DOCUMENT RESUME

ED 388 259 IR 017 408

AUTHOR	Jehng, Jihn-Chang J.; And Others
TITLE	TurtleGraph: A Computer Supported Cooperative Learning Environment.
PUB DATE	94
NOTE	7p.; In: Educational Multimedia and Hypermedia, 1994. Proceedings of ED-MEDIA 94--World Conference on Educational Multimedia and Hypermedia (Vancouver, British Columbia, Canada, June 25-30, 1994); see IR 017 359.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Cooperation; Foreign Countries; Instructional Design; Models; *Problem Solving; *Programming; Thinking Skills
IDENTIFIERS	*Collaborative Learning; *Learning Environment; LISP Programming Language

ABSTRACT

This paper discusses a computerized learning environment called TurtleGraph that is designed and developed to support collaborative problem solving. Within the learning environment, learners are requested to write computer programs to generate geometric figures. The instructional focus of the system is to enhance the learner's List Processor (LISP) recursive programming skill by making strategic thinking more explicit, inducing reflection through reciprocal evaluation and criticism, and fostering an active role of learning through collaboration. A conversational model is formulated to address the role of knowledge in the collaborative problem solving process, and the design of the system is mainly guided by the theoretical model. In addition to the model, several instructional design principles are also incorporated to make the learning environment more educationally effective and efficient. (Contains 17 references.) (Author)

TurtleGraph: A Computer Supported Cooperative Learning Environment*

JIHN-CHANG J. JEHNG
Institute of Educational Media & Library Sciences
Tamkang University, Taipei, Taiwan, R.O.C.
E-Mail: tkut067@twnmoe10.bitnet

YU-FEN SHIH
Department of Information Management
Ming-Chuang College, Taipei, Taiwan, R.O.C.

STEVEN LIANG & TAK-WAI CHAN
Institute of Computer Science and Electronic Engineering
National Central University, Chung-Li, Taiwan, R.O.C.

Abstract: The paper discusses a computerized learning environment called TurtleGraph that is designed and developed to support collaborative problem solving. Within the learning environment, learners are requested to write computer programs to generate geometric figures. The instructional focus of the system is to enhance the learner's LISP recursive programming skill by making strategic thinking more explicit, inducing reflection through reciprocal evaluation and criticism, and fostering an active role of learning through collaboration. A conversational model is formulated to address the role of knowledge in the collaborative problem solving process, and the design of the system is mainly guided by the theoretical model. In addition to the model, several instructional design principles are also incorporated to make the learning environment be more educationally effective and efficient.

Theoretical Framework

Influence of peer collaboration on individual cognitive development has been investigated by a number of educational and psychological researchers, and findings of previous research have provided us rich information in this regard. First, collaborative problem solving could be an effective catalyst for the enhancement of individual cognitive skills. Piaget (1965) indicates that peer learning could reduce egocentrism and provide scaffolding for cognitive development. Alternative to this view, Vygotsky (1978) proposes the concept of the zone of proximal development that social interaction plays an important role in shaping individual cognitive structures. The development of individual cognitive skills could be determined through problem solving in collaboration with more capable peers. Second, collaborative problem solving may appeal more intensity of learner's involvement and attention to the learning activity. Collaborative problem solving activities usually are intrinsically motivating to the extent that they engage individuals in a process of seeking to solve problems and accomplish goals that require the exercise of valued personal skills, such as the ability of self-determination and interpersonal skills (Lepper, 1985). Third, collaborative problem solving is important for inducing individual metacognitive skills. Participants in a group could learn to control and monitor their behaviors, and to develop more effective problem solving strategies through reciprocal evaluation and criticism (Palincsar & Brown, 1984).

Over the last several years, researchers have shifted the focus and become intrigued by the reciprocal relations between individual cognition and social interaction (Resnick, Levine, & Teasley, 1993). Rather

* The research project is under the group project entitled LISA, which is currently funded by the National Science Council, Taiwan. LISA (Learning IS Active) is a group project with social learning as the central focus. The goal of the project is to establish a multi-channel learning envrionment, where a student engages in social learning and interacts with various agents who may be human beings or computer simulated agents. A dozen of researchers and graduate students from different universities in Taiwan are involved in the LISA project.

than seeking to understand cognitive and social processes in isolation or treating one process as context for the other, a growing number of studies seek to provide conceptual schemes that allow the investigation of thinking as socio-cognitive activity. Particularly the development of the situated and social learning theory has allowed us to rethink the nature of human learning and problem solving. Learning is considered as the development of socially shared cognition (Brown, Collins, & Duguid, 1989). Knowledge is no longer merely regarded as stored representation of acquired experience. Knowledge is always a novel construction that continues to develop in every action in relation to the practice of a community (Clancey & Roschelle, 1991; Clancey, 1992).

The dynamic view of learning as continuous alteration of an individual's current knowledge state has certain instructional implication. In a typical teacher-student instructional situation, the goal of a teacher's work can be regarded as providing support to alter the student's evolving knowledge state so that it may converge to his own. An important function that the teacher's exercise is to monitor the learning activity and assist the student so that the convergence goes effectively (Chan, 1991). An analogous process can also be used to depict the collaborative problem solving. Participants in a group learn to construct knowledge by mutually examining, monitoring, assessing and justifying each other's knowledge. The more capable learners play a role of cognitive supporter to the less capable ones. A successful collaboration occurs when different versions of knowledge merge into one that is conducive to the solution of the problem.

In our preliminary study, subjects were arranged in pair and were requested to write LISP codes to solve some geometric figure problems. In order to explore the structure of a successful collaborative problem solving, the subjects' conversational dialogues were recorded and analyzed. By examining the dialogues, we found that the role of knowledge is critical to a successful collaboration. A conversational model was developed to capture its importance. In this model, we roughly partition a collaborative problem solving process into three knowledge construction stages, and they are knowledge communication, knowledge negotiation, and knowledge consolidation stage. It is argued that in the knowledge communication stage, participants in a group try to acquire as much problem information as possible. They merely exchange and transmit information to make the problem as clear as possible. Communication of knowledge requires least coordination of social interaction and involves little maintenance of shared conception. As participants acquire more problem information and understand each other better, they will be more involved and start to make contribution to the collaboration. At the stage of knowledge negotiation, an individual begins to formulate problem solutions and attempts to share his ideas with his partner as partial solutions are developed. He may make efforts to explain his ideas or seek to draw evidence to convince his partners while disagreement occurs. Proposals to the problem solution continue to be refined and be more consonant as more dissonance is resolved. Knowledge negotiation requires certain degree of coordination of participants and involves more maintenance of shared understanding. In the knowledge consolidation stage, participants of a group continue to collaborate to construct and maintain a shared conception of a problem. They work together to examine and validate their solutions. Knowledge consolidation is highly situated and requires the highest degree of coordination among the partners. The design of our collaborative learning environment is on the basis of this theoretical model. Figure 1 shows the three knowledge construction stages.

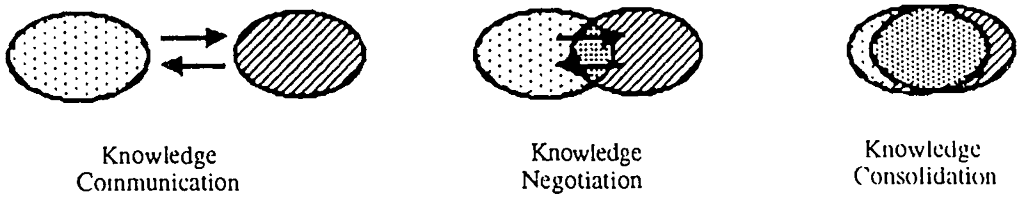


Figure 1. The conversational model

Since knowledge is critical to a successful collaboration, it is believed that cooperative learning environment must be designed in a manner that enables group members to communicate and share their ideas effectively. Baker (1991, 1992) indicates that an intelligent teaching system could provide some guidance in conflicting situations to facilitate student's explanations within a collaborative context. In order to provide guidance to facilitate knowledge construction, the development of our learning system uses a deliberately-designed restricted communication interface to induce individual meta-communication skills. Within the learning environment, the student is encouraged to express their thoughts explicitly by

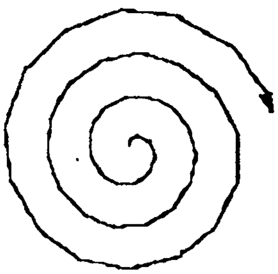
* identifying the goal of the problem,
* clarifying conflicting arguments,
* elaborating the supporting reason of explanatory propositions, and
* elaborating the opposing reason of argumentative propositions.

It is argued that a learning system that promotes meta-communication skills could help students maintain shared cognition, carefully form their argumentation, reduce ambiguous propositions, and construct thoughtful explanation. A successful collaborative problem solving activity can then be easily accomplished.

Principles Of Instructional Design

The LISP recursive function, that is taught at the introductory level of computer science courses, is the subject selected to explore the design and development of the cooperative learning environment. For students with computer science major, understanding the LISP recursive function is fundamental to the advancement of higher level of computer science courses. However, the student always have a hard time to learn to write the LISP codes, since little knowledge can be transferred from their previous experience about the procedure-based language, such as PASCAL and C. The goal of our system is to train students who have learned the syntax of LISP language with little LISP programming experience.

In the TurtleGraph, the student will learn the specific nature of the recursive function by programming LISP codes in combination with several simple LOGO commands. LOGO is a language developed by Papert and his colleagues in MIT, and it is used to teach students about the notion of computer programming and some deepest ideas of science and mathematics (Papert, 1980). In the TurtleGraph, the task is designed in a manner that the student is requested write codes that move the turtle at the pre-designated direction and angle to draw a geometric figure. While drawing the geometric figure and tracing the movement of the turtle on the screen, the student can readily capture the profound nature of the recursive function. By situating the learning activity in the problem solving context, the system can help the student understand the specific feature and function of acquired knowledge. Figure 2 shows a spiral form of geometric figure generated by the TurtleGraph codes.



```
(defun spiral ( counter length angle )
  ( cond
    ( ( /= counter 0 )
      ( fd length )
      ( rt angle )
      ( spiral ( - counter 1 ) ( + length 1) angle )
    )
  )
)
```

Figure 2. A spiral form of geometric figure with TurtleGraph codes.

Several instructional principles are also incorporated in the system to make each learning activity be more educationally sound.

A Controlled Conversational Environment. The design of the TurtleGraph applies the "button control" strategy (Schank & Jona, 1991) to help the student regulate the conversational protocol so that it can promote more effective and efficient collaborative problem solving. Each time when the student presses a conversation button to make communication with his partner, the button prompts him to elaborate his conversational statements depending on the type of conversation button selected. It may request the student to elaborate what he is trying to propose, or the reason why he agrees or disagrees with his partner's ideas. The button is designed to assist problem solvers to carefully develop their ideas and regulate their conversational behaviors during the collaboration. All the conversational dialogues are recorded and displayed in sequence on the computer screen. The student can utilize the dialogue protocol to (a) understand the interaction, (b) reflect his own problem solving behaviors, and (b) organize the conversational action and sequence. With the assistance of the conversation button, students could hold one another accountable for meaningful participation in a collaboratively sustained problem solving situation.

A Conceptually Visualized Learning Environment. In the TurtleGraph, geometric figure problems are used as means of teaching the concept of LISP recursive function. It is a display-based problem solving environment that emphasizes representing problems in an external display (Larkin, 1988). Within the TurtleGraph, the student can immediately see and reflect on the process of how geometric figure is generated by the turtle while manipulating the LISP codes. The external display can help the student explicitly envision his current knowledge state and help him closely examine, analyze, and modify his knowledge. External display can also reduce the student's cognitive load by making the knowledge acquisition and interpretation processes perceptual, rather than logical. Processing information perceptually is easier for the novice learner than processing information logically. Moreover, arguments among participants in a group can easily be resolved through perceptual comparisons and judgments than through logical inferences from purely nonvisual facts.

A Structured Problem Solving Environment. In the TurtleGraph, various geometric figure tasks were designed in accordance with three different types of recursive function. The three types are head, middle, and tail recursion. The classification is based on the location of recursion in the program. A good number of representative problems were sampled with different complexity and difficulty. Each problem begins with a story and the student is assumed to play the role of computer graphic designer inside the story. Story-based problems help the student to envision how the knowledge could be used in realistic situations (Cognition and Technology Group at Vanderbilt, 1990, 1992). Just as an effective teacher pays close attention to a student's current level of ability and gives his tasks that build on his prior learning, the TurtleGraph is a structured practice environment that sequences learning experience based on each individual student's problem solving performance. Each time a student accomplishes a task, the TurtleGraph will assess the student performance and make a decision to give an easier or a more difficult problem to the student for next learning. The student must master a certain level of problem solving skill before advancing to attack another level of task.

The Architecture Of The System

The system is coded by the SuperCard application and the Lisp interpreter, and is run in the Macintoch work station. At the first section of the system, a tutorial is provided to teach the student several simple LOGO commands that are used to draw figures by moving the turtle on the screen. As soon as the student finish the tutorial, he will read a story containing the problem. By tracing and analyzing the student's prior performance history, the system will select a problem that corresponds to the student's current ability level. If the student chooses to attack the problem, the system will bring him to the TurtleGraph Cooperative Problem Solving Environment, where he can choose to solve the problem alone or seek a partner to collaborate with. The environment contains five instructional areas, including the Control Panel, the Turtle Window, the Dialogue Recorder, the Program Editor, and the Listener. Figure 3 shows the TurtleGraph Cooperative Problem Solving Environment.

The Control Panel contains two groups of control button: system and communication button. All learning activities are operated by the system button. The Help button contains general help information that could guide the student to use the system. The student can press the help button at any time to request assistance from the system. It is designed in a manner that is sensitive to the student's need. Each time the student presses the Description button, the system will provide detailed problem information. The Example button contains example programs for student's reference. It is designed to help the student organize his programming knowledge and develop problem solving strategies. The student can run the example programs and also can copy and adapt them for his own use. Within the TurtleGraph, the student can choose to solve a problem independently or seek to collaborate with other students. Once the student presses the Commu button, all the communication buttons become activated, and the student can start to work with his partner. By pressing the Check button, the system will judge the student's solution. The Check button also provides the Answer option if the student wants to find out the correct solution. The student can press the Exit button to quit the system and leave it at any point, and the system will record his performance status and update his performance history in the database. The TurtleGraph will provide instruction and suggestion at the end of each learning session.

All communication buttons are designed to support the collaborative problem solving. The student can press any communication button to discuss with his partner during the problem solving. Once the student presses the button, the system shows a dialogue box prompting him to either make a clear statement about what he is trying to do or elaborate the reason why he agrees or disagrees with his partner. The prompt is designed to help the student monitor his problem solving behavior, regulate the conversational protocol and make an effective collaboration with his partner.

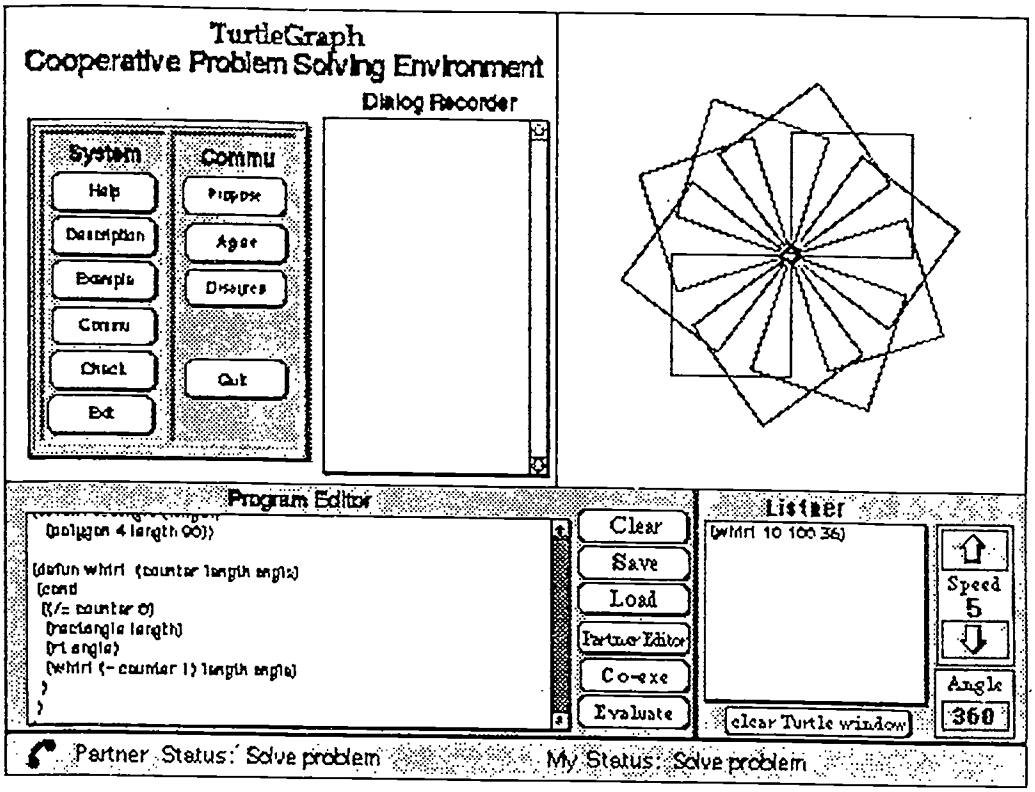


Figure 3. The TurtleGraph Cooperative Problem Solving Environment.

The Turtle window contains a turtle for drawing. Inside the window the screen will show the figure with the movement of the turtle immediately after a program is executed. The Dialogue Recorder keeps track of all the dialogue statements in sequence maoe by the problem solvers during the communication, and all statements can be saved for future reference. Each dialogue statement in the Dialogue Recorder is marked with the communication button that the student just pressed to highlight the content of the statement. The student can read and reflect on the dialogue statements during the collaboration.

The lower part of the screen is the Program Editor and the Listener. The student generates the LISP codes inside the editor window. The editing function like copying, cutting, and pasting can be accomplished by pressing the pre-defined shortcut keys. The purpose of the save and load button is for saving and retrieving programs. The student can take a look at his partner's program by pressing the Partner Editor button. Another special design in the Program Editor is the Co-exe button. The student can press Co-exe button to execute his program and the figure that the program generates will simultaneously appears in his and the partner's Turtle Window. The design of the Partner Editor and Co-exe button aims to facilitate the process of collaboration.

The student can evaluate his TurtleGraph codes at any time inside the Listener box. He can also press the Speed button to control the speed of Turtle movement. The Angle Box shows the degree of the angle when the turtle make a turn. At the bottom of the screen the system shows the message of the collaborative status. All facilities provided by the TurtleGraph Cooperative Problem Solving Environment fully support the collaborative problem solving activity.

Discussion

This paper presents an ongoing research project that has designed and developed the TurtleGraph collaborative learning environment. With the restricted communication interface, the system can prompt

students to make more effective collaboration by reducing the ambiguity of the explanation, elaborating the augmentative reasons, organizing the dialogue sequence during the conversation. In addition we plan to make greater efforts to include AI techniques to expand the capability of the communication facility and the answer diagnosis. The technical infrastructure of the TurtleGraph cooperative learning environment will continue to develop with more thorough evaluation.

Acknowledgments

The authors gratefully acknowledge the valuable suggestion made by Dr. Michael Jacobson in the final draft of this paper.

References

Baker, M. J. (1991). The influence of dialogue processes on students' collaborative generation of explanation for simple physical phenomena. In *Proceedings of the International Conference on the Learning Sciences* (pp.9-19).

Baker, M. J. (1992). Modeling negotiation in intelligent teaching dialogues. In M. Elsom-Cook, R. Moyse (Eds.), *Knowledge negotiation* (pp.191-240). London, UK: Paul Chapman Publishing.

Brown, J. S., Collins, A., Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-42.

Chan, T. W. (1992). Integration-Kid: A learning companion system. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (pp. 1094-1099). Sydney, Australia: Morgan Kaufman Publisher, Inc.

Clancey, W. J. (1992). Guidon-Manage revisited: A socio-technical systems approach. In *Proceedings of Intelligent Tutoring Systems '92*. Berlin, Germany: Springer-Verlag.

Cognition and Technology Group at Vanderbilt. (1990). Anchored instruction and its relationship to situated cognition. *Educational Researcher*, 19(6), 2-10.

Cognition and Technology Group at Vanderbilt. (1992). The Jasper experiment: An exploration of issues in learning and instructional design. *Educational Technology: Research and Development*, 40(1), 65-80.

Larkin, J. H. (1988). Display-based problem solving. In D. Klahr & Kotovsky (Eds.), *Complex information processing: The impact of Hebert A. Simon* (pp. 1-39). Hillsdale, NJ: Lawrence Erlbaum.

Lepper, M. (1985). Microcomputer in education: Motivation and social issues. *American Psychologist*, 40(1), 1-18.

Palincsar, A. S., & Brown, A. L. (1978). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition & Instruction*, 1(2), 117-175.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.

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

Resnick, L. B., Levine, J. M., & Teasley, S. D. (1991). *Perspectives on socially shared cognition. Washington*, DC: American Psychological Association.

Roschelle, J., & Clancey, W. J. (1991). *Learning as social and neural*. Paper presented at the 1991 Annual Meeting of the American Educational Research Association at Chicago.

Schank, P., Linn, M. C., & Clancy, M. (1990). *How does an on-line template library help students learn PASCAL?* Paper presented at the 1990 Annual Meeting of the American Educational Research Association at Boston.

Schank, R., & Jona, M. Y. (1991). Empowering the student: New perspectives on the design of teaching systems. *The Journal of Learning Science*, 1(1), 7-35.

Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press. (Original work published in 1935.)