more is better, such as seeing each other's face, sharing spaces for text or multimedia, and so on? The extreme case reduces to face-to-face interactions (e.g. two students sitting together and working on the same computer). But our preliminary study of TurtleGraph shows that this situation leads to qualitatively different learning behavior when compared to the limited communication through the network.

Techniques

Strictly speaking, there are no established special techniques for the development of social learning systems. From the architecture of Integration-Kid (Figure 12), we can see the usual components of ITS such as a problem solver, teaching strategies (reside inside the teacher agent), and an interface, which are all included in a social learning system. Integration-Kid does not have a sophisticated student model to understand the learning status of the student, although this component is usually important for the intelligent behavior of a social learning system. However, Integration-Kid supports modeling of multiple agents' behavior. Agents post their messages on the blackboard. Based on the messages, the agent scheduler administrates the interactions among different agents. The student agent mainly supports student's inputs such as an editor facility and a hypertext-like information sub-system is also accessible to the student. To be able to portray the behavior of the teacher and the companion agents, a problem solver, a syntax checker, and an utterance processor are needed.

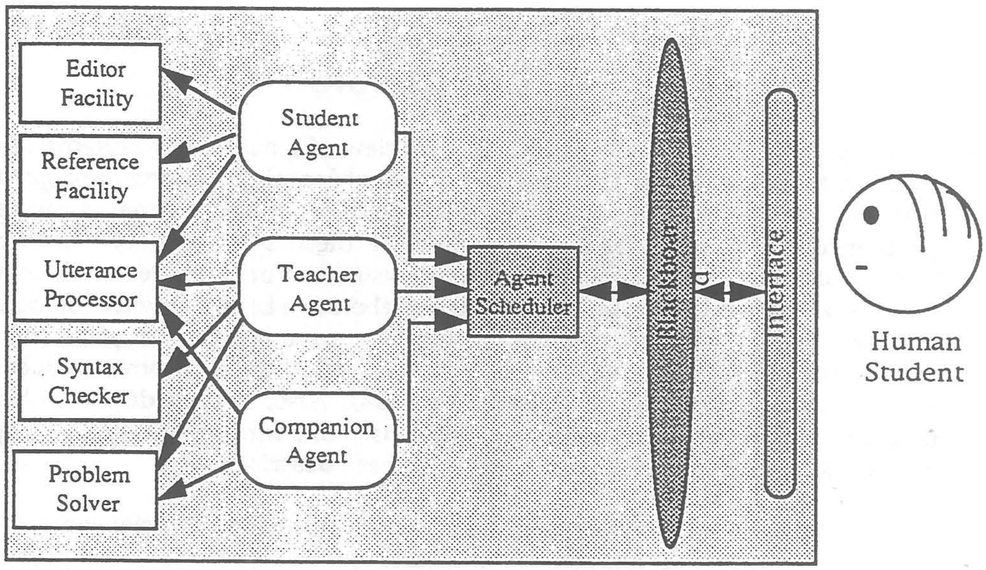


Figure 12. Architecture of Integration-Kid

As can be seen, the foremost part of the architecture is to model the multiple agents' knowledge and their interactions, whether these agents are computer simulated or real human beings. This research is closely related to the work of distributed artificial intelligence. Recent work by Shoham (1993) on agent-oriented programming summarizes some implementation details of modeling multiple agents in a system as well as a formal approach of modeling the agents' interactions. Central to modeling their interactions is the need of the *group student model*. In responsibility sharing, various issues remain to be investigated, for example, how to model a particular agent's cognitive process, how the agent relates his or her own solution of a sub-task to the complete solution, and how the system maintains these evolving student models. Furthermore, student modeling in social learning should go beyond cognition; other desired outcomes such as meta-cognition, motivation, attitude, etc., should be captured and achieved.

If an agent is a simulated companion, modeling its ability is a challenging issue. Integration-Kid adopts the straight forward *simulation approach*. The companion accumulates domain knowledge as learning proceeds. Parts of the companion's knowledge are erroneous. Through the conflicts in communicating with the student and the teacher, this erroneous knowledge is replaced by correct knowledge. Despite the erroneous part of knowledge, the companion's knowledge is always a subset of the teacher's knowledge. Another approach is to adopt AI machine learning techniques so that the learning companion can really *learn* through the learning process together with the student. The final issue, but not the least, is the modeling of verbal interactions among agents which is handled by the different agents and the utterance processor in Integration-Kid. This dialogue modeling, in particular, negotiation and argumentation, has been a subject of growing body of research (Moyse & Elsom-Cook, 1992; Baker, 1993).

CONCLUSION

In this chapter, a brief background, an overview of a number of systems, and a simple framework of designing and classifying social learning systems, are given. These systems, as can be seen, are rather different from the normal tutoring systems. Furthermore, the varieties demonstrated by these systems only represent a small portion of the spectrum of social learning systems. For example, a sub-area of CSCW which is known as Computer Supported Collaborative Learning where students work on a network or in a virtual classroom, with the same application program, have real-time communications or video-conferencing, and share a common database of their project work, has not been discussed (Pea, 1994). Also, systems discussed here are real-time and synchronous, asynchronous systems -- like many electronic mail applications or the Internet school kids projects, have not been described.

Despite emerging learning theories with emphasis on social context in learning, a generative theory is needed for analyzing and comparing the effects of different social learning protocols. Besides, it is also desirable to have an elaborate design methodology of social learning system development that can generate productive research result and ensure the systems usable in the real world. Research on social learning systems is still in its infancy. It is our hope that this tutorial would serve to stimulate more research on this area. Indeed, the recent advances in high performance computing and communications offer opportunity to create new breeds of social learning environments that may bring significant impact to education as we enter the next century.

References

Aronson, E., Blaney, N., Stephan, C., Sikes, J., & Snapp, M. (1978). *The jigsaw classroom.* Beverly Hills, CA: Sage Publications.

Baker, M.J. (1993). Dialogic Learning: Negotiation and Argumentation as Mediating Mechanisms, invited talk, *The Proceedings of World Conference on Artificial Intelligence in Education*, Edinburgh, Scotland, 4-11.

Bednar, A.K., Cunningham, D., Duffy, T. M., & Perry, J. D. (1991). Theory into Practice: How Do We Link? In G. Anglin (Ed.), *Instructional Technology: Past, Present, and Future*, Denver, CO: Libraries Unlimited.

Bhuiyan, S., Greer, J.E., & McCalla, G.I. (1992). Learning Recursion Through the Use of a Mental Model-Based Programming Environment, The 2nd International Conference of Intelligent Tutoring Systems, *Lecture Notes in Computer Science*, 608, Springer-Verlag, 50-57.

Bloom, B.S. (1984). The 2 Sigma Problem: the Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, Vol. 13, 4-16.

Brown, J.S., Collins, A., & Duguid, P. (1989) Situated Cognition and the Cultural of Learning. *Education Researcher*, 18(1), 32-42.

Burton, R. & Brown, J.S. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, Vol. 11, 5-24.

Bull, S., Pain, H. & Brna P. (1993). Student Modelling in an Intelligent Computer Assisted Language Learning System: The Issues of Language Transfer and Learning Strategies, *The Proceedings of International Conference on Computers in Education*, Taiwan, 121-126.

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. (1991). Integration-Kid: A Learning Companion System. *The Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, Morgan Kaufmann Publishers, Inc., 1094-1099.

Chan, T.W. (1992) Curriculum Tree: A Knowledge-Based Architecture for Intelligent Tutoring Systems, The 2nd International Conference of Intelligent Tutoring Systems, C. Frasson, G. Gauthier & G. McCalla (Eds.). *Lecture Notes in Computer Science*, 608 Springer-Verlag, 140-147.

Chan, T.W. (1993). A Multimedia Social Learning System. *The Proceedings of World Conference on Educational Multimedia and Hypermedia*, Orlando, Florida, USA, 99-106.

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*, 1988, 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., Lin, C.C., Lin, S.J. & Kuo, H.C. (1993). OCTR: A Model of Learning Stages. In S. Ohlsson, P. Brna, and H. Pain (Eds.),*The Proceedings of the World Conference on Artificial Intelligence in Education,* Edinburgh, Scotland, 257-264.

Chan, T.W. & Wang W.C. (1992). Problem-Solving Model Approach for Teaching Multiple Introductory Programming Languages. *The Proceedings of East-West Conference on Emerging Computer Technologies in Education*, Moscow, 51-56.

Cheng, Y.S. (1993). Rescue: A Role Playing Game for Learning Object-Oriented Programming Concepts, Unpublished Manuscript, Software Research Center, National Central University.

Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P. & Glaser, R. (1989). Self-Explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 5*, 121-152.

Clancey, W.J., Bennett, J.S., & Cohen, P.R. (1982). Applications-oriented AI research: education. In A. Barr, & E. A. Feigenbaum (Ed.), *The Handbook of Artificial Intelligence*, Reading, MA: Addison-Wesley.

Clancey, W.J. (1992). Guidon-Manage Revisited: A Socio-Technical Systems Approach. Invited Talk, The 2nd International Conference of Intelligent Tutoring Systems, *Lecture Notes in Computer Science*, 608, Springer-Verlag, 21-36.

Collins, A., Brown, J.S., & Newman, S.E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Collis, B. (1993). Cooperative learning and CSCW: research perspectives for inter-networked educational environments, Invited paper for IFIP Working Group 3.3, Working Conference "Lessons from learning", theme B, September, Archamps, France.

Cumming, G., & Self, J. (1990). Collaborative intelligent educational systems: Identifying and decoupling the conversational levels. *Instructional Science*, 19, 11-27.

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.

Doise, W., Mugny G., & Perret-Clermont A. (1975). Social interaction and the development of cognitive operations. *European Journal of Social Psychology* , Vol. 5, No. 3, 367 - 383.

Gilmore, D. & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self, (Ed.) *Artificial Intelligence and Human Learning, Intelligent computer-assisted instruction*, New York: Chapman and Hall, 179-196.

Hoppe, H.U., Baloian, N. & Zhao, J. (1993). Computer Support for Teacher-centered Classroom Interaction, *The Proceedings of International Conference on Computers in Education*, Taiwan, 211-217.

Hopper, S. (1992). Cooperative Learning and Computer-Based Instruction, *Educational Technology: Research and Development.*, Vol.40, No.3, 21-38.

Jehng, J.C., Shih, Y.F., Liang, S. & Chan, T.W. (1994) TurtleGraph: A Computer Supported Cooperative Learning Environment, *The Proceedings of the World Conference on Educational Multimedia and Hypermedia*, Vancouver, Canada, AACE, 293-298.

Johnson, D. W., & Johnson, R. T. (1991). *Learning together and alone.* Englewood Cliffs, NJ: Prentice-Hall.

Jonassen, D.H. (1992). Objectivism Vs. Constructism: Do We Need a New Philosophical Paradigm? *Educational Technology: Research and Development*. Vol. 39, No.3, 5-14.

Keller, J.M. (1983). Motivational Design of Instruction. In C.M. Reigeluth (Ed.) *Instructional Design Theories and Models: An Overview of the Current Status*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Lai, J.A. (1994). Contest-Kid: A Distributed Competitive Learning Environment. Master Thesis, Institute of Computer Science and Electronic Engineering, National Central University, Taiwan.

Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

Liang, S. (1994). TurtleGraph: A Distributed Collaborative Learning Environment. Master Thesis, Institute of Computer Science and Electronic Engineering, National Central University, Taiwan.

Lieberman, H. (1986). Using prototypical object to implement shared behavior in object-oriented system. *Proceedings of the First ACM Conference on Object-Oriented Programming System, Languages, and Applications*, USA.

Lin, G.L. (1993). Three's Company: A Social Learning System. Master Thesis, Institute of Computer Science and Electronic Engineering, National Central University, Taiwan.

Lin, S.J. & Chou, C.Y. (1993). GLASS: Learning by Observing and Discussing, *The Proceedings of International Conference on Computers in Education*, Taiwan, 218-223.

Liskov, B., Atkinson, R. & Schaffert, C. (1977). Abstraction mechanisms in CLU. *Communications of the ACM*, 20 (8), 564-576.

Looi, C.K. (1994). Private Communication.

McManus, M. M., & Aiken, R. M. (1993). The group leader paradigm in an intelligent collaborative learning system. In S. Ohlsson, P. Brna, and H. Pain (Eds.), *Proceedings of the World Conference on Artificial Intelligence in Education*. Charlottesville, VA: Association for the Advancement of Computing in Education, 249-256.

Moyse, R. & Elsom-Cook, M. (1992). *Knowledge Negotiation*. London: Academic Press.

Mugny, G., & Doise W. (1978). Socio-cognitive conflict and structure of individual and collective performances. *European Journal of Social Psychology* , Vol. 8, 181-192.

Nichols, D.M. (1993). Intelligent Student Systems: Learning *The Proceedings of World Conference on Artificial Intelligence in Education*, 576.

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.

Pea, R.D. (1994). Seeing What We Build Together: Distributed Multimedia Learning Environments for Transformative Communications, *The Journal of Learning Sciences*, 3(3), 285-299.

Piaget, J. (1965). *The moral judgment of the child*. New York: The Free Press.

Pilkington, R.M., Harley, J.R., Hintze, D. and Moore, D. (1993). Learning to argue and arguing to learn. *Journal of Artificial Intelligence in Education*, 3, 275-295.

Roschelle, J. (1992). Learning by collaborating: Convergent conceptual change. *Journal of the Learning Sciences*, 2(3), 235-276.

Self, J. (1985). A perspective on intelligent computer-assisted learning. *Journal of Computer Assisted Learning*, Vol. 1, 159-166.
Self, J. (1986). The application of machine learning to student modelling. *Instructional Science*, Vol. 14, 327-338.
Self, J. (1988). Bypassing the intractable problem of student modelling. *International Conference of Intelligent Tutoring Systems*, Montreal, Canada, 18-24.
Shoham, Y. (1993). Agent-Oriented Programming. *Artificial Intelligence*, 60, 51-92.
Sleeman, D.H., & Brown, J.S. (1982). *Intelligent Tutoring Systems*. London: Academic Press.
Stevens, A., Collins, A., & Goldin, S.E. (1982). Misconceptions in Student's Understanding, *Intelligent Tutoring Systems*. In Sleeman D. H., & Brown, J. S. (Eds.), London: Academic Press.
Twidale, M. (1993). Redressing the Balance: The Advantages of Informal Evaluation Techniques for Intelligent Learning Environments, *Journal of Artificial Intelligence in Education*, 4 (2/3), 155-178.
VanLehn, K., Ohlsson, S. & Nason, R. (in press). Applications of Simulated Students: An Exploration, Journal of Artificial Intelligence in Education.
Vygotsky, L. (1978). *Mind in society* . Translated by M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Cambridge, MA: Harvard University Press.
Winogard, T. & Flores, F. (1986). Understanding Computers and Cognition: A New Foundation for Design. Norwood, NJ: Ablex.

Acknowledgements: This research was supported by National Science Council of Taiwan, R.O.C. The author gratefully thanks the useful comments from Chi-Kit Looi, Margaret McManus, and Wing-Kwong Wong and the support of the colleagues of the LISA project: Jihn-Chang Jehng, George Chou, Yao-Sheng Cheng, Jong-An Lai, and Steven Liang.