Jl. Of Artificial Intelligence in Education (1996) 7(2), 125-159

Learning Companion Systems, Social Learning Systems, and the Global Social Learning Club

TAK-WAI CHAN
Department of Computer Science and Information Engineering
National Central University, Chung-Li
Taiwan, 32054 R.O.C.

This talk reviews our work in the past and presents a vision of future research based on our current approach. I take a rather personal perspective in describing how the idea of the learning companions emerged. Then I discuss how, after the development of a series of experimental systems over some years, the idea of the learning companions was extended to a class of systems which we call social learning systems. Failures or successes of some of these systems are discussed. After that, I describe the recent readjustment of our approach and take a leap to aim at a global social learning club. It is becoming clear that such a virtual learning community will be typical or prevalent in the next century.

LEARNING COMPANION SYSTEMS (A STORY, 1982-1989)

My bachelor degree was in mathematics, pure mathematics. I chose that subject because it is a fascinating subject. However, despite its beauty and elegance, 1 gradually realized that I would have been confined to the ivory towers of academia if I had continued my scholarly pursuits there. In my final undergraduate year, I decided to switch my focus to another area in my graduate study which would bear a greater relevance to human life. However, when I became a graduate student of computer science in 1982, I found that the professors who were willing to give me a research assistant- ship were those who did theoretical research and were interested in me solely because of my background in mathematics. It was a frustrating experience for me because on one hand, the research topics were of little practical importance; on the other hand, I had difficulties in adapting my thinking patterns to learning less theoretical computer science courses. I still remember the times during this period when I had dinner alone in a fast food restaurant and I watched the people outside on the street and wondered whether my work would relate to them in any meaningful way. I decided to take a year off from school and went back to my home town, Hong Kong, and taught mathematics in a high school.

When I resumed my studies, I was able to choose a more practical area for my research topic. My choice was relational database. I did my master thesis on the subject. I noticed that while there were many people talking about relational databases at that time in the mid eighties, many important research problems were already explored or solved and some major players had begun to switch their attention to object-oriented databases, an area too new for me to get sources of information. Fortunately, I had a new lover, namely, artificial intelligence. This is a subject with a clear intention to benefit humans through technology. I attended classes in artificial intelligence, logic programming, and machine learning. At the same time, I read literature of the early work of intelligent tutoring systems (ITSs) and sat in a computer aided instruction course offered by the education school. I was thinking about the possibility of a research topic on applying artificial intelligence to education. The machine learning course offered by Michalski was very interesting. He never mentioned any algorithm in detail for those systems he designed. He encouraged us to observe how learning happens in daily life and look for new ways to model learning. I remember some researchers in machine learning mentioned that education is a potential application area of machine learning, but never said how. I began to think about the possibility of applying machine learning to ITSs.

There was an incongruity there. While ITSs assume that there is a smart machine to teach a student who is to learn, machine learning, however, assumes that the machine is not that smart and demands the machine to learn. A natural way to resolve this paradox is by having the computer simultaneously represent the two explicit artificial agents, one as a tutor and the other as a learning companion. The learning companion may use machine learning to learn and interact with the human student. This was the idea, a very simple idea. The notion of the learning companion was just an accidental outcome of an effort to provide a reasonable framework for applying machine learning to ITSs, not inspired by the Chinese proverb "studying with the Prince," as mentioned in the original paper. A prominent researcher in the field told me that the learning companion idea drew considerable attention because people were getting bored by the work of applying expert system technology to ITSs. Nevertheless, the fundamental assumption of existing -wisdom—the computer as an intelligent tutor—was being challenged and extended, and a paradigm shift of ITSs research towards social issues of cognition had begun.

After we published the idea (Chan & Baskin, 1988, 1990), we discovered several interesting things.

1. I found that Self and his colleagues' idea of treating the computer as a collaborator (Self, 1986; Gilmore & Self, 1988); Palthepu, Greer, and McCalla's (1991) idea of learning by teaching; and perhaps the exploration of the use of simulated students by VanLehn, Ohlsson, and Nason (1994) were somehow triggered by the attempt of applying machine learning to ITSs. This really intrigued me and perhaps many of you, too.
2. My advisor discouraged me from applying machine learning methods to implement the learning companion in Integration-Kid (Chan, 1991), the first prototype of the learning companion system (LCS) in my doctoral thesis. He reasoned that machine learning techniques were not mature enough to meet the demands of the learning ability of the learning companion (though I found the attempt intriguing). Instead, I used a simpler approach-simulation-where the learning performance of the learning companion was governed by a subset of problem solving expertise and some faulty knowledge. The problem-solving expertise of the learning companion was expanded and the faulty knowledge was tuned by simply deleting and adding knowledge units.

Indeed, if the student does not care how the learning companion learns, whatever approach we adopt to make the learning companion acquire knowledge is of no difference.

3. Another decision to make was whether to use the student model in Integration-Kid. During that time, the practical and theoretical difficulties of building an extensive student model were revealed (Self, 1988; Newman, 1989). Therefore, I decided that Integration-Kid would not support explicit student modeling in the sense that it does not contain an explicit representation of the student's knowledge induced from the student's behavior. The learning companion, however, approximates a typical student and can be viewed as a form of *active* student model. When given a specific problem solving situation within a particular protocol of learning activity, the companion tries to share similar views and find a sub-optimal solution like the student by linking its responses to that situation. This sub-optimal solution might contain an error performed by the student. The learning companion reflects a knowledge image that is modeled like the student's own knowledge. In such a way, the current knowledge of the companion acts as if it were another student model but for the companion. Note that this description is essentially along the line of a suggestion by Self (1988)—the student modeling difficulty could be relieved if the student works in collaboration with the computer instead of being tutored by the computer. I was happy with this position until recently. We found that, in Reciprocal-Tutoring-Kids (to be discussed later), if the learning companion plays the role of a tutee and is to be tutored by the student tutor, a student model is necessary for modeling the behavior of the learning companion.
4. Once the technical difficulties involved in the implementation of the learning companion were solved, I had more time to explore a range of possible learning activities at different learning stages among the three agents in the LCS environment. Interactions among agents were described by some finite state graphs. In particular, I was interested in architecture and tools that could be used to support interactions among agents and subsequently developed the Curriculum-Tree (Chan, 1992).
5. When the prototype was finished, I found it difficult to get people to test the system. Only my wife's friend and a high school student used the system. In fact, to date, there has been no serious evaluation of the learning effectiveness of having such a learning companion in a learning environment.
6. The most pleasant experience perhaps was that the learning companion idea was well accepted by members of my thesis committee. Although many of them were not experts in the area, I'm grateful for their acceptance of the idea of my thesis and, of course, my doctoral degree!

That is the story. Now, looking back, I find that having the will to change my research area might not be enough because the external environment and my inflexibility in learning dynamics might block my will. Also, although most computer scientists look very busy, some areas of computer science are rather narrowly defined. Relational database research is an example. It basically focuses on a particular data model. But at the outset hundreds of researchers began working on the problem domain and many significant problems were quickly explored. By contrast, in this field, there are many subject domains to learn and people need to catch up with rapidly emerging new knowledge and tools or techniques to facilitate learning are unlimited and ways to learn vary. Therefore, the research space of this area is much larger. Computers have been successfully applied to science, engineering, and commerce for years. It may be time for more computer scientists to take a look at the area of education.

SOCIAL LEARNING SYSTEMS (AN OVERVIEW AND REFLECTIONS, 1990-1994)

Like many of you, as a faculty member of a university, I began to extend my doctoral research work. At first, my research followed two lines: One was the further exploration of different models of LCSs; the second was the design of a programming language that could support the development of LCSs. Later on, a research group was formed and our work became even more diversified. I will now discuss our work during this period.

Extensions of Learning Companion Systems (1990-1992)

Distributed West. Distributed West is a distributed system (Chan et al., 1992). The first version runs on two PCs via R232 and the second simpler version runs on two Power Macs via AppleTalk. Distributed West is a reimplementation of a CAI and the coaching system classic, West, where a student plays against the computer opponent in trying to reach the destination before the opponent does. An agent (the student or the computer opponent) receives three randomly generated numbers at each turn. The student forms an arithmetic expression with these numbers using a set of arithmetic operators. Then the system evaluates this expression to a number and moves the icon representing the student towards the destination, with the number of steps equal to the value of the expression. The goal of the original West system is to engage and motivate students to practice arithmetic calculation using a game environment. In Distributed West, instead of a computer opponent, real students (usually two) at one site play against the other students at the other end of the network. Besides the human competition mode, students at the two ends can also collaborate to work against the computer opponent. Two interesting issues were raised in Distributed West. First, there are a tremendous number of possible variations of such environments that may be adopted to influence the student's motivation and performance in learning, making it difficult to decide which environment will be the most suitable for students. Second, evaluation shows that 3 out of 4 students prefer playing against each other in teams to cooperating together on two sides against the computer opponent.

While the West game has a very[1] special design that is not easily generalized to other domains, competition can in general be applied or embedded as a part of the learning activity in many domains. For example, the use of a score board may be the most typical method of promoting competition. It is interesting to note that score board encourages users' effort, but not their ability. Effort can increase the chance to win a position on the score board but the number of attempts will not be recorded. However, despite being a strong motivator, competition should be carefully designed to avoid harmful effects to students such as making them look "dumb" in the comparison process of the competition game. In general, West is considered to be a successful system since students enjoy the game and it is very popular during public demonstrations of our laboratory.

CAROL. Designing AI programming languages is intriguing since whenever important AI concepts become mature, the related programming languages emerge. For example, the development of expert system shells capitalizes on the results of knowledge representation of frame systems and rule-based systems. Our design of CAROL is the continuation of my interest in developing an authoring architecture, Curriculum-Tree, in the implementation of Integration-Kid. We initially intended to design a high level AI language. Later, after we noticed that the simple data model of Curriculum-Tree could be embedded in the language, we changed our design objective to support easy construction of multiple agents.

CAROL has been evolving in five versions. CAROL 1 is an initial attempt to integrate frame, rule, and object-oriented representations. CAROL! focuses on the rule representation and incorporates the searching mechanism of Prolog. CAROL3 is a fully fledged programming language obtained by extending CAROL2. The body of a procedure must be a set of rules with their behavior being governed by some external control. CAR- OL4 significantly simplifies CAROL3 by employing the data model of Curriculum-Tree which is essentially a prototype or classless data model. Because of the simplicity of CAROL4, its role in object-oriented languages is, arguably, the same as the role of Scheme in functional languages. CAROLS extends CAROL4 by designating some objects in CAROL4 to be agents that function continuously through responding to external events they are interested in. CAROL4 allows objects to make promises (A “promise” is a language construct that supports the behavior of an agent to have some weak autonomy. It is a delayed conditional action or a rule waiting to be fired in an agent. Promises can be dynamically generated in CAROLS). CAROL4 has been fully implemented in C++ with a real time garbage collector we developed. There is an on-going effort to develop a virtual machine of the language so that it can run more efficiently (CAROL4 will be released in early 1996). Both CAROL4 and CAROL5 will be used in the development of our subsequent systems, especially during the development of ‘virtual learning companions or other artificial agents. This work provides an example which shows that research into AIED sheds light on research in other seemingly different areas. This is no surprise since learning, problem solving, programming, or, the whole field of software, is in general a study of how human endeavor is supported by computing technology.

We noted that in the process of designing a programming language, we appreciated, from time to time, the elegance of the artifact we produced; but such satisfaction was rare in the process of designing learning companion systems. A programming language is suggestive of a model which can be used to solve problems at hand while a learning companion system only provides a model for the users to learn. A programming language serves as a “tool” for the users only, leaving all the details of the problem solving activities to the users. However, learning companion systems, or, in general, computer assisted learning systems, take a far more interactive role with the user and thus a more ambitious position. They try to supervise or deal with the dynamics of the conceptual or knowledge changes of the users. Unfortunately, these responsibilities of computer assisted learning systems can be enormous and simplifications are typically made by tying the system response to a specific content area of the user input.

Pedagogical languages. This is a coherent set of highly simplified programming languages implemented in Lisp and Scheme on both PC and Mac platforms. I have been using them for teaching an *Introduction to Computer Science* course for several years. The set consists of six languages, Core, Block, Lisp, ADT, Prototype, and SOOP. Core is designed for beginners to learn programming. It consists of several basic elements that are common to most high level programming languages. Block can be viewed as a subset of Pascal when used to teach block structure via nested function declarations as well as procedural abstraction, structured programming, and top-down design. Lisp is a tiny subset of Common Lisp used to teach recursion and a large data type, namely, list. The last three languages are designed for teaching the concepts of the now prevalent object-oriented programming. ADT that imitates CLU is used to teach students Abstract Data Types. Prototype is a simplified language designed for introducing the concepts of prototypical objects, delegation, and message-passing. Finally, SOOP is a Small Object-Oriented Programming language that is a combination of ADT and Prototype.

LISA: A Group Project for Social Learning Systems

At the beginning of 1992, we conceived an idea for a larger project based on the following two reasons: The first reason is that LCSs should be "larger." Instead of using centralized systems like Integration-Kid, we seek distributed learning systems with networks, like Distributed West. Instead of just having virtual learning companions (VLCs), we also want the capability of real human learning companions. Instead of the sole use of cooperative learning, we adopt competition, peer tutoring, and so on. Thus, we have a larger scenario—social learning systems—which are environments where multiple participants, either computer simulations or real human agents, work at the same computer or across connected machines, taking various roles via a wide range of activity protocols.

The second reason is that we have designed a set of pedagogical programming languages. We may try different social learning activities for each of the programming languages as well as other general material in the "Introduction to Computer Science" course. We hoped that when we finish all the systems in a few years, they could be used for a complete course.

In 1993, a group project called LISA (Learning IS Active) was funded by the National Science Council of Taiwan and the National Central University, with about two dozen researchers and graduate students from different universities in Taiwan became involved. The project has acquired twenty six Mac Quadras and Power Macs to be used by a class of students and for software development. These computers have good resources of multimedia programs and are able to communicate with Lisp.

Below is an overview of six sub-projects. For more details, see (Chan, 1995b). There are a few more sub-projects, but they are neither significantly implemented for different reasons nor along the line of our discussion.

Learning Stage Model. Social learning does not provide a broad enough context for describing the whole process of learning a course for a subject. We sought a "complete" conceptual framework for the system and proposed a model, OCTR (Chan, et al., 1993) which consists of four stages: (a) Orientation (prior knowledge connection), where the system helps the students relate their prior knowledge of the domain to learn; (b)

Coaching (knowledge growth), where the teacher first models the task for the students, then help them handle the task on their own mainly by scaffolding and fading; (c) Tuning (knowledge articulation), where knowledge restructuring is exercised via peer interaction and the teacher plays a less active role; and (d) Routinization (knowledge solidification), where students solidify their knowledge by practicing repeatedly, perhaps under some form of peer pressure.

Learning under OCTR advances gradually as the amount of social activities increases. OCTR also serves as a framework for describing the cognitive development and motivation of students, categorizing many existing teaching strategies so that they might be used appropriately, and providing a better understanding of how to effectively use meta-cognition ideas and learning attitudes in learning systems. It can also serve as a bridge between traditional CAI, ITS, and learning companion systems. Traditional CAI seems to work best in the orientation and routinization stages while ITS concentrates more on one-on-one coaching. Social learning, on the other hand, is more appropriate in the tuning and routinization stages. OCTR may not be accurate enough to describe the entire repertoire of the learning stages. Nevertheless, it clearly illustrates that there are many dimensions of the factors we have to consider when designing environments for learning a whole course.

Three's Company and Glassroom. These two centralized systems (standalone computers) were developed on Mac Quadras and Power Macs. Both of them are direct extensions of the original centralized LCSs. Three's Company and Glassroom teach concepts of Lisp and ADT respectively. In Three's Company (Lin, 1993), instead of a single VLC, there are two. We hope that the different performance and persona of the two companions generate interesting cognitive and motivational issues. GLASS (Glassroom society dialogue model), is a more complex model (Lin & Chou, 1993). It consists of two parts: The inner world (glassroom) and the outer world. The outer world is an LCS model. The inner world is the subject to be observed and discussed by the outer world. Glassroom is a general term which may imply an object, an episode of video, or another small learning society. A daily life analog of this model can be found in the operating room in a teaching hospital where medical students outside the room observe the operation process through a glass window and listen to explanations by an instructor at some critical points in the process.

I consider these two systems as failures. The results of these systems cannot significantly go beyond the original simpler LCSs. Although the additional agents involved make the design more complicated, there was no evidence that the students' learning was enhanced as a result. Whether an agent should be added to a system depends on whether it can function significantly in the learning process, for example, sharing a critical portion of the cognitive load. Unfortunately, this is really domain dependent and requires depth analysis prior to the selection of agents. For learning the concepts of programming languages, it seems unnatural to design many virtual agents with different roles.

Rescue (A Role Playing Game). Rescue was implemented on Mac Quadra (Cheng, 1993) for learning Prototype, a small language. It is a role playing game (RPG) and the student plays the role called MacGyver with the task of rescuing the kidnapped President's daughter. MacGyver meets and talks to other roles (or people) and explores different environments in order to collect information to achieve the task. MacGyver communicates with other roles by constructing and sending messages for them with a menu bar. The message construction process is in effect calling a particular method of an object with a message sending to another object. The activity is designed so that the student may utilize object-oriented programming concepts such as object, state variable, method, delegation, and message passing. The objective is to enable students through playing the game to master some of these concepts and to appreciate that such concepts mimic how people model the real world. Students generally find that Rescue is fun to play but the effect of learning the concepts of Prototype with this system is not clear yet. One of the problems of RPGs is that they are rather labor intensive to implement and there are few authoring systems available.

Role playing games engage the user to interact with various roles in a scene. They are rather naturally situated social learning environments, especially for learning communications with people and decision making skills in various situations. Two undergraduate students are now working on another RPG for freshmen orientation. Using this program, students can learn information or stories about different places on campus. Furthermore, the program simulates four-year undergraduate life and encourages students to make decisions and plan about their studying and other activities. If a student keeps making poor decisions, he or she may find out that dropping out of the university will be a surprising and unfortunate outcome of choices. The system provides explanations before the game is over. Our long term objective with this project is to develop an effective form of campus-consulting program where, in a simulated world, students can make decisions involving collegiate activities and have a chance to visualize and explore possible consequences.

TurtleGraph. TurtleGraph (Jehng, et al., 1994) is a distributed collaborative system. It requires two Mac Quadras or Power Macs to run. Using TurtleGraph, two students engage in free discussion through the network during solving non-trivial recursive programming problems in LOGO. There is no student model to detect their difficulty and no natural language understanding capability to interpret their dialogues. Students can communicate at anytime by pressing the communication button and writing message in the dialogue box. The student may also see or copy the other student's partial solution. Usually, students attempt to solve the problem independently at first and only at a later stage, when difficulty arises, will one or both of them initiate discussion. Thus, discussion is a free invitation and students of similar ability tend to collaborate well. When students are of different abilities, a higher proportion of questions or requests are usually raised by the weaker student. The more capable student has to understand his or her partner's situation by reading the written messages of the partner before giving responses. Students usually illustrate some form of competitive behavior partly because they may compare their own partial solutions.

One of the problems we found with this type of system is that when both students can solve the problem, they seldom communicate. This is because the collaboration is optional and therefore discussions between the two students are not frequent. Another problem is that when both students had trouble, they were not necessarily able to solve the problem through discussion. This is an ongoing project and further investigation into ways to improve collaborative techniques is needed.

Contest-Kids. Contest-Kids (Chan & Lai, 1995), like Distributed West, is a distributed competitive social learning system. It connects two to seven handshaking-connected Mac Quadra or Power Mac computers. Learning proceeds in two phases: The learning phase and the testing phase. In the learning phase, individual students learn by watching a short video. In the testing phase, students are engaged in a competitive game and score points by correctly answering a series of questions. The system allows users to play against real or pretend companions of various knowledge levels depending on student needs. Pretend companions are virtual sites simulated by the server. I regard this system as a success since, despite its simplicity, students enjoy using it and improve their knowledge in the process. It is a rather general distributed social learning system for helping students learn factual information.

Reciprocal-Tutoring-Kids. Reciprocal-Tutoring-Kids (Chan et al., 1995; Chan & Chou, 1995) is a set of systems which can run either on a standalone or on two handshaking-connected Mac Quadras or Power Macs. The systems support a form of cooperative learning called reciprocal tutoring in solving Lisp recursive problems. During reciprocal tutoring, two agents, where an agent is either a real student or a VLC, interact and assume the roles of a tutor and a tutee alternately for different problems. Thus both students have the opportunities to learn by being a tutor or being a tutee.

This system represents an initial effort to study both learning by tutoring in network environments and also how a VLC can be used in the reciprocal tutoring paradigm. We can compare learning in reciprocal tutoring paradigm with individual learning or learning in an ITS environment. In individual problem solving, the student does various things: Constructing the solution, testing and diagnosing it, explaining possible error, and self-advising for correction or alternative. In ITS, diagnosis and advice are usually the responsibilities of the system. In Reciprocal-Tutoring-Kids, these two activities are performed by another student at the other end of the network. Because of the broad applicability of the reciprocal tutoring model, it will be the focus of our considerable future efforts.

Zone of Proximal Development with Respect to Virtual Learning Companion (Chan, 1995d)

As we conducted some human testing of distributed systems such as Distributed West, Contest-Kids, and Reciprocal-Tutoring-Kids, we were surprised that we successfully fooled some students by asking them to work with other human students; in fact, they were working with VLCs. This implies that, given a narrow bandwidth of communication, the modeling of a human student in the system is psychologically plausible or students believed what we told them simply because they trusted us. This observation triggers an interesting hypothesis related to Vygotsky's zone of proximal development.

In his work, *Mind in Society*, Vygotsky (1978) hypothesizes that social interactions play a fundamental role in shaping internal cognitive structures. Vygotsky's *zone of proximal development* (ZPD) is the 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. If we make the following weak Turing (WT)[1] assumption:

> Given a sufficiently narrow communication bandwidth, we can simulate, possibly based on data taken from students' use of the system, a VLC with behavior similar to an average human student using the system so that when the VLC sneaks in the system, a student cannot detect that his/her counterpart is a VLC.

Under this assumption, and if the VLC is adaptive, that is, the VLC can facilitate learning by reacting properly to the varying and changing needs of different students over time, then we may hypothesize that there exists an additional zone of proximal development with respect to the VLC (ZPD-VLC) which is defined as:

> The difference of the distance between the actual developmental level achieved by learning with an average human peer to the level of potential development as determined by learning with an adaptive VLC under the WT assumption (see Figure 1).

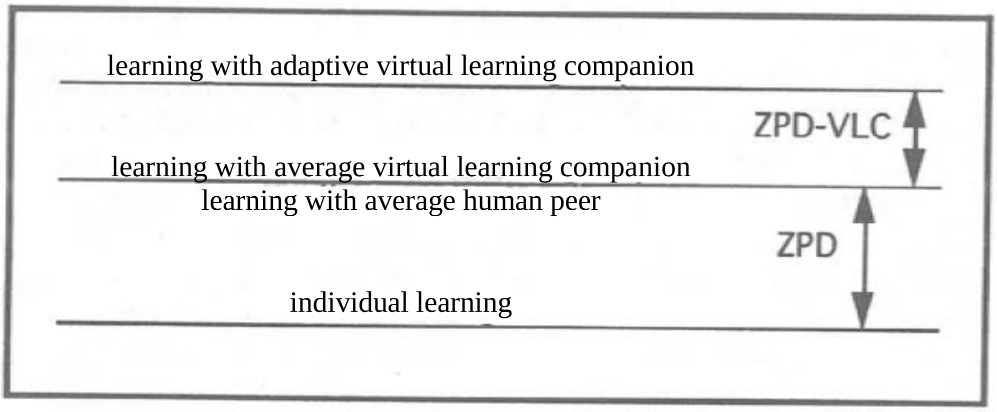

Figure 1. Zone of proximal development with respect to VLC (Chan, 1995d)

Note that we have generalized the original ZPD to an average peer, who sometimes behaves better or worse than an average student, since students also learn with less capable peers, though perhaps in qualitatively different ways. Under the WT assumption, an average human peer can then be replaced by an average VLC. The WT assumption is necessary for we can expect students' motivation would be different in interacting with a real or with a virtual peer. The difference, in turn, affects students' cognitive development in different ways.

The notion of ZPD-VLC mainly provides a conceptual goal for future research of network based VLCs. We do not intend to advocate the prevalent use of VLCs to fool students in the real world practice but merely as a research goal to explore ZPD-VLC. Having a VLC in the network can enhance the availability of synchronous learning since the VLC is always present. It is likely that some students will prefer to interact with VLCs before they communicate with their real peers.

Reflections

Sometime in 1993, when I read again our original paper on LCSs, I was stunned to discover that all what we had been doing were precisely described in its last section, and had not gone further:

> ...the paradigm of LCS represents a broad spectrum of ITS design due to the possible varieties on the number and the identities of the agents in an LCS. Each of these varieties gives rise to particular cognitive issues in the student's learning. First, it is possible to have no teacher involved....Then the student may observe how the companion solves the problems and improves performance. In this way, the student *learns how to learn by teaching the learning companion....* At the other extreme, it is possible to have multiple teachers with different persona ...The student may choose one of them to respond adapting to his own learning style. LCS may also be a simulation of peer group learning, which means more than one learning companion with different knowledge level or persona involved in the learning environment....Imagine in the near future as the price of computers falls and the technology of computer networks becomes more accessible, students can learn together through geographically distributed networks of computer....We believe that current LCS research is preparing for such a *futuristic intelligent computer classroom....*

Now, after these years of subsequent work, it is time to reflect:

- ***Too diversified.*** This is a salient problem in our approach during this period. One reason for the lack of constraint is that we did not have a good initial guess as to which model would have potential. Many times we found it difficult to go deeply enough to undiscover interesting issues. Later we discovered that when there are multiple agents, it is important that each agent shares a significant portion of the learning task. Models such as Three's Company and Glassroom are too complex for their intended domains. The applications of these models to learning programming concepts are an unnecessary and inefficient complication. However, some domains may require multiple agents. An example is reading if we use Palincsar and Brown's reciprocal teaching method (1984) since the task is divided into questioning, summarizing, clarifying, and predicting. So, an effective social learning model should match properly with the learning domain.

- ***Too many alternatives within a model in the design process.*** When we extend the one-to-one and the fixed roles of the two-agent ITS model to a multiple-agent model of LCS with roles to be determined in the design process, it is easy to have many alternatives. An example is the design of Distributed West where we enumerated 768 alternatives and we only implemented 3 of them. A good but difficult rationale is needed when facing these choices.

- ***Evaluation methods not effectively utilized.*** Formative evaluation was useful in the development process. However, having more students use the systems did not seem to be more helpful in gaining insights. This was partly due to our lack of experience in doing serious evaluation work. As competing choices arose, decisions made were based on formative evaluation, and more on our intuition. It was tempting to keep different versions and develop them in parallel. But we understand that this could be confusing, apart from being resource consuming. In other words, we should adopt a "depth first search," not a "breadth first search." Perhaps designing a Mercedes-Benz and a BMW in parallel will be several times harder than designing a single

Rolls-Royce. In the early 1980s, there were many word processors and only a very few of which survive in today's market. Are these survivors the ultimate word processors? Users do not seem to care about this issue and merely regard those processors as a set of presently viable options along with paper and pencil. In the future, perhaps, we need to develop more effective, efficient, and principled formative evaluation methods.

- ***Taking advantage of other good ideas.*** One invaluable experience we had was a cross-fertilization of ideas from other research. We do not need to create if we can find that existing knowledge can serve our purposes. What is important is to borrow' suitable knowledge and reuse it aptly in the context. For example, the excellent design of the classic West game borrowed in the Distributed West system contributed greatly to the students' enjoyment in using the system. Also, the Petal-like scaffolding tool (Bhuiyan, Greer, & McCalla, 1992) we used in the Reciprocal-Tutoring-Kids adopts a calculator metaphor. It is a powerful design and students using it can construct programs easily without making syntactic errors.

- ***Applying competition with care and faith.*** Many students find competition a powerful motivator for learning and enjoy competition although there is still a considerable proportion of students who do not like it. Competition is a straightforward way to review one's weaknesses through a comparison process. Thus, it is important for the system to provide the opportunity to improve one's perception of competence by making further effort. I am in favor of using competition in social learning environments but we have to be cautious to avoid its possible harmful effects. In general, there are various ways of applying competition. For example, students competing against each other will take different attitude from students collaborating together to compete against some artificial opponent. In the latter case, competition is just an ingredient to enhance collaboration. If the main feature of the system is competition, the design is usually quite simple and less intriguing than collaborative systems, unless we want to detect the student's psychological state during the competition process. In collaboration and other learning activities, however, we can consider to employ some mild competitive devices such as a score board to stimulate students' attention and effort, but, again, harmless.

- ***Different uses of student model.*** As with ITS, the student model remains a crucial component in some social learning systems, especially for some non-simple learning tasks. However, the student model can be used in different ways, other than hiding behind the student as an internal component inside the ITSs to tutor the human student. In Reciprocal-Tutoring-Kids, the student model is an important component in modeling a virtual tutee so that it appears to be learning and making mistakes while being tutored by the student. Furthermore, the student model in this system is represented as a tool that can be manipulated by the student tutor who diagnoses and gives hints to the computer tutee in a stand-alone computer or to the human tutee at the other end of two connected machines. Different ways and purposes of using the student model will become a trend.

- **Taking advantage of students' human intelligence.** We have been preoccupied with the thought of creating machine intelligence to help students learn for quite a while. In one scenario using Reciprocal- Tutoring-Kids, a student is at one end of two connected computers tutors and another student is at the other end. The student tutor can see the solution with its explanation and use that information for tutoring. This is an attempt to capitalize on students' human intelligence. We must admit that, given the state of the art of artificial intelligence, humans are much better natural language processors, understanders, learners, visual perceptors, and so forth, than machines. In designing social learning systems, we should bear in mind that machine and human intelligence should be complementary.

- **Failure to form an intelligent classroom.** The social learning systems we developed were for the class I taught, "Introduction to Computer Science." Apart from the need of evaluation of some systems, we note that we did not have a strong desire to urge students to use the systems as parts of their learning. We had no clear idea of how student use of the system might benefit our research. Furthermore, although we had tested the prototypes, we did not have enough confidence that they were really "usable" for students as parts of their regular learning. There were other reasons and constraints too. Despite a relatively high availability of machines, some machines for students to use were also designated for development, a conflict that we were reluctant to resolve. Management work such as arranging time and space, setting up computers, answering students' questions, and so on, was needed to make the systems available to students. Consequently, we were not sure whether the combination of this additional activity with the regular classroom teaching would be an undue burden or a necessary complement for students.

GLOBAL SOCIAL LEARNING CLUB (A VISION, 1995 - ...)

As seen from our reflections, even though we had extended our research to a wider spectrum of social learning systems, little progress was made in applying these systems in the real world. Starting this year, we readjusted our research direction and we forgot about our old idea of an intelligent *real* classroom.

Rationales for Change

We are now aligning our efforts to build a testbed for a global social learning club via the Internet. The first version of the system will be used this fall in three classes, one in our university in north Taiwan and the other two in the Yunlin Institute of Technology in central Taiwan. In the second version, probably in the next year, we shall seek cooperation with some overseas universities. There are two impetuses that cause change:

- The National Science Council of Taiwan has recently supported several universities for group research projects on distance learning research. Also, last year, Taiwan's National Information Infrastructure was launched and as a part of this project, a few universities use high speed network and ATM switches to demonstrate how learning could benefit from this technology. The outcomes of these projects are beyond the scope of our interest here. But I have a chance to examine whether our social learning systems research that uses handshaking networks, artificial intelligence, and multimedia, has kept up with the pace of technology change. Indeed, the advancement of technologies in this decade has been much faster than what I have been expected (things have happened much faster than in the years of my graduate study). How the information era will evolve in the next century is difficult to predict. Can our research in social learning systems meet the needs of the next century? As a researcher at this time, I have learned that if we cannot anticipate what will happen with technologies in the next 10 or 20 years, or even further, we are bound to an endless chase after technologies. This is a fact of life that we cannot ignore or opt out of.

* The soaring interest in using Internet or computer mediated communication (CMC) among different communities for delivering courses indicated that together with teacher monitoring, CMC is a viable alternative to traditional classroom teaching. Empirical studies by Hiltz and her colleagues (Hiltz, 1995) show that mastery' of course material in CMC classes will be equal or superior to that in the traditional classroom. Also, access to educational experience, access to the teacher, participation in a course, benefit from the course, attitudes toward the course, knowledge in the use of computers, and learning on the whole have been found to be improved.

The aims (provisional only since targets are always moving) of the current leap of the project LISA from an unsuccessful attempt of a real and intelligent classroom to a virtual global social learning club are two: First, to investigate and prepare a learning environment that may be pervasive in the next century by applying and developing artificial intelligence techniques, possibly incorporating them with other technologies. Second, to seek potential opportunities to achieve the Five Educational Goals. To attract more universities to join us to use our system, we have changed the subject domain to learning Scheme, a popular programming language for teaching introductory computer science course. I will now proceed to describe briefly the infrastructure of future learning environments (Chan, 1995a, 1995c), the Project A/B approach, and the five educational goals.

Infrastructure of Learning Environments

In terms of computer hardware and network connectivity, current computer-based learning environments can be roughly categorized into five types (see Figure 2): (a) stand-alone computers are natural environments for individual learning. Examples include one-on-one tutoring (ITS model) and learning with multiple virtual agents (LCS model); (b) handshaking- connected computer environments are not popular yet. They are a group of locally connected computers over a short distance, for example, using ApplyTalk in Macs or R232 in PCs. These networks are convenient for developing and testing prototypes that may be applicable to larger area networks; (c) wide area network and local area network are learning environments with mounting usage. On the Internet, there are a lot of tools and resources which are shared globally, and the population of users is growing rapidly all over the world; (d) integrated cable TV and computer networks will be substantially used by local communities before high speed networks become popular; (e) high speed networks will be rather expensive and there are still years before they can be widely available (the timetable for full implementation largely depends on the marketing of other services such as entertainment). More research is needed to obtain a clear picture of what is the benefit of using such wide bandwidth networks for educational purpose, when compared with face-to-face classroom teaching.

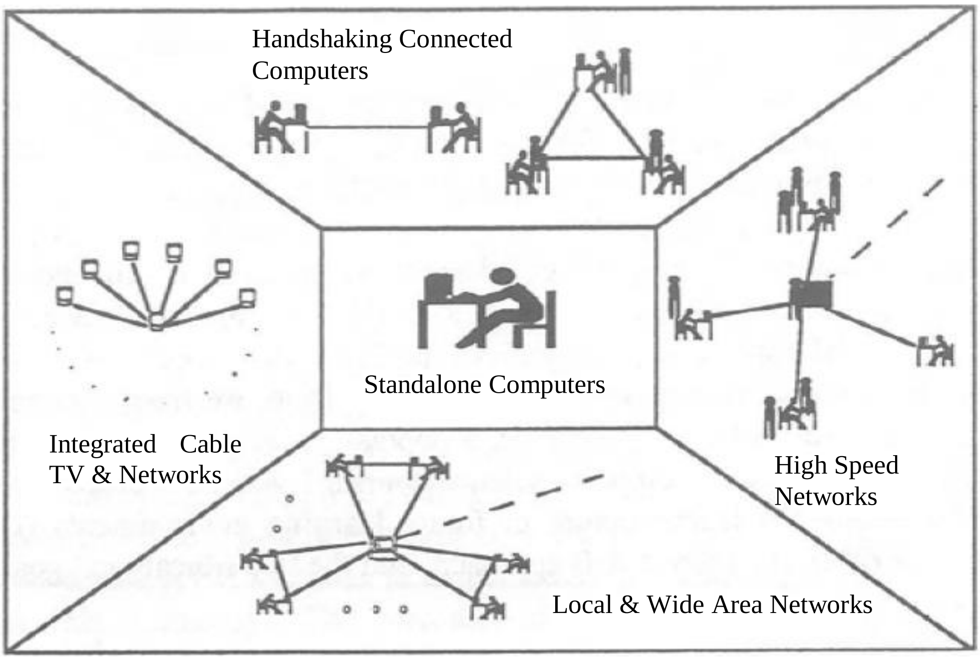

Figure 2. Types of networks (Chan, 1995a)

Now, if we consider the perspective of a learner who is at the center of the learning environment (cf. Barnard & Sandberg, 1993), the environment consists of the following set of utilities (see Figure 3):

1. Agents other than the student include human teachers, human teaching assistants, human learning companions or fellow students, virtual teachers, VLCs, or virtual personal assistants.
2. Cognitive tools are a set of tools that the student can manipulate. Students use them to learn, build, construct, and test answers or solutions, and analyze and reflect on what they have performed. Examples of such tools are text editors, concept maps, calculators, microworlds or simulation programs, and scaffolding tools that help the student by accomplishing a part of the learning task.
3. Learning material includes primary learning material that the learner has to learn, a communal database which is a collection of comments, questions, answers, and partial answers given by students, and secondary learning material which is accessible by the student such as dictionaries, encyclopedias, digital libraries, multimedia museums, large databases, and so on.

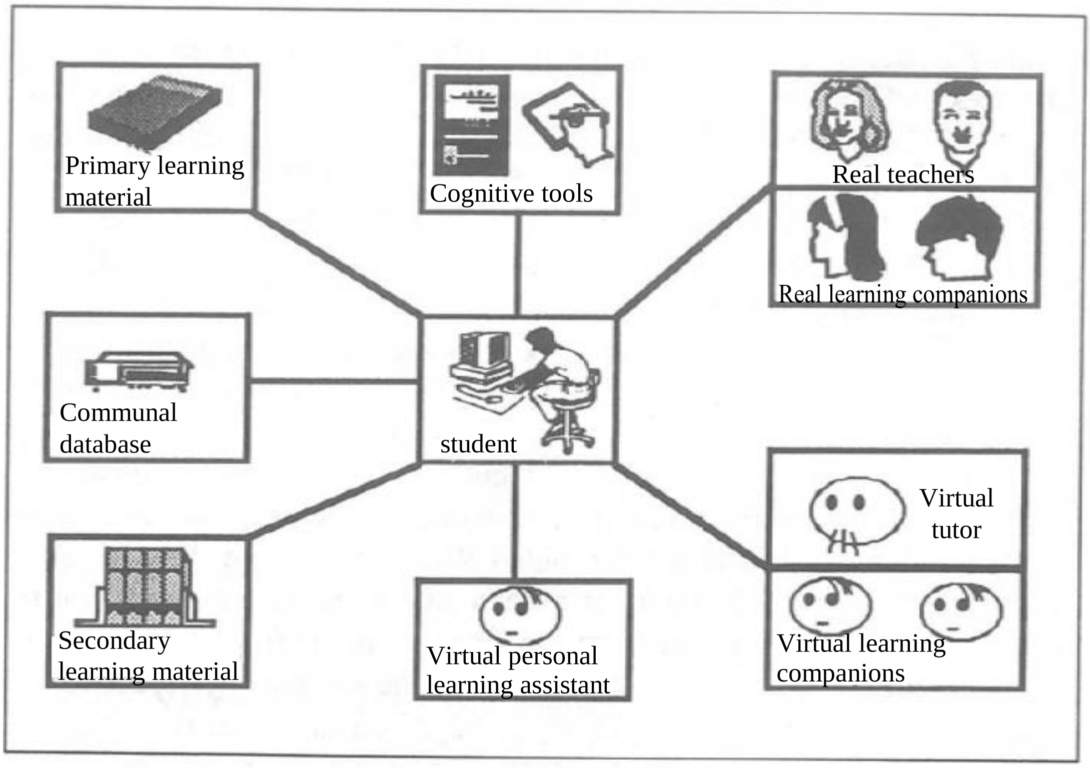

Figure 3. Student-Centered learning environment

Project A/B Approach

In describing our current efforts at this point, we found that it is helpful to separate our work as Project A and Project B. Project A is to utilize public network software to teach the *entire course* by delivering the primary learning material through the network and having students, teachers, teaching assistants, and researchers discuss the material and communicate. Project B is responsible for developing *niche* systems that focus on a particular content area of the course or aspect of the system to enhance the processes of learning, teaching, and other educational goals.

The first version of this global social learning club system will be implemented this fall using a Sun workstation as server and PCs as clients. Project A, like most other network courses on-line, uses the World Wide Web (WWW) to deliver the course material and information related to the "Introduction to Computer Science" course, incorporating a conferencing system on the WWW. The dialogues collected in the conferencing system will be a communal database. Furthermore, we can use a chat room for discussion and communication and provide an email grader to automatically grade the students' electronically submitted homework. This version of Project A allows students to access learning material easily and teachers to monitor students' progress through the network. Note that, as technology advances and wide bandwidth synchronous communications become feasible, it will be possible to run the whole course using Project A. However, the effectiveness of the result will still depend on the ability of teachers and teaching assistants to organize the course, monitor the process, and assist students. This capability enables us to retain some of the better qualities of the traditional classroom settings.

For Project B, we are currently porting the Reciprocal-Tutoring-Kids, Distributed West, and Contest-Kids systems to the Internet. Once students involved in Project B learn Lisp recursion, they can use Reciprocal-Tutoring-Kids to practice their skills. Students learning about the history of computers can play the Contest-Kids game. Students practicing binary number arithmetic can play Distributed West game. Note that, for all of these systems, if there is no other human student partner available on the network to participate in the learning activity, an artificial learning companion will be adopted for a partner. From the students' perspective, the systems will be analogous to playing "bridge" on the network. When you make a bid, you will wait for the others to respond to your bid before you take another action; but your partner could be a virtual agent and you may have an intelligent coach to help you.

For the Reciprocal-Tutoring-Kids system, a "fading" procedure is applied in four practicing sessions, with six problems in each session. In the first session, the student tutee will use a Petal-like scaffolding system to construct a program while the student tutor, at the other end of the network, watches the solution process using an AI component of the system to help diagnose the errors of the student tutee. This is the original Reciprocal-Tutoring-Kids system. In the next session, the student tutor can only see the solution and help the student tutee to diagnose the program and the tutee has only a manual of the language constructs as a reference and can key in the program solution. In the third session, the student tutor cannot see the solution and the student tutee does not have the manual. Finally, in the last session, all students must solve the last six problems using the interpreter.

We call the system a "club" because students can interact with each other freely in Project A and they can learn concepts and practice the skills in Project B whenever and wherever they want, and no penalty on their class grade if they don't participate. But we hope that students can develop a perception that their efforts in the involvement of these activities is positively correlated to their test scores. Also, systems in Project B possess some game elements. For example, besides maintaining the statistics of the scores of all the students, we keep a score board showing the 10 students with the best performance. If students want their records to make it to the board, they may have to practice with the system more than once.

While Project A lasts for the whole semester, each of the systems in Project B can only be used for several hours only. Also, to minimize learning overload for end-users, we'll integrate all our systems, including systems in Project B, under the WWW environment. For example, students will click a button on the WWW browser to invoke an interpreter for doing their homework or Reciprocal-Tutoring-Kids to practice recursions.

Although the idea of Project A/B looks trivial; it helps to clarify conceptually how to proceed with the development of a global social learning club, at least for the first few versions. Some members of our project are from different universities (they could come from different countries in the future). It is important that members of the project can use project A to teach their course. Thus, Project A is a "common core" for the research team. Project A spans over a whole course—it is like a "surface"—whereas Project B consists of a number of individual research sub-projects (or "points") that may focus on specific areas of the course.

The experimental results of Hiltz and her colleagues (Hiltz, 1995) ensure that networks can be used to teach a whole course. Using Project A as a bedrock, we can observe the specific requirements of the environment and develop AI systems in Project B to enhance the performance. This tactic perhaps can be called “research on demand.”

When used in parallel with Project A, the systems in Project B enhance student’s learning in the course. We feel, at least psychologically, that these systems are “usable” and we are more compliant to maintain and improve the systems. In Project A/B, AI is not pitted against other technologies, but integrated with them. It provides a framework to demonstrate that, no matter how advanced and prevalent the network technology, the environments with the furtherance by AI technology will always prevail.

Although the first version of Project A applies the current available technology, it allows us to establish a learning community quickly. With such a learning community, we can develop a stronger intuition and a better feeling of what is really going on in the world. It is also natural in such an environment to adopt the participatory or socio-technical approach to design (Clancey, 1992). We may simply refer this as “situated research.” “Users (students and teachers) must participate from the very beginning. But also, because the world is a messy place (we cannot specify[1] once and for all how the world or people work), we must develop our designs in the course of use, incrementally, with relative quick periods of use, observation, reflection, and redesign. That is, our computer systems, as artifacts that fit into people’s lives, must develop in a context that includes the user’s everyday adaptation (Clancey, 1992, p. 28).”

In participatory design, we can observe how students and teachers use the systems, what difficulties arise, and what will be needed. The network server allows us to collect data directly through the networks and have direct discussion with the students and use other forms of inquiry such as online questionnaires. These observations and data collection processes are necessary for the evolutionary development of the systems and last for the whole course.

It is worth noting that although there are no constraints on when and where students use the system, the server can easily collect a large amount of data from users’ interactions with the system. For example, the average response time of the student tutor to the tutee’s question or the average time interval between hitting two buttons or the rate of hitting the correct buttons when the tutee is constructing a solution. Thus, data mining or deriving information from this large amount of collected data is a precious resource to assist the users. Also, email graders can vary from very simple to very intelligent. For example, if the homework is a programming assignment, the grader may just run several test cases unknown to the student and grade the homework based on whether the program can give the correct answers to these cases. More intelligent email graders can function as remote intelligent tutors to diagnose students’ errors and give appropriate hints or advice to the students or as VLCs to discuss and share ideas with the student.

The important implication of Project A/B idea is that it is a “mini-infrastructure” for growth. General tools such as the WWW provide a broad ground on which researchers can develop niche systems and tools for specific purposes. It is an ecosystem which through continued use can evolve, grow, and integrate new technologies. Sub-systems will continually flourish, split, merge, or disappear. New members of the research team will join and old members will drop out, in a process similar to the natural selection of nature which will ensure the survival of the system and a high standard of excellence.

Learning cannot be isolated from other activities in our daily life. A salient feature of network communication is that it can free students from the constraints of time and physical space, resulting in a better chance that these systems can pierce into their daily activities. Thus, as a step to our research, it is important to establish such a virtual learning community that can increase the likelihood that the students life will be integrated. Indeed, one of the greatest achievements of the computer science community has been the use of networks in their daily activities with consequent formation of user communities and the ongoing evolution of the communities into broader and larger structures, including other social institutions. In other words, a tool for daily life usage can be effectively tested and modified if the community that creates the tool uses it in their daily life.

In the next version of our global social learning club, we plan to add facilities in Project A to improve students' communication. Also, in the Reciprocal-Tutoring-Kids system, improvement of the performance learning companions in the system will continue. We shall add a visualized tracer (Tung. 1995) so that missing steps in running a program will be visible and able to be manipulated by the students. When students can trace the execution of their programs step by step using the tracer, their difficulty in debugging programs can be lessened.

Other possible future work will develop artificial agents that help students reduce their information overload by sorting, filtering, and archiving electronic messages from other students. In cases of difficulty, virtual agents will search for other students on-line who might be appropriate partners for discussing the relevant problem. This process might involve consulting the assistant agents of other students, or searching the communal database for dialogues or secondary' learning material that might relate to the student's question (Chan, 1995c). We may take advantage of our specially designed agent-oriented programming language, CAROL5, to develop programs for this purpose. Now two areas of research merge. Also, the process of integrating new technologies can continue to draw insights and ideas for further development. For example, incorporating video on demand and a video conferencing systems such as CU-SeeMe in Project A may lead to revised ways of designing the learning material and communication strategies, respectively.

Five Educational Goals

There is a Chinese adage stating that the *Five Educational Goals* are *Ethics*, *Wisdom, Athletics*, *Sociability*, and *Art.* Ethics comes before wisdom and knowledge is only one of the factors contributing to wisdom. In view of our expectations concerning the nature of the future technology supported learning environments, the following revised set of the Five Educational Goals may be more appropriate: Cognition, Motivation, Sociability, Attitude, and Ethics (Figure 4). Although the immediate goal of the system is cognition, we cannot isolate cognition from other goals that contribute profound effects to each individual. These effects are immense and subtle and lead to individually different cognitive behavior and performance.

Cognition. The cognitive goal is always the foremost and immediate concern. Whether it is accomplished using one-on-one tutoring or with other environment that regards the computer as a tool for learning, cognitive development is always based on the idea of "learning by doing." The cognitive goal of social learning design is: To add social elements in the spontaneous process of learning by doing, thereby encouraging students' *articulation* and *reflection.* The learning process, therefore, can be characterized as cycling phases of conceptualization, construction, articulation, and reflection. First, learners should form their own view of the material presented. After which the material should be used and applied in new situations so that it is better assimilated. Then, view's of the subject matter should be shared with others, mutually questioned, articulated through explanations, reflected upon, and responded to (Hietala, 1995).

Motivation. For psychologists, the term motivation is concerned with questions related to behavior regulation: What energizes action; how action is directed; and what action is under voluntary control (Deci, 1992). Motivation certainly relates to learning. It concerns whether a student is willing to learn, not whether the student is able to learn. If a student is making an effort to learn without conscious awareness of the effort, then the motivational goal has been achieved (Chan & Lai, 1995). A further step in the process perhaps is the *continuing motivation* as when a student returns from a previous use of the system without external constraint to do so. The study of motivation in learning, on one hand, can go deeper to gain fuller understanding of the very nature of human beings and, on the other hand, can help accumulate information that is relevant to the building of learning environments that nurture motivation.

Sociability. Knowledge can be regarded as a common consensus of a community, and therefore is rooted in social activity. Learning activity in a sociocultural context reassembles the process for arriving at such a consensus and serves as a catalyst of knowledge cultivation. Social learning simply aims at fostering the recurrence of some social learning mechanisms such as alternatives seeking, conflict generation, mutual reflection, consensus formation, and appropriation[2], but in a more condensed and effective form. There are of course effects of social contexts on motivation since people have an innate need to relate to and affiliate with others as well as to seek social approval. The learning of a subject involves a prolonged period of human interaction. Students not only communicate, collaborate, or compete to learn, they also learn how to communicate, collaborate, or compete. Through these activities, they learn how to understand others, show care and respect, keep commitments, and trust; all of these behaviors are advantageous to learning how to build human relationships.

Attitude. Attitude is a special type of belief that describes some evaluation of an object or situation and serves as a heuristic to influence interpretations, explanations, reasoning, and judgment. Consequently, it may usually assign this object or situation to favorable and non-favorable categories. In a social learning environment, we should make efforts to help students develop attitudes that are open, enthusiastic, willing to seek out help and input from others, able to share success and failure with others and appreciate that making errors is a part of the natural learning process—we all learn by making errors as long as we can recognize them. Furthermore, values of the effort expended rather than the product achieved or ability should be emphasized (Anderson, 1992).

Ethics. Ethics is perhaps the ultimate educational goal. In his international bestseller, Covey (1992) pointed out that, in the past, a character ethic was the foundation of success. Things like integrity, humility, fidelity, temperance, courage, justice, patience, industry, simplicity, modesty were essential factors of success. After World War I, the basic view of success shifted to personality ethic. Success was a function of personality, of public image, of behaviors, skills and techniques that lubricate the processes of human interaction. Unfortunately, some parts of the new personality approach were manipulative and even deceptive. Ethics evolve as society progresses. In the information era, the ethics we adopt will certainly change.

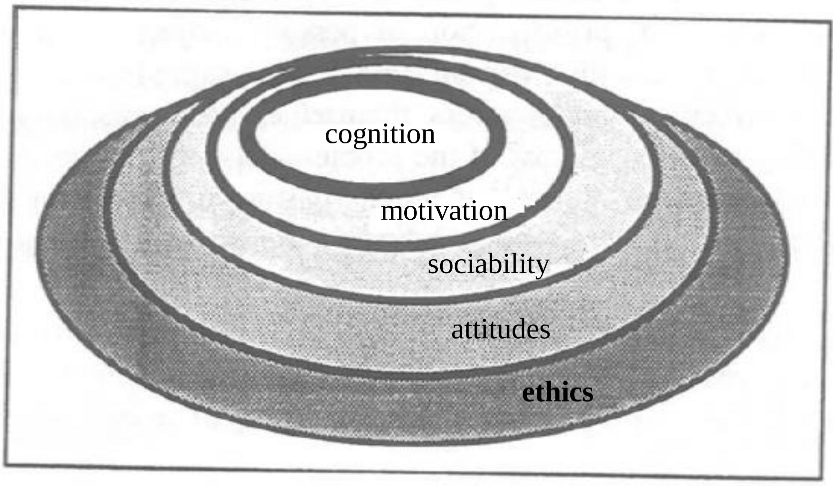

Figure 4. Five Educational Goals

In the past, our design of social learning systems has been dominated by the consideration of the first three goals, though none of them can be claimed to have been satisfactorily achieved. For learning, we have designed a number of social learning activities and developed artificial agents in the hope that they may help and promote students to learn. For motivation, it can either be achieved by the design of the system or the learning activities—collaboration, communications, and appropriately designed competition can all be motivating—or by detecting students' motivational states and adopting strategies to maintain it in the learning process (Soldato, 1992). For sociability, most learning activities we designed are intended for this goal, besides learning. For attitude, the system may advocate or discourage some particular attitudes at opportune moments during learning. It is desirable that the system is able to detect students' established attitudes and then create situations either to reinforce some attitudes or challenge and doubt other attitudes. For ethics, we may explicitly teach them (Roberts, 1995). Certainly this does not seem to be an interesting way nor the best way. Students who use networked environments to learn are already residing in a society in which they can learn ethics through their practice, possibly with the advice of the system. This process may be slow but only through this way, will students become aware of this larger objective of learning. In general, how to achieve the five educational goals will remain a long term, if not the ultimate, problem.

If we can shift more of our attention to the adaptation of systems to daily life, perhaps we will be able to gain insight into how to explore these goals all together. To this end, developing intelligent agents which serve as the personal assistants of students may be a good entry point, especially in situations where students are taking a number of intelligent virtual classes. These assistants can help students better organize and manage their learning as a part of their daily life. They may also play the role of a counselor to advise students on things indirectly related to their academic work, such as time management, prioritization of personal objectives, participation and sharing of vision with other students, and learning how to better communicate with others and express themselves. As assistant agents for teachers, they can be assessors of the problem solving abilities of students, advisors to teachers, matchmakers for the partnership and teaming of students, counselors to parents, career advisors, and so on (Tanimoto, 1995).

DISCUSSIONS AND CONCLUSIONS

The infrastructure and the educational goals discussed above are essentially dimensions of issues that we have to consider in our design, that is, how to put the artifacts of our design in the space (or hyperspace) generated by these dimensions. The approach of Project A/B for the development of global social learning club is receptive. It encourages integration and compromise of the use and the development of the system by different team members. Besides, it shows that, like Clancey's advice on the socio-technical approach, the strategy of doing AIED research is important, possibly a subject of research itself.

Teachers will be a crucial element in the future of the global social learning club even though there is a clear picture of how their roles will change. We should utilize teachers' and students' human intelligence in our design which in many aspects are still superior to AI. Also, AI techniques should assist teachers to run and manage classes as well as students' personal management. Our goal should go beyond the cognition goal and more attention should be paid to other educational goals that may, in turn, have enormous but implicit effects back to the cognition goal.

In addition to the learner-centered view of the learning environment discussed in the section on Infrastructure of Learning Environments, it is worthwhile to consider the teacher-centered view. Indeed these two views are complementary and sometimes mutually beneficial. For example, an email grader is a convenient tool (saving the work of grading) from the view of a teacher; but it turns out that it is also a marvelous tool for the students. Students not only get immediate feedback by electronic submission once the homework is done, but also may compare the scores obtained with other students' shown in the grade book, urging them to make further efforts to improve their homework for a better score.

Cooperative projects involving different nations and cultures can help us think big and identify future research issues. In particular, when advanced networks become more accessible, the need for ingenious design of effective learning activities will be acute. However, we do not have to do big. We still need more theoretical work before we will be ready to implement models of the complex and dynamic future learning environments. More powerful tools such as visualization tools are needed. Individualized learning will remain an important part of the learning repertoire in all kinds of learning environments. The student model will be used in various ways, rather than just a covert component inside the system. It is also desirable for a system to possess natural language processing ability to engage students to have dialogue, negotiation, argumentation (Baker, 1994) on topics that may go beyond the learning task and touch other educational goals. Indeed, I believe that the need for natural language processing ability in a limited subject domain in networked social learning environments will become more and more salient. AI is not everything, but empowers us in the course of designing.

Traditional computer scientists are used to focus on the development of tools such as operating systems and database management systems. AI researchers, on the other hand, look at humans and center on their cognition, sometimes to the molecular level. As interconnectivity increases at all levels, we will have a unique opportunity to create and help manage a virtual society. Our imagination should take a wider perspective, not only in terms of human machine interactions, but in terms of humanity.

Computers and networks are currently marching into our daily life and families. As further widespread use of this information technology continues, a radical change in learning environments will emerge. Facing the rapid change of the unknown future world, learning and lifelong learning seem to be the only keys for further growth and survival. As people from all walks of life begin to recognize that what we know about and rely on today will be obsolete tomorrow; a global wave of learning will emerge. The notions of “learning individual,” “learning organization,” “learning society,” and “learning country” will then become popular. As a result, there will be a mounting interest in this field and the research population comprising researchers from various backgrounds will be blooming. If we can anticipate such a trend, then we can prepare for it

References

Anderson, J.R. (1992). Intelligent tutoring and high school mathematics. In C. Frasson, G. Gauthier, & G. McCalla (Eds.), *Lecture Notes in Computer Science, 608,* (pp.1-10).

Berlin, Heidelberg: Springer-Verlag. Baker, M.J. (1994). A model for negotiation in teaching-learning dialogues. *Journal of Artificial Intelligence in Education, 5,* 1 99-254.

Barnard, Y., & Sandberg, J. (1993). Open learning environment: What support should they offer? *Proceedings of International Conference on Computers in Education,* Taipei, Taiwan, 156-161.

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.

Chan, T.W. (1991). Integration-Kid: A learning companion system. *The Proceedings of the 12th International Joint Conference on Artificial Intelligence,* Sydney, Australia, Morgan Kaufinann Publishers, 1094-1099.

Chan, T.W. (1992). Curriculum Tree: A knowledge-based architecture for intelligent tutoring systems. *The 2nd International Conference of Intelligent Tutoring Systems, Lecture Notes in Computer Science,* Springer-Verlag, *608,* 140-147.

Chan, T.W. (1995a). The research and development of intelligent computer assisted learning systems, *National Science Council Monthly*, *23*, 456-468.

Chan, T.W. (1995b). A tutorial on social learning systems. In T.W. Chan & J. Self (Eds.), *Emerging Technologies in Education* (pp. 71-96). Charlottesville, VA: AACE.

Chan, T.W. (1995c). Artificial agents in distance learning. *International Journal of Educational Telecommunications, 1,* 263-282. Charlottesville, VA: AACE.

Chan, T.W. (1995d). Global social learning club, unpublished manuscript. National Central University, Taiwan.

Chan, T.W., & Baskin, A.B. (1988, June). Studying with the prince: The computer as a learning companion. *The Proceedings of International Conference of Intelligent Tutoring Systems,* 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.* New Jersey: Ablex Publishing Corporation, 6-33.

Chan, T.W., & Chou, C.Y. (1995). Simulating a learning companion in reciprocal tutoring system. *The Proceedings of Computer Support for Collaborative Learning Conference,* 49-56.

Chan, T.W., Chou, C.Y., Lee, M.F., & Chang, M.H., (1995). Reciprocal- Tutoring-Kids: Tutor-Tutee Role Playing Systems. *World Conference on Artificial Intelligence in Education*, Washington, DC, USA.

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

Chan, T.W., & Lai, J. A., (1995). Contest-Kids: A competitive distributed social learning environment. *Proceedings of World Conference on Computers in Education,* Birmingham, England, 767-776.

Chan, T.W., Lin, C.C., Lin, S.J., & Kuo, H.C. (1993). OCTR: A model of learning stages. *Proceedings of the World Conference on Artificial Intelligence in Education,* 257-264.

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

Clancey, W. (1992). Guidon-manage revisited: A socio-technical systems approach. In C. Frasson, G. Gauthier, & G. McCalla (Eds.), *Lecture Notes in Computer Science, 608,* 21-36. Berlin, Heidelberg: Springer- Verlag.

Covey, S.R. (1992). *The seven habits of highly effective people.* Simon & Schuster.

Deci, E.L. (1992). The history of motivation in psychology and its relevance for management. In V.H. Vroom and E.L. Deci, (Eds.), *Management and Motivation,* Second Edition, Penguin Books.

Dillenbourg, P. (1995). Collaborative learning and the Internet. *International Journal of Educational Telecommunications, Special Issue on Emerging Perspectives of Distance Learning, 1* (2/3), 131-146. Charlottesville, VA: AACE.

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.

Hietala, P. (1995). Personal Communication.

Hiltz, S.R. (1995). Teaching in a Virtual Classroom™. *International Journal of Educational Telecommunications, special issue on Emerging Perspectives of Distance Learning, 1(2/3),* 185-198).

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.
Lin, G.L. (1993). *Three's company: A social learning system.* Master's 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.
Newman, D. (1989). Is a student model necessary? Apprenticeship as a model for ITS. In D. Bierman, J. Breuker, & J. Sandberg (Eds.), *Artificial Intelligence and Education.* Amsterdam, IOS, 177-184.
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.
Roberts, P. (1995). Ethics: The neglected factor in computing education, *Proceedings of World Conference on Computers in Education*, Birmingham, England, 921-928.
Self, J. (1986). The application of machine learning to student modelling. *Instructional Science*, *14*, 327-338.
Self, J. (1988, June). Bypassing the intractable problem of student modelling. *International Conference of Intelligent Tutoring Systems*, Montreal, Canada, 18-24.
Soldato, T.D. (1992). Detecting and reaction to the learner's motivational state. *The 2nd International Conference of Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Springer-Verlag.
Tanimoto, S. (1995). Personal Communication.
Tung, S. (1995). Teaching programming in viewscheme. *Proceedings of Fourth International Conference on Computers Assisted Instruction*, Hsinchu, Taiwan, R.O.C., S16-1-S16-10.
VanLehn, K., Ohlsson, S., & Nason, R. (1994). Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education*, 5(2), 135-175.
Vygotsky, L. (1978). *Mind in society*. M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Trans. Cambridge, MA: Harvard University Press.

Acknowledgements

I gratefully acknowledge the National Science Council of Taiwan for its support of the LISA project. I am particularly grateful to Yen-Wen Kuo for her continuous encouragement and urging that caused me to think of my research in a longer term and with a broader perspective. I would like to thank the following people for their discussion of this paper: Pentti Hietala, Simon Tung, Wing-Kwong Wong as well as my students and other members of the LISA project. I also thank for the editorial help, Mike Evans and John Self. Thanks are also due to my wife, Lisa, and daughters, Carol and Joyce, for their moral support while I was preparing this paper.

Questions:

You mentioned that evaluation was a weakness in earlier projects but you didn't say much about it. Is it something that you do, and if not, why isn't it emphasized more in your projects?

This is a very difficult thing for us. For one thing, I don't think we are well- trained for that. Also, we like to build systems, vary them, look at them in a month, and decide if they're any good. So the systems are very unstable. We find these kinds of formative evaluations very useful. But we don't have enough man power to do summative evaluations. Summative evaluation is indeed another independent project. In the future, we are planning to put our systems on the Internet and in two or five years time when our systems are stable we might be able to do some more serious formal evaluation.

There's a bit of a theme at this conference that collaboration is good. Is this just the flavour of the decade? Do you have any signs that learners differ, that some learners don't learn well in collaborative learning? Do you have any ideas about how this trend should be limited?

We didn't observe anything that indicated collaboration was a mistake, but we did notice that some types of learners don't collaborate. That's why we need some kind of nice design to set up some rules of the game to help them collaborate. Also, competition game may not be that good because some students don't like competition. I cannot see why we need to limit the trend. Apart from learning how to learn with collaboration, we need to help students learn how to collaborate, which is a training needed for our youngs in the next century. Of course, you can seek other learning paradigms besides collaboration.

Can you use the student models of the distributed learners to help perform evaluations?

Well, it depends what you mean by a "student model." Once we have installed the system on the Internet and have a workstation as a server it will be no trouble collecting the data. Right now, students can choose either a real learning companion (a real student) to interact with or, if they prefer, an artificial learning companion. So we can evaluate whether the artificial learning companion is an effective agent. At the moment, most students prefer to have a real learning companion but sometimes students prefer to practice first with the artificial companion before they interact with the real one.

Will AI have much of a role in modeling aspects of social cognition?

Sure, I have the confidence that AI can model something like motivation and will help to manage the learning of social skills and these kinds of things. But that's still a matter for research and we haven't done much work on that yet.

On the one hand, you say that some students don't like competition, so be careful with competition, and on the other hand, you say some students don 7 *seem to collaborate, so we have to force them to collaborate, almost, by changing the environment, and so forth. Why these two different treatments of these processes?*

Well, it's true that some students don't like competition. If it's not well designed, it can even be harmful and that's why we have to be careful. For collaboration, I didn't really mean we have to “force” the students to collaborate. What I meant was that the design of the collaboration activity must be “obligatory.” For example, you entered this room because you are attending an invited talk. To be a member of the audience, you found a seat there, then sat down to listen. So, when you entered the room, you’ve made an implicit agreement to follow the rules of the game by attending a talk. You’re a member of the audience and I’m a speaker. Look, if there were no seats here then the circumstances were informal and it’d be a completely different situation—all audiences would be standing or walking around. So the invited talk scenario is analogous to collaborative learning in the sense that there are coercive elements in the design of these social activities and the environment. Therefore, when students use the system, we, the designers, and they, the users, all assume implicitly that the users will follow the rules implied by the learning activities of the environment. This obligation or enforcement is therefore induced by the social assumption that everyone will follow the rules of the collaborative activities, even if they don’t know what the rules will be. The point is to design these rules that govern and coordinate the learning activities as well as the communications of the participants in such a way that a positive environment for learning is created.

Note

Based on an edited transcript of an invited talk at the World Conference on Artificial Intelligence in Education, Washington, DC, August 1995.

1. The Turing Test is a proposed experiment by Turing that asks a human person to detect whether the agent on the computer is a simulation program or a real human who controls the computer remotely.
2. See Dillenbourg (1995) for more details of such mechanisms.