Personal Computers and the Problem-Solving Skills of Young Children

Sandra Munro-Mavrias

A Thesis

in

The Department

of

Education

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Arts at
Concordia University
Montréal, Québec, Canada

March 1985

ABSTRACT

Personal Computers and the Problem-Solving Skills of Young Children

Sandra Munro-Mavrias

The purpose of this study was to observe the effect of selected personal computer experiences, especially Logo, on the general problem-solving skills of kindergarten children.

A group of forty-four kindergarten children were tested, using a non-verbal creativity test to assess skill in divergent problem-solving, and a standardized conservation test to assess skill in convergent problem-solving. The subjects were randomly assigned to one of three treatment groups. The groups were introduced to the personal computer shortly after the testing and had weekly small-group twenty-minute sessions with personal computers for the following twenty weeks. One group learned a simplified form of Logo for non-readers and used it to develop their own programs. A second group worked with software packages designed to teach thinking skills. The third group functioned as a control, working with drill-and-practice software. All subjects were assessed again by the same tests after the treatment was completed.

The programming behaviour of the Logo group was observed in terms of goal-setting and treatment of errors. The behaviour was recorded by checklist. It was

hypothesized that kindergarten children would be able to set their own programming goals and to use strategies to solve their programming problems.

It was also hypothesized that the exercise of certain problem-solving activities on the personal computer would enhance the development of general problem-solving skills of kindergarten children, and that actual programming experiences with Logo would be more effective than other computer experiences.

The experimental results demonstrated no significant differences in the problem-solving skills of the three groups. The Logo programming activities showed that there was significant development in programming proficiency and that strategies for solving problems were applied. On the basis of observations from this study some recommendations were made for the classroom use of personal computers with kindergarten children.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

LIST OF TABLES

CHAPTER I

INTRODUCTION

It has been suggested by Papert (1980) that the personal computer is an exceptionally good tool to teach generalizable problem-solving skills and to expose the hidden processes of thinking, particularly for young children. To put it somewhat simplistically, the personal computer can be used to teach the child in two very different ways: it can program the child (the computer as tutor), or, as Papert proposes, the child can program the personal computer (the computer as tutee) by using Logo, an interactive programming language especially designed for young learners.

The present classroom use of personal computers with young children has very little foundation in research, and arises from the introduction of computers into elementary education. It is mainly an exploratory activity arising from the interest of teachers coupled with some pressure from parents and school administration. Reactions to this activity are, on the whole, very positive (Conlin, 1981; Nelson, 1981; Swigger and Campbell, 1981). There is a real need for research into the appropriate uses of the computer in the early childhood classroom, particularly with regard to the appropriate uses of Logo and other types of software which are meant to develop the cognitive skills of young children (Brady and Hill, 1984).

The development and exercise of intelligence is a major goal of education. According to Resnick and Glaser (1976), a main aspect of intelligence is the ability to solve problems; and since the incompleteness of instruction, even in school, is a recognized fact of education, we must examine this ability to solve problems as a means of promoting the development and exercise of intelligence, of learning on one's own.

Klahr (1978) suggests that we need to understand and ultimately improve the problem-solving abilities of young children for two reasons: on the practical level because these abilities are often implicitly assumed in early school activities, and on the theoretical level because children learn about problem-solving even without direct instruction. Further, Flavell (1976) points out that children often fail to solve problems for which they possess the necessary solution procedures, and it is important to find out why this is so.

The teaching of strategic thinking is rarely an overt classroom activity, but is considered a covert outcome of all the other activities. De Bono (1972) strongly asserts that it needs to be directly taught and, since he sees problem-solving as a special case of thinking, recommends it as a convenient format with a defined objective. He maintains that young children can be brilliant thinkers, very fluent with ideas. They enjoy thinking, "the deliberate (mental) exploration of experience for a purpose" (De Bono, 1976, p.32), enjoying the use of the mind as they enjoy the use of the body.

The purpose of this study is to examine the types of behaviours and strategies demonstrated by kindergarten children in programming the personal computer, and to examine the effect of different types of personal computer experiences on the general problem-solving skills of kindergarten children.

The questions posed here are as follows:

1) How do kindergarten children cope with computers generally?

2) What were their reactions to the different types of software?

3) What was the nature of their social interactions as they worked with the computers?

4) What behaviours do kindergarten children demonstrate while using Logo and what strategies do they use to solve their programming problems?

5) Can computer-assisted learning enhance the general problem-solving skills of kindergarten children?

6) Can Logo enhance the general problem-solving skills of kindergarten children to any greater extent than other forms of computer-assisted learning?

CHAPTER II

A REVIEW OF THE LITERATURE

This chapter will attempt to provide the background from the research literature for an examination into the use of personal computers for the enhancement of young children's problem-solving skills. It will deal with a discussion of young children's approach to problem-solving and examine some of the recommendations that have been made to foster these skills. It will then address the use of the personal computer with young children in terms of its value and its limitations, in terms of the types of computer-assisted instruction and computer-assisted learning, including Logo, that have been examined, and lastly in terms of the social concerns.

Definition of Terms

Young Children. Young children, for the purposes of this review, will be considered to be those of preschool age, from three to six years.

Cognitive Development. These young children, according to Piagetian theory, are pre-operational in terms of cognitive development, with the following characteristics of thought: egocentrism, syncretism, juxtaposition, transductibility, irreversibility. Pre-operational children pass swiftly from a state of belief to a state of invention, and there exist two realities: play and observation. When they are in the presence of one reality, they forget the other. Until the transition to the stage of concrete operations, the desire for causal explanations proper is extremely weak (Piaget, 1928).

Learning takes place in a process of adaptation, through assimilation and accommodation, as the young child interacts with the environment (Ginsburg and Opper, 1969). Children actively construct knowledge; noticing features of objects or events is not enough: until something makes children uncomfortable about their present construction of the situation, they do not learn much from observation (Copple et al., 1979).

Neo-Piagetian theorists describe the development of thinking in the pre-operational period in more detail. Children go through at least four substages and at each of these stages they are capable of encoding and reproducing all features noticed in the immediately previous stage, plus one more. Children's progress through these stages is affected by their specific experiences. The growth of children's short-term memory plays an important role in regulating the rate of progress. The sequence grows from one unit at age one, to five units at four to five years (Case, 1978b).

<u>A Problem.</u> What then is a problem? According to Newell and Simon (1972), it is when "a person wants something and does not know immediately what series of

actions he can perform to get it." This is supported by Resnick and Glaser (1976) who describe it as "a situation in which an individual is called upon to perform a task not previously encountered and for which externally provided instructions do not specify completely the mode of solution."

According to Resnick and Glaser, that which has a clearly defined procedure for solution is not a problem but a task. In accepting this constraint to the definition we then eliminate algorithms as problem-solving procedures. Problem-solving then is not the application of a formula but a metacognitive process.

Problem Recognition. A given in any problem-solving situation is the recognition that there is a problem to be solved. A study by Resnick and Glaser (1976) with grade one children suggests that problem detection is indeed an important part of the process, in that the way problems are initially detected, and therefore encoded, will probably affect the quality of the observed solution behaviour. Bruner (1966) says that children will try to solve problems if they recognize them as such, but they are not often either predisposed to or skilful in problem-finding. He maintains that they can be quite quickly led to such problem-finding by encouragement and instruction. On the affective side, to encourage children in problem-finding and solving, we must establish in children's minds their right not only to have their own private ideas but to express them in the public setting of the classroom.

Problem-Solving. Newell and Simon (1972) describe problem-solving behaviour as adaptive behaviour which is iterative, consisting of repeated loops around a circuit: select a goal, select a method, evaluate the results, select a goal again. A strategy is the control structure governing this behaviour of the problem-solver in a given task environment.

They describe man as an information processing system (IPS) when he is solving problems. An information processing system comprises "input-process-memory-process-output." The human IPS "must grow itself into its adult normal state, starting from a primitive neonatal state that is not yet well understood - at each moment of growth the ongoing changes in the cognitive system must be capable of being assimilated by the system that is already there, at the same time that the latter remains in reasonable working order" (p. 866). Newell and Simon assert that the programmed computer and the human problem-solver are both species belonging to the genus IPS and that the flexibility of computers is the key to understanding human problem-solvers.

Polya's (1957) model presents a means to operationalize the process of general problem-solving. He describes four steps in the process:

1. understand the problem;
2. devise a plan;
3. carry out the plan;
4. look back on the solution to see if it works.

There appear to be two major subskills identifiable in Polya's model. Convergent thinking is necessary for step 1 in order to be able to focus on the nature of the problem; divergent thinking is required initially for step 2 to generate a variety ideas for the solution and then convergent thinking is necessary to judge the potential suitability of these ideas; step 3 probably employs both types of thinking in applying strategies; and step 4 calls for convergent thinking to evaluate the outcome.

The pattern of problem-solving skill from novice to expert begins with little or no intelligent self-regulation, which is where most young children are. It progresses to familiarity with the necessary rules and subprocesses, an increasingly active period of deliberate self-regulation, and on to smooth performance with the automatic running of subprocesses at the expert level (Brown and Deloache, 1978).

Flavell (1976) coined the term "metacognition" to refer to this process of self-regulation. It is one's knowledge concerning one's own cognitive processes and products: the active monitoring and consequent regulation and orchestration of these procedures. Can metacognition be fostered?

Case (1978b) describes this metacognition as kinds of executive cognitive strategies that evolve as the children move toward the stage of concrete operations. He suggests that specific experience can exert a considerable effect on the acquisition of these executive strategies, and that practice, feedback, cue highlighting and strategy modeling can influence progress. In other words, young children can be taught to become more efficient problem-solvers.

Convergent Thinking. Convergent thinking is the narrowing down of a wide range of possible actions into a smaller and definitely ordered sequence. According to Guilford (1967) convergent thinking requires the satisfaction of a unique specification or set of specifications. This analytical thinking, or logic, is but a part of the thinking process, and can be regarded as convergent problem-solving, i.e. a series of strategies which looks for one definitive solution. Piaget considered the attainment of conservation as necessary for the development of logical thinking. A child's application of the principle of conservation is the criteria for identifying that child's cognitive developmental level as the stage of concrete operations (Travers, 1977). Therefore it would be expected that children who have not yet reached the stage of concrete operations would not have well-developed convergent thinking skills. The teaching of analytical thinking, or logic, has been the traditional approach to teaching thinking (De Bono, 1976). Much of the development of the executive strategies relates to analytical thinking, but it is not the whole picture.

Divergent thinking. Divergent thinking, the search for various and different solutions to a problem, has very close links with creativity, and indeed each term is often used to define the other (De Bono, 1972). Guilford (1967) introduced the interpretation of creativity as that which distinguishes the difference between convergent and divergent thinking. He described the divergent dimensions of thinking as fluency, flexibility, originality and elaboration. Some researchers raised the question of whether the fluency of ideas ought to be considered of any real value (Travers, 1977). A more recent study by Moran et al. (1982) asserts that there seems to be a relationship between the quantity of ideas and the quality of ideas, with more conventional and popular responses occurring earlier and more original responses occurring later in the sequence. The testing of creativity has continued to be a controversial area, with researchers expressing concern over such aspects as context effect, motivation effect and instructions to test-takers (Willhoft and Lissitz, 1982).

Programming. For the purposes of this study, Papert's definition of computer programming will suffice: "Programming a computer means nothing more or less than communicating to it in a language that it and the human user can both 'understand'" (1980, p. 5). The type of computer programming undertaken by young children is simple in nature and generally in the immediate mode, i.e. no real procedures are created or saved to disk.

Young Children's Problem-Solving

According to Piagetian theory, young children are natural problem-solvers. Klahr (1978) points out that observations of preschoolers demonstrate that they are often aware of constraints, facts, inferences, i.e. general problem-solving processes. Papert (1980) describes how they learn to speak, learn the intuitive geometry needed to get around in space and learn enough of rhetoric to get around parents - without being taught. They begin their lives as eager and competent learners and have to learn to have trouble with learning.

Young children are neophytes in most formal problem-solving situations. They lack skills and are also deficient in terms of self-conscious participation and intelligent self-regulation of their actions. Their insensitivity to their problem-solving potential is the result of a lack of exposure to such situations rather than age per se (Brown and Deloache, 1978).

They suffer from a limited memory capacity. The main difference between young children and mature memorizers is that the mature memorizers tend to employ a variety of mnemonic strategies whenever feasible. Young children lack both production and mediational skills in this area. Their skills can be improved with training provided the distance between the existing knowledge and the new information is not too great (Brown and Deloache, 1978).

Little children have a very limited attention span, in both breadth and depth. They reason from one particular instance to another particular instance, often unaware when their conclusions contradict one another (Almy, 1966). They tend to ignore parts in favour of the whole, or ignore the whole in favour of the parts, being unable to consider several aspects of a situation simultaneously (Ginsburg and Opper, 1969). Since they cannot reverse the direction of their thought, solutions to problems are either all right or all wrong (Almy, 1966).

There often is a great discrepancy between verbal skill and cognitive skill. According to Ginsburg and Opper (1969), children may fail to solve a problem when its solution requires verbal expression, yet be quite able to deal with the same dilemma on a practical, behavioural level. Children's actions, more than their words, give clues to their cognitive stage. Almy (1966) recommends direct observation of a child's functioning in a problem-solving situation, to derive many clues to actual function.

Young children approach intellectual tasks with strategies that are reasonable but oversimplified (Case, 1978a). Seigler (1978) says that young children's problem-solving behaviour can be explained in terms of inaccurate or incomplete encoding, and that their strategies are rule-governed, progressing from less to more sophistication with age. According to his research, three year-olds are intuitive and five-year-olds are systematic: that is, three-year-olds do not use rules to solve problems, four-year-olds sometimes do, and five-year-olds nearly always do.

Klahr (1978) studied the performance of three- to five-year-olds on a variety of well-defined tasks, such as the Tower of Hanoi puzzle. His results showed that age effects were clear, significant and monotonic. Planning was difficult to elicit, but notable was the absence of any seriously incorrect plan. The children either said what they were going to do correctly or said nothing at all. The very little children's strategies contained no tests; they were direct in their attempt to achieve the goal. Older children made tests before they moved, but moved directly, rather than generating new goals for further testing.

Steinberg (1980a) conducted a study to assess the processes by which kindergarten children tried to develop a correct strategy during repeated attempts to solve a problem. Most of them understood the need to change strategies, and did so at one time or another, but did not generate an appropriate one. She hypothesized that they may not have known that they should try to match the strategy to the problem. She concluded that kindergarteners are not likely to evaluate a strategy if a match answer is right, but that they may do so for a wrong answer, or a number of wrong answers. The key to their evaluation appeared to be feedback.

These studies point out that children need to be guided to generate and to assess their own problem-solving strategies. Flavell (1976) says that they are not very good at transferring strategies from one situation to another, but that they can be helped to develop this. The key to developing their metacognition is to help them to learn about how, where and when to store information and how, where and when to retrieve it as a means to a variety of real-life goals. The how and the where refer to a variety of storage and retrieval strategies and sources. The when is a child's growing sense that some situations call for active, deliberate attempts to learn and store and to retrieve and apply.

Children are naturally motivated to solve real-life problems. This motivation is part of their attempts to make sense of their surroundings. They are stimulated by ambiguity, novelty, and surprise. When they encounter something new, they demonstrate two phases of exploratory behaviour: first, what an object does (curiosity about properties), and second, what can be done with this object (personal sense of mastery and control) (Copple et al., 1979). This curiosity and the drive to achieve competence are the intrinsic motives for problem-solving and the exercise of problem-solving is its own reward (Bruner, 1966).

Bruner also describes the phenomenon of extrinsic problem-solving, which is that children in school spend an extraordinary time and effort figuring out what it is the teacher wants from them vis-a-vis the "hidden curriculum." They are quite skilled at finding out, another indication of their problem-solving potential.

To Foster Problem-Solving Skills

In order to develop problem-solving skills, young children need encouragement in remembering, ordering, classifying, reasoning and deducing from clues, and divergent thinking. They must also be taught to look and listen with precision and to think and talk about what they see and hear (Golick, 1976).

The development of problem-solving skills which apply over a broad domain of problems gives children a sense of mastery over the environment and a sense of their own worth as individuals. Using problems as they arise in the context of daily living is perhaps the most effective way of fostering children's skills in working through situations on their own (Copple et al., 1979). Flavell (1976) suggests devising problem situations for young children that are more natural and less "school type." Their efforts to generalize may be unsophisticated and incomplete, but encouraging their conscious awareness of the existence of strategies will only enhance their development.

Copple et al. (1979) suggest that in order to foster the development of problem-solving teachers need to create an environment that encourages children to think for themselves. They recommend the development of strategies and objectives for the following processes to aid preschoolers in their problem-solving efforts:

1. problem identification and observation
2. generation of hypotheses
3. testing of hypotheses
4. classification
5. awareness that hypotheses have been substantiated

This is further supported by Flavell's (1976) strong recommendation that we must encourage children to become self-conscious about their strategies and objectives.

Brown and Deloache (1978) urge that young children be made aware of metacognition and its universal applicability. Their very practical suggestion is to train children to stop, check and self-question before responding.

In a study of the problem-solving of grade one children, Resnick and Glaser (1976) found that the most powerful determinant of performance was to direct children to "look ahead", to plan. They suggest that strategies of planning ahead and considering an alternative goal may be a very powerful component of the problem-solving process for young children and that looking ahead may be generalizable over a variety of tasks. If so, it would improve their ability to learn on their own.

Steinberg (1980b), on the basis of her studies of kindergarten children interacting with computers, offers these recommendations for problem-solving activities:

1. Be sure that students understand the problem and that there are no other elements that allow students to guess correctly.
2. Feedback ought to be available at the level of the student's understanding.
3. Set conditions so that the student can figure out the essential details of a problem and discriminate between the essential and non-essential aspects.

Papert (1980) suggests that the role model provided by the teacher is a critical component to the development of problem-solving skills. A teacher's attitude toward errors should be that they benefit us because they lead us to study what happened, to understand what went wrong, and through understanding to fix it. A teacher also makes mistakes, but pursues a problem until it is understood. Teachers and learners together need to be engaged in true collaboration in this process of solving problems.

Spivack and Shure (1974) have approached the study of the social adjustment of young children through a problem-solving model. Their position is that preschool children need language and cognitive skills to solve problems and that they need to be taught how to use these skills in real-life interpersonal difficulties. They define the thought processes which bear a direct relationship to human adjustment: a sensitivity to the consequences and causes and effects in human behaviour, the ability to imagine alternative courses of action, and the ability to conceptualize the means to solve a problem. We need to teach children how to think, not what to think.

Computers and Young Children

Why consider the introduction of computers to such young children? Conlin (1981) makes the point that young children must be able to interact with the personal computer if they are to become at ease with the burgeoning use of microprocessors in today's world. She recommends four years of age as a good time to begin. Zeiser and Hoffman (1983) suggest that even at eighteen months of age toddlers are capable of meaningfully interacting with a computer; touching any key elicits a colourful screen response from the computer and the child begins to see a cause and effect.

Swigger and Campbell (1981) introduced a personal computer into a nursery school in order to explore the following questions: Are very young children capable of using a computer? Is it necessary to know how to read? Is it possible to teach young children how to control the keys? They made the assumption that since children learn almost everything from their environment, a computer could be introduced to this environment to accelerate acquiring certain learning skills. The results demonstrated that preschoolers were highly motivated to use the computer. They were attracted to the interactive process that is so similar to their first learning experiences. All children were able to master its operation, and none were afraid or incapable of understanding. The children seemed to accomplish more than their teacher's assessment appeared to indicate. Their conclusion was that young children can and will use computers, and they call for further studies into what activities with what techniques and with what effect.

What is the value of personal computers for young children's problem-solving? The computer itself presents a problem to young children: they must find the means to interact with it. Grimes (1981) suggests that children's self-concept as learners generally increase as they master the machine. Conlin (1981) points out that it is self-pacing and non-threatening, with speed and ease of feedback. The attractiveness of the graphics and sound capabilities is a strong motivator. It would also appear to offer a real opportunity to observe children's problem-solving strategies without dependence on their verbalization. Conlin also points out a social value with cognitive implications: she recommends that children work in pairs or small groups at the computer, that social interaction and cooperation is enhanced and that the amount of language the computer stimulates is phenomenal. This has particular relevance in that Travers (1977) points out that verbalization is a key component in the process of the transfer of learning.

In order to ensure that children are ready to join the "computer revolution," Bowman (1983) recommends a continued emphasis on play, on problem-solving, on self-directed learning, and on cooperation in learning. She suggests that "we must" investigate how to initiate computer literacy, and especially that "we must" be alert to inequalities in society.

Limitations. While the computer would appear to lend itself to encouraging discovery learning, it is doubtful that children can accelerate the development of their metacognitive functioning by interaction with computer alone. There is a strong need for a guiding teacher and the interaction with peers. Computer-assisted learning cannot stand alone, but must be closely tied to teacher-pupil and pupil-pupil interactions (Papert, 1980). Ziajka (1983) emphasizes that it is simply another curriculum material whose value is dependent on how it is used by sensitive and knowledgeable teachers.

Brady and Hill (1984) present a thoughtful review of research exemplifying the kind of questions early childhood researchers are asking and they also identify directions for parents and teachers. They fear that the computer is being looked upon as the latest solution to current needs and dilemmas. They also point out that researchers have yet to answer the major question: What are appropriate experiences on personal computers for young children?

Because of the inability of young children to read, there is a real need to limit the variety of responses required on the conventional keyboard. One solution would be to devise a simpler keyboard such as that devised by Lauzon (1982), but in a real school situation, it is often easier to work within the constraints of what is available. Piestrup (1981) found that dividing the keyboard into quadrants worked well for nursery school children, and Nelson (1981) reported that preschoolers were able to find and use a limited number of keys effectively.

<u>Computer-Assisted instruction (CAI).</u> Much of the commercially-made software for young children is of questionable value, although Swigger (1983) reports improvement in the overall quality among the increasing quantity. She offers three caveats: add two years to the age level indicated, all require the help of adults, and none are child proof.

Piestrup (1981) devised a computer program using coloured graphics, music and digitized voice to present reading readiness concepts in a nursery school. This was later developed commercially as "Juggles' Rainbow." The purpose of the study was to show the possibilities for early skill learning with the personal computer. The study was conducted in an activity-centered nursery school, where the computer was offered as one of the options. The following outcomes were observed:

1. The computer and the program were enthusiastically accepted by all involved.
2. The children under three did not seem to make the connection between a keypress and an effect on screen. For three years and older there were several layers of mastery.
3. The average improvement on basic prereading skills was considerable.
4. The children successfully used the computer within their ability.
5. The learning of concepts could not be attributed to the computer alone, but there is a question of a possible acceleration of acquistion that needs to be investigated.
6. More girls than boys asked for turns and persisted in getting them.
7. Non-English children were more hesitant in their

approach to the computer.

8. The computer was never out of use.

Computer-Assisted Learning (CAL). The real potential of the personal computer in education appears to be its ability to foster thinking skills. These are essentially regarded as the main goal of education. They are rarely directly taught, but are assumed to be the outcome of all teaching and learning activities (De Bono, 1976). Steffin (1983) points out that public school objectives, the conventional classroom environment and traditional media work together to influence young learners toward convergent thinking styles, whereas the computer will tend to foster divergent thinking. Hines reports that five-year-olds are able to use the computer as a tool for thinking, and that Logo provides a particularly good environment for that use (1983).

The Value of Logo. The development of Logo, a programming language designed especially for young learners and based on Piaget's model of children as builders of their own knowledge systems, offers greater possibilities for the development of problem-solving skills within computer-assisted learning. In particular, its Turtle Graphics capability presents a visual display that can be readily programmed by young children.

According to Papert (1980), the child should program the computer, not the computer program the child. The most powerful use of the computer, in terms of the child's learning, is for the child to have to teach the computer what to do. Nelson (1981) says that this process lures children into learning by exploration and discovery. He describes the work of a nursery teacher who created a series of procedures taking a single-key response to introduce Logo to her preschoolers. It became a small step from this to the direct use of Logo.

Lawler (1982) reports a case study of a three-year-old introduced to Logo. One outcome of this was the creation of a new kind of pre-reader's ABC book, where the keypress of a certain letter called up the relevant picture: that is, to key A calls up an apple to be drawn on the screen. This relationship is exactly opposite to the traditional one: instead of the picture being the access to the letter, the letter is the access to the picture. Children here are in control of their own learning. Although the technology is artificial, it makes possible a more "natural" absorption of knowledge than learning to read from the printed page. Personal computers put reading and writing together from the start. Learning the alphabetic language occurs in a similar way to the infant's learning of vocal language. The computer here plays the role of a responsive, interested person who listens and reacts. This ties in very well with a psycholinguistic approach to teaching reading, where reading is regarded as a problem-solving activity (Smith, 1971).

A preliminary study by the author (1983) indicates that Logo Instant, a procedure which takes single-letter commands to create simple Turtle Graphics, is very much enjoyed by Kindergarten children. Their simple programming activities demonstrate an understanding of the principle of reversibility that belies the results of conventional tests of their cognitive developmental level, suggesting the possibility that the computer may offer a more accurate method of assessment than conventional measures. Hines (1983) reports success with Kindergarten children using a very similar form of Logo.

A study by Gregg (1978) presents some guidelines for introducing programming with Turtle Graphics. He conducted a year-long study with four- and five-year-olds using a robotic turtle with three control buttons: forward, left and right. His hypothesis was that, with a fairly minimal amount of experience, the children would be able to make appropriate discriminations, learn appropriate names and then devise complex sequences of button-presses (sentences). He found that children need to go through a number of identifiable stages in learning to solve Turtle problems, and that most difficulties emerged when the concepts necessary for the tasks had not yet been mastered. In the first stage the discovery was that the buttons have some effect on the Turtle. In stage two there is a differentiation of the "forward" button from the other two. The third stage is achieved when it is recognized that the turn buttons have separate functions. In stage four the children understand that the turn buttons make Turtle turn on its axis as opposed to changing direction and going forward. The last stage is a full understanding of the Turtle task, and is much too difficult for a pre-operational child to grasp. Egocentric children have trouble mapping an external symbol system onto their representation of a spatial array. Five-year-olds, who are in transition into the stage of concrete operations, are able to perform an "exocentric" representation, an image of objects and their spatial relationships, and so should be able to benefit from Turtle Graphics activities. The implication here is that we must be on guard not to expect levels of programming that are beyond the child's developmental level.

Solomon (1976) describes these powerful aspects of the Logo culture:

1. a strong anthropomorphization, creating a metalanguage, with "playing Turtle" as an important problem-solving tool
2. learning to see projects as research enterprises
3. debugging as opposed to making mistakes
4. a "bug collection": a checklist; bugs and debugging as a part of life (developed algorithms as tools to

solve problems)

Solomon and Papert (1976) report a case study of a six-year-old child in follow-up activities after being introduced to Logo. The authors were quite directive in their approach; it was not random exploration. They recommend that in problem-solving activities with Logo that it is important to follow the child's lead in programming, but to be quite ready to become involved as a guide and a tutor.

The Social Concerns. We must be aware of the anti-social aspect of the isolation of the learner with the machine, away from any interaction with fellow human beings. This is particularly critical with young children, who are just developing their social skills.

Grimes (1981) points out that the computer has influence in the affective domain, both in the development of self-concept and of interpersonal interactions, and suggests pairing students of different ability levels. Many authors (Nelson, 1980; Conlin, 1981; Piestrup, 1981) address this issue, stressing its importance on many levels, including interpersonal problem-solving.

That there is a possibility to deal with interpersonal problem-solving skill development at the computer presents an argument for the introduction of the computer into the early childhood classroom, especially for children from disadvantaged backgrounds. This group is at high risk in any educational situation, and Spivack and Shure (1974) argue that if these individuals can be helped to develop a cognitive problem-solving style for real-life problems and to generate their own ways of solving the typical interpersonal problems that arise, they will cope better and manifest improvement in behavioural adjustment. They need language and cognitive skills to solve these problems, and working in groups at the computer offers an opportunity to exercise and develop these skills.

Implicit in all of this is, of course, the mediation of a concerned and caring adult. The computer on its own cannot achieve any of this, but is a tool with a powerful potential for motivating those who interact with it.

The Focus of the Study

Since the use of computers by kindergarten children is still a novel field, it would be helpful to attempt to identify some of the factors that could have importance in terms of computer-assisted learning for this group. Since the use of Logo by kindergarten children is relatively unexplored, it would appear that a case study of a group of kindergartners working with Turtle Graphics could be of particular value.

Wise, Nordberg and Reitz define a case study as a collection of available evidence that helps explain an individual or an entity; its purpose is to discover underlying themes to understand the individual or entity in its own particular situation (1967).

Johnson (1977) outlines the steps in the process. First is to define the problem; what it is that students are expected to do. Next is to describe the factors that influence this expectation. Then data must be collected in some systematic fashion, describing the principal features of the case, citing observations, interventions and changes. The researcher must then look for patterns and try to relate them to theory, which can lead to tentative conclusions and further hypotheses.

Behr (1973) describes a case study as research only when there is a comprehensive interpretation of the responses in the total situation. It is by modifying the conditions as the need arises that the investigator can test the genuineness of the responses and gain insight into the subjects' thought processes. He sees the purpose of all this as a means of identifying meaningful factors in a learning situation and to generate hypotheses for further study.

The hypotheses for this study are as follows:

1) that kindergarten children can cope with computers, managing the hardware and working positively with the software;

2) that they will react favourably to certain types of software;

3) that the nature of their social interactions will depend partly on the type of software;

4) that kindergarten children will be able to set elementary programming tasks for themselves while using Logo;

5) that they will use strategies to solve their programming problems with Logo;

6) that computer-assisted learning will enhance the general problem-solving skills of kindergarten children;

7) that Logo will enhance the general problem-solving skills of kindergarten children to a greater extent than other forms of computer-assisted learning.

CHAPTER III

METHOD

There are two main parts to this study: the first being a case study, with observation of the subjects leading to categorizations of behaviour and then the development of more formal tools for the observation of behaviour, and the second being an experiment which compared the performance of three groups, each using a different kind of educational software.

Subjects

The study used forty-four kindergarten students in a French Immersion urban public school, with a fairly evenly distributed blue-collar and middle-class population. There were sixteen subjects in the Logo cell, fifteen in the CAL cell and thirteen in the CAI cell (all randomly assigned).

Design of the Case Study

The behaviour of all three treatment groups, particularly with regard to their response to task requirements and their social interactions, was observed throughout the study and informally noted.

The progress of subjects who were working with Logo was more rigorously observed over the twenty weekly treatment sessions. Observations from a previous study were used initially as a guideline for expected observations (Munro-Mavrias, 1983).

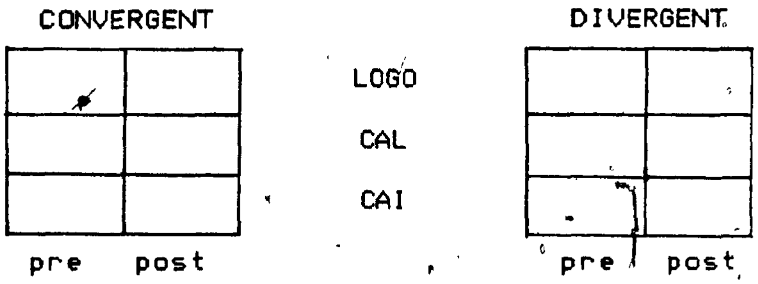
The period before Session 7 could be considered as the training and practice sessions, in which the children were gaining skill in operating the Logo turtle. During this time patterns of behaviour relating to goal-setting, programming style, types of errors and strategies for error debugging were observed which led to the development of the checklist. No record of the children's activity was kept.

The period from Session 7 onward falls into three distinct parts: Sessions 7 to 11 in which the children were given little or no direction, Sessions 12 to 15 in which teacher-suggested goals were introduced and in which the children were given pencil and paper to design their own tasks, and Sessions 16 to 20 in which the children were once more given little or no direction, but materials from the previous period were readily available.

For the case study, programming behaviours and strategies to solve programming problems were recorded on a specially-developed checklist from observation of the recorded pupil action data on the computer disks, and from audio recordings of the subjects' comments as they worked. This offered a basis for analysis of the types of behaviours and strategies used, and their frequency, to provide some insight into the "programming" practices of kindergarten children.

Design of the Experiment

The experimental study consisted of two 3 x 2 mixed experiments: one for each level of the within-subject problem-solving factor, as illustrated below.



There were groups for the three levels of instructional strategy (Logo, CAL or CAI); subjects were assigned to these groups according to a random number table. The two kinds of problem-solving skill, convergent thinking and divergent thinking, were tested for effects over time by means of analysis of covariance. Because of the small number of subjects in each group it was decided that an analysis of covariance, with the pretest scores as covariates, offered the most sensitive measure to discriminate differences between groups.

Materials

Two Apple II+ computer systems were used, along with a portable audio cassette recorder with a built-in microphone.

The CAI group served as a control, using software designed for drill and practice of letter and number recognition, which served in this study simply as a means of computer familiarisation. The activities were part of a commercially prepared software package, Early Games for Children from Counterpoint Software.

The CAL group used two software packages designed to teach problem-solving skills. Gertrude's Secrets from The Learning Company teaches logic skills: ordering, sequencing, attribute recognition, and hypothesis-testing, i.e. elements primarily of convergent thinking, but with some possibilities for the exercise of divergent thinking. Facemaker from Spinnaker Software allows the child to make a great variety of choices and decisions in creating a series of funny faces and then animating these in different sequences, thereby encouraging imagination and fluency and originality of ideas, qualities of divergent thinking.

The Logo group used Logo language disks that have been modified to start up with Logo INSTANT immediately upon booting and that will automatically save the subjects' programming efforts. INSTANT is a procedure which takes single keystroke commands without a need for an accompanying numerical input to control the movement of the cybernetic turtle nor for a subsequent RETURN keypress. F causes it to move forward ten increments, R and L cause it to turn right or left thirty degrees, and C clears the screen.

The personal computer was introduced to the subjects by means of an instructional module designed by the author, The Computer Goes to Kindergarten, with which children become familiar with the computer and its operation and learn to get the system up and running with the required software.

Development of the Checklist

The programming behaviours checklist (see Appendix A) was developed in order to be able to categorize and then record and subsequently analyze the types of behaviour exhibited by the subjects as they programmed the computer. Categories of behaviour were: Nature of Goal, Programming Style, Types of Errors, Debugging Action and Results. The checklist was freely adapted from Chait (1978) and Greenberg (1984). Chait developed categories for classifying programming behaviours of elementary-age children using Logo in an observational study. Although many of these were not appropriate for use with the much younger subjects in this study, the error types and debugging actions provided a starting point. Greenberg developed a checklist to evaluate Logo programming activities, again with older elementary school children. Her categories of the nature of the goal and of programming style were helpful in providing an orientation for observation.

Nature of Goal. The children were not asked during sessions if they had set a goal for themselves, therefore it had to be deduced from the evidence in the protocols. If a pair announced an intended goal it was noted and entered on the protocol later. Output judged to have a non-existent goal had a general appearance of scribbling or doodling, with no identifiable pattern. Output judged to have a vague goal was that in which a pattern could be perceived at some point: sometimes initially, in which case it became obscured by haphazard programming, or sometimes later, in which case the preceding haphazard programming seems to have suggested a goal. Very often the protocol revealed a very clear and recognizable design or pattern which was judged to be a fixed and self-chosen goal. The introduction of the task cards led to the category of fixed and teacher-set goal. If a goal was related to a previous one had to be deduced from the protocols; this category offered some clues to programming style as well as to persistence in trying to solve a problem.

Programming Style. This category of behaviour pertains to evidence of deliberate programming intent, and relates very closely to goal-setting behaviour. A haphazard, exploratory style would generally indicate that there is little evidence of a goal, although even with a fixed goal the children often started out with very "random" programming behaviour.

Types of Errors. Errors were only counted as such if they were not corrected or debugged. Redundant commands were counted only in those protocols which indicated some evidence of a goal.

Debugging Action. Three of the debugging strategies, CLEARSCREEN/restart, reverse commands, and complete rotation, occurred spontaneously as the children programmed, while the fourth, UNDO, was introduced by the instructor in Session 14 to encourage the use of more debugging. CLEARSCREEN/restart is self-explanatory and the least powerful of any debugging actions since it involves throwing out the good with the bad and starting all over again. The reversing of commands occurred with both length and angle errors and involved doing the exact opposite of what had just been programmed: in the case of angles it simply involved turning left as much as just having turned right and vice versa, and with length of line it involved turning around 180 degrees retracing the path the right amount and then turning around 180 degrees once more, a fairly complex manoeuvre. Complete rotation involved turning 360 degrees in order to correct too large an angle. UNDO is a procedure in which the output is erased from the screen and then redrawn with the last step deleted.

Result. A complex figure was defined as one in which there are at least three distinct lines and two distinct angles. A goal was considered to be achieved if it closely resembled the task card or drawing or what the children had declared to be their goal. The number of commands was considered to be an indicator of attention to task and persistence toward goal.

Validation of the checklist. Once the checklist was considered to be in a fairly usable form, it was validated in Sessions 5 and 6, first by recording from direct observation and then from protocols derived from print-outs of programming activity. The criteria for a "valid" checklist were: that it should provide data relevant to the hypotheses; that it should offer an ordering of the behaviours observed; that it should contain clear and distinct categories for all the behaviours observed; and that it should be easy to use.

Choice of Test Instruments

In choosing the instruments to measure the children's thinking skills, two concerns expressed by the participating school had to be heeded: that the time required for administration of the tests should be brief, and that the process should not provoke anxiety reactions in the children. This therefore had to be taken into consideration in the selection process along with the concern of the study for validity and reliability of the instruments themselves.

The Torrance Thinking Creatively in Action and Movement Test (TCAM) was used to measure divergent problem-solving. The Goldschmid and Bentler Concept Assessment Test: Conservation was used to measure convergent problem-solving. Each of these tests can be administered more than once without concern for test practice effects with children of kindergarten age, as long as there is a period of a few months between testings.

To test for convergent thinking. Goldschmid and Bentler (1968) worked to develop a standardized form of the Piagetian conservation experiments that would be appropriate for North American children. Their purpose was to develop an efficient tool to provide a normatively adequate, brief and practical assessment of the principle of conservation.

The test offers six different conservation activities. The manipulative materials provided (plasticine, popping corn, toy eggs and egg cups, for example) are simple and familiar to a child, and the test can be administered to an individual child in approximately twenty minutes. Each item is scored on both behaviour and explanation, with a point being awarded for the correct response in each sector, for a possible total score of twelve points.

The authors report high internal consistency (.95 - .97) in data from studies with lower- and upper-middle class children in kindergarten and first and second grade. A test-retest reliability coefficient of .94 was reported.

Valid criticisms have been made of the test's limitations. Smock (1970) expressed concern in three areas: the test was not stringent enough, there were not enough instances for each type of conservation, and there was too much opportunity for variation in the interpretation of verbal responses. DeVries and Kohlberg (1969) raised the problem of the transitional period between non-conserving and conserving where there is vacillation and where answers may be equivocal. They ask if a single conservation response for a content area may serve as a valid criterion for conservation in that area.

The purpose of this current application of the instrument was to provide an indication of the subjects' developmental level toward logical thinking. It appeared that it offered a possibility to do this, although its limitations must be borne in mind. While the very early ceiling effect is a problem, since we cannot distinguish any development if a child is already conserving, it should quite clearly indicate definite non-conservers and those heading into transition or beyond, and at the same time fulfil the need for a brief and non-threatening test situation.

<u>To test for divergent thinking.</u> Torrance's work in developing standardized tests of creativity (Torrance Tests of Creative Thinking) came directly out of Guilford's work (1967). Because of their dependence on verbal facility among other things, many researchers, including Torrance himself, have been concerned with the difficulty of using them effectively with very young children. Moran et al. (1982) strongly recommend that appropriate measures for preschoolers should be composed of materials with which the children have hands-on familiarity.

In response to these concerns Torrance developed <u>Thinking Creatively in Action and Movement</u> (TCAM), especially for young children. The test yields three distinct scores on the dimensions of fluency, originality and imagination. This test administered individually takes approximately twenty minutes per subject, does not require verbal responses, and uses materials and activities that are a part of the child's everyday world (1981).

Because of its relatively recent publication, there is a scarcity of critical evaluation in the literature. Most of what follows is gleaned from the test manual, with one independent report which includes Torrance in the list of authors.

The manual reports a test-retest reliability coefficient of .84. That it makes use of familiar materials from the child's environment and that it does not require verbal responses would appear to satisfy the content validity requirements described by Moran et al. (1982). Again, according to evidence offered in the manual, TCAM correlates positively with other tests of creativity developed for young children and is unrelated to measures of intelligence, race, sex, SES, and previous school attendance.

A study by Reisman et al. (1981) provides support of the test's construct validity for divergent thinking. TCAM was used as a predictor of performance on three different tests, a traditional Piagetian conservation measure used as an example of convergent thinking, a modified Piagetian test requiring divergent responses, and a math readiness test requiring mostly divergent responses. The hypothesis was that the scores on TCAM would predict achievement on all three measures, and would better predict for the modified Piagetian test than for the traditional Piaget measure. The subjects were thirty-four multiracial, varied SES preschoolers from thirty-four to sixty-eight months of age. The results yielded no significance for the traditional Piagetian test. Twenty-six per cent of the variance in the modified Piagetian test and thirty-five per cent of the variance in the math readiness test was explained by TCAM.

Although author-reported reliability and validity data is open to questions of bias, although there does not appear to be any completely independent evidence of the value of this measure, and although the whole area of creativity is subject to controversy, the instrument has value for this context. It should provide some indication of the scope of a child"s divergent thinking skills, while also fulfilling the need for a brief and non-threatening test situation.

Procedure

Permission to conduct the study was obtained first from the principal and involved teachers, then from the parents. All subjects were administered the pretests as soon as parental consent forms were obtained and the groups were formed.

As soon as the testing was completed, all groups received the treatment, four subjects at a time, for twenty minutes, once a week. In all three conditions, two children shared a computer, and the partnerships were rotated each week among the four children within a group.

In all conditions, the children started up the software and worked together through the activities. It was made clear that collaboration and cooperation were expected. The instructor was available for assistance if needed.

In the Logo condition, once subjects had achieved proficiency with the commands, they were encouraged in a very general way to set their own tasks. Most children were not especially able to set their own tasks, so that instructor-designed "puzzles" (task cards with simple line drawings) were introduced at the Session 12, and then from Session 15 onward the children made use of paper and crayons to design their own tasks as well. The instructor also attempted to structure the activity so as to bring about collaboration once the task cards were introduced. It was suggested that one partner hold the task card and to verify that the screen image was evolving correctly while the other partner did the actual keyboard work. Once again, the instructor was available if needed, but no further formal teacher intervention took place. The procedure UNDO was introduced at Session 14 to encourage debugging activity. Any relevant anecdotal information to support the data collected on the disks was gathered, and all of this was analysed and recorded on the checklist at the end of each weekly session. (Since there was only one observer throughout the study, there was no means to obtain inter-rater reliablity.)

Immediately after the twenty weeks of treatment were completed, the posttests were administered.

CHAPTER IV

RESULTS

The Case Study

The data from the checklists falls into three distinct periods: Sessions 7 to 11 in which the children were given little or no direction, Sessions 12 to 15 in which teacher-suggested goals were introduced and in which the children were given pencil and paper to design their own tasks, and Sessions 16 to 20 in which once more the children were given little or no direction but materials from the previous period were readily available.

Time on task. The relationship between an episode and time on task must be clarified here. The children in each programming pair tended to alternate command of the keyboard from episode to episode. The one currently keyboarding was observed to be totally engrossed in the task, while the other sometimes was and sometimes was not. But at least one of the pair was devoting total attention through the duration of the episode. Therefore counting episodes per session is a reasonable measure of time on task.

Figure 1 indicates the number of episodes per twenty-minute session and the number of commands per episode. An episode is defined as the programming activity from one CLEARSCREEN command to the next.

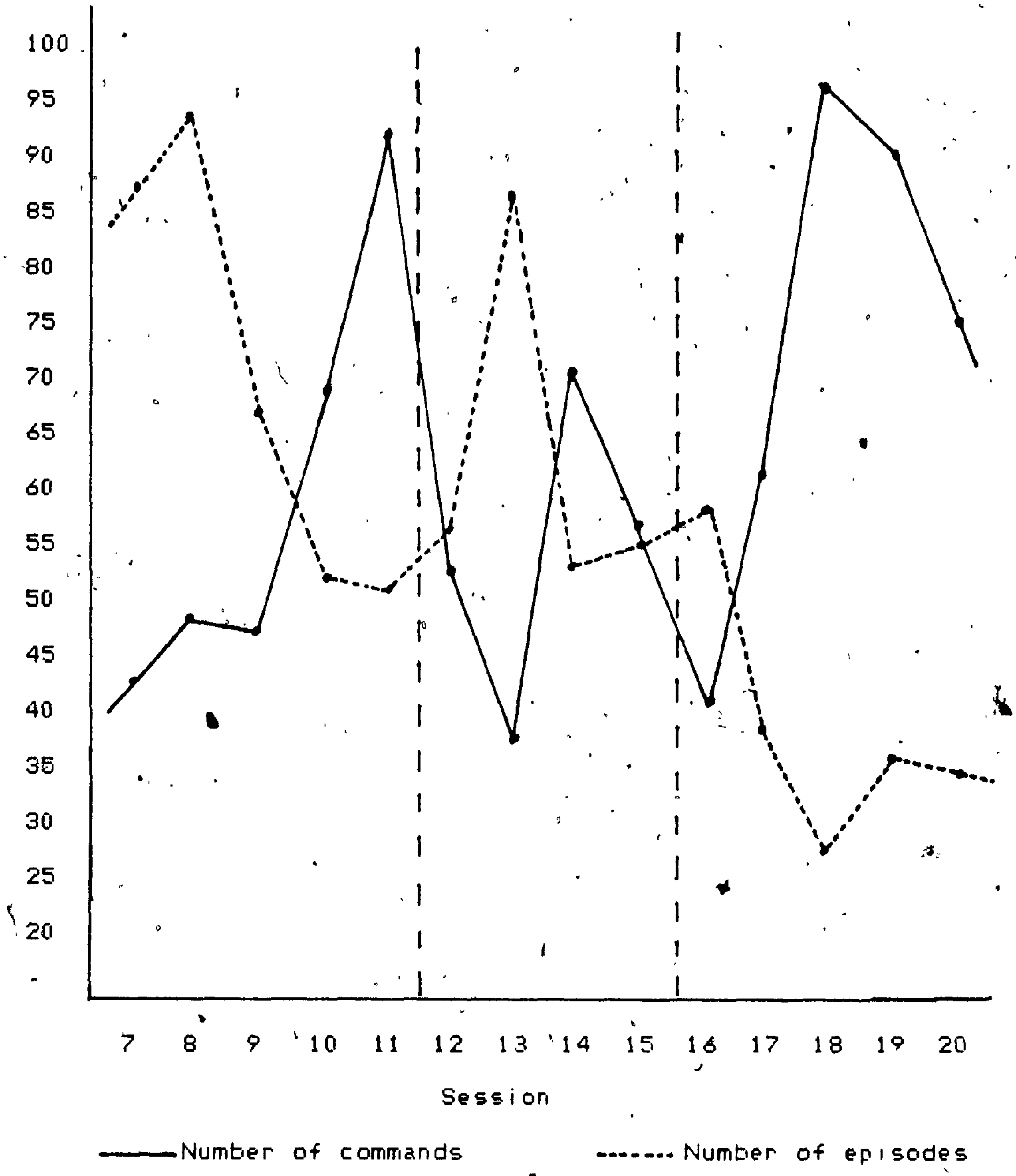


Figure 1. The average number of commands per episode and the number of episodes per twenty-minute session.

Episodes per session became fewer as time progressed and each episode became longer, demonstrating that the children were spending more time on task. Table 1 also indicates time spent on task by indicating the number of episodes terminated by an immediate CLEARSCREEN, that is with no other commands intervening.

Table 1

Episodes Terminated

by an Immediate CLEARSCREEN

Session	Total	n Terminated	%
7	87	29	33
8	94	28	30
9	67	13	24
10	52	3	6
11	51	14	28
12	54	5	10
13	86	4	5
14	52	7	14
15	54	0	0
16	57	9	16
17	36	0	0
18	25	1	4
19	34	0	0
20	33	2	6

A dependent, one-tailed t-test of the means for Sessions 7-11 and Sessions 16-20 was significant at the .05 level (+3.55). This supports the other evidence that the children became more focused on the task as they progressed through the treatment. Table 2 indicates the number of simple and complex figures created by the children's programming activities.

Table 2

Number of Simple and Complex Figures per Session

Session	n	Simple %	n	Complex %
7	27	47	31	53
8	38	58	28	42
9	25	46	29	54
10	22	45	27	55
11	14	38	23	62
12	4	8	45	92
13	31	38	51	62
14	12	27	33	73
15	25	46	29	54
16	17	35	31	65
17	15	42	21	58
18	4	17	20	83
19	4	12	30	88
20	7	23	24	77

A dependent, one-tailed t-test of the means for complex figures for Sessions 7-11 and Sessions 16-20 was not significant at the .05 level (-1.02), although there appeared to be improvement overall in the complexity of the figures produced.

Goal-setting. Table 3 indicates the amount of the different types of goal-setting behaviour in each session.

Table 3

Number of Types of Goal-setting Behaviour

Session	Non-existent	Vague	Fixed-self	Fixed-teacher
7	50	13	24	0
8	80	5	9	0
9	37	16	14	0
10	21	21	10	0
11	30	3	18	0
12	7	0	0	47
13	4	3	7	72
14	7	1	1	43
15	0	1	48	5
16	9	0	47	1
17	0	0	27	9
18	1	0	18	6
19	0	0	17	17
20	1	0	28	4

A dependent, one-tailed t-test of the means for Sessions 7-11 and Sessions 16-20 for the non-existent type was significant at the .01 level (+4.05) and for the fixed, self-chosen type was significant at the .05 level (-3.51). It is very clear that initially most children were not able to spontaneously set goals for themselves, and that goal-setting behaviour became much more common after some instruction. The children's success rate in achieving their goals is set out in Table 4.

Table 4

Number of Goals Achieved

Session	n Achieved	%
7	10	42
8	8	89
9	12	86
10	6	60
11	8	44
12	25	53
13	27	34
14	20	46
15	11	21
16	8	17
17	14	39
18	13	54
19	24	68
20	9	29

Table 5

Number of Episodes Related

to Previous Episodes in Session

Session	n	Related %	n	Unrelated %
7	18	23	62	77
8	17	20	69	80
9	14	23	48	77
10	9	20	36	80
11	10	23	34	77
12	20	42	28	58
13	34	44	44	56
14	21	48	23	52
15	34	74	12	26
16	35	70	15	30
17	16	53	14	47
18	5	29	12	71
19	8	31	18	69
20	16	64	9	36

Although the number of episodes related to previous episodes remained pretty much the same for Sessions 7-11 and Sessions 16-20 with no statistical difference (t = -.55), the number of episodes per session was decreasing overall. There were fewer unrelated episodes per session. A dependent, one-tailed t-test of the means for unrelated episodes was significant at the .01 level (+5.32). This would signify that the children became more persistent in trying to solve their problems in order to achieve their goals.

Programming style. Table 6 indicates the number of each type of programming style per session.

Table 6

Programming Style

Session	Haphazard	Contingent	Planned
7	27	29	2
8	54	0	12
9	35	18	1
10	39	9	1
11	19	18	0
12	2	43	4
13	10	69	3
14	2	43	0
15	26	20	8
16	8	38	2
17	1	28	7
18	2	16	6
19	0	28	6
20	9	20	2

Dependent, one-tailed t-tests of the means for Sessions 7-11 and Sessions 16-20 were significant, at the .01 level for the haphazard programming styles (+4.09) and not significant for contingent (-2.03) and planned programming styles (-.75). As the children became more proficient at programming, they left off using a haphazard style and used a contingent style for the most part. Although they infrequently planned ahead what they would program, the frequency with which they did so increased somewhat after instruction in goal-setting.

<u>Types of errors.</u> Table 7 indicates the number of these errors overall and Table 8 indicates the number of these errors per session.

Table 7

Number of Types of

Programming Errors Overall

Type	n	Overall %
Length	173	10
Angle	136	8
R/L confusion	443	27
Redundant commands	921	55

As is consistent with their developmental level, the children had a great deal of trouble to distinguish right from left. The large number of redundant commands is probably indicative of two things: difficulty in attending to task and a lack of a strategy to solve a problem.

Table 8

Number of Types of Errors Identified per Session

Session	Length	Angle	R/L confusion	Redundant
7	10	10	25	79
8	0	0	19	180
9	8	4	24	91
10	5	4	24	65
11	7	8	30	179
12	31	13	61	38
13	23	13	71	28
14	21	3	45	54
15	11	8	34	32
16	1	15	18	68
17	6	2	22	17
18	17	21	26	15
19	23	21	28	31
20	10	14	16	44

Dependent, one-tailed t-tests of the means for Sessions 7-11 and Sessions 16-20 showed no significance at the .05 level for length errors (-1.23) and R/L confusion errors (+.69), and significance for the angle errors (-2.97) and redundant errors (+2.89). Length and angle errors increased over time, angle errors significantly. This means that as the children became more task-oriented the probability of committing these types of errors increased. Right and left confusion persisted at more or less the same level throughout. Redundant commands decreased significantly over time, which is also consistent with greater orientation toward task.

Debugging strategies. Table 9 indicates the amount of use of debugging strategies overall and Table 10 indicates the number of times they were used in each session.

Table 9

Amount of Use of Debugging Strategies

Type of Strategy	n Use	%
CLEARSCREEN/restart	184	17
Reverse commands	424	38
Complete rotation	41	4
UNDO	459	41

Although the most popular debugging strategy (UNDO) was teacher-supplied, the persistent use of the reverse command strategy showed that the children can develop a strategy to solve a problem and use it effectively.

Table 10

Number of Debugging Strategies per Session

Session	CS/restart	Reverse	360 rotation	UNDO
7	6	10	6	0
8	1	0	0	0
9	0	6	2	0
10	6	5	3	0
11	2	23	0	0
12	18	11	9	0
13	30	37	3	0
14	16	30	1	106
15	37	44	0	52
16	34	41	0	48
17	17	40	8	71
18	4	56	2	59
19	8	60	6	63
20	6	61	2	60

Dependent, one-tailed t-tests of the means for Sessions 7-11 and Sessions 16-20 were significant at the .05 level for both the CLEARSCREEN/restart and reverse commands strategies (-2.17 and -9.93, respectively), while the 360 rotation strategy was not significant (-.62). Debugging strategies were already in effect before UNDO was introduced, but debugging as a means of solving a programming problem increased considerably after its introduction, suggesting that it may have enhanced the acceptability of debugging as an appropriate behaviour.

General Observations. There were some interesting behavioural observations that suggest differential development of problem-solving abilities of a general nature among the three treatment groups. The aspects of the Logo treatment group were dealt with at length above.

The CAI control group's software, with its large variety of drill-and-practice activities, did not present much of a challenge to the children. They very quickly became bored with the software and frequently asked if they could have some different programs, although they always expressed enthusiasm to come to the computer room. Their way of dealing with the simplistic nature of the games was to add their own complexity to them. Creatively they invented little chants, actions and songs to entertain each other and created guessing games of varying appropriateness.

The CAL group, which worked with software specifically designed to teach thinking skills, worked best overall. There was the least evidence of frustration and the attitude of the children was very positive. Gertrude's Secrets was structured in such a way that the children were led to make hypotheses and test them out, which they appeared to do with remarkable accuracy. With Facemaker the children appeared to be motivated to produce as many different possibilities as they could. A lot of verbal collaboration took place with both software packages, much more than with the Logo activities.

There were differential social interactions among the three groups. The pairs in the Logo treatment often seemed to be working in a parallel fashion instead of working together, so that in many cases an individual was working at the computer, with the partner waiting to have a turn next. (It was found possible for the teacher to structure the activity so as to bring about collaboration once the task cards were introduced. It was suggested that one partner hold the task card and to verify that the screen image was evolving correctly while the other partner did the actual keyboard work.) There seemed to be the most negative interactions with Logo; there was a lot of complaing and arguing. This parallel turn-taking behaviour was also quite common in the CAI group. There was a considerable amount of negative interactions, such as squabbling, complaints of boredom and arguing for turns. The pairs in the CAL group seemed to genuinely collaborate and cooperate as they worked their way through the activities, celebrating their successes and showing very little evidence of interaction that could be construed as negative.

The Experimental Study

The means and standard deviations for all treatment groups are given in Table 11. The divergent thinking measure is presented in two ways: each creativity factor is presented separately as recommended, but also included in the table is a combined score for the creativity factors.

Table 11

Means and Standard Deviations for All Treatment Groups

		CAI n=13	CAL n=15	LOGO n=16
Convergent (Conservation)	Pretest	6.4 (4.1)	4.5 (4.0)	5.6 (4.6)
	Posttest	9.5 (3.3)	7.7 (4.7)	8.1 (4.4)
Divergent (Fluency)	Pretest	23.2 (13.4)	20.9 (10.7)	21.6 (19.8)
	Posttest	25.5 (11.1)	30.2 (13.8)	33.1 (19.4)
Divergent (Originality)	Pretest	31.9 (24.1)	29.4 (19.5)	29.1 (28.4)
	Posttest	36.7 (18.3)	43.9 (25.8)	53.8 (37.2)
Divergent (Imagination)	Pretest	17.4 (4.8)	16.9 (6.5)	18.7 (4.0)
	Posttest	19.2 (4.6)	20.8 (2.6)	20.8 (4.2)
Divergent (Combined Fluency, Originality & Imagination)	Pretest	21.4 (24.8)	22.9 (23.3)	25.3 (28.8)
	Posttest	24.0 (25.2)	32.4 (32.1)	39.1 (40.9)

Note: Standard Deviations are enclosed in brackets.

These results compare favourably with those reported by the authors of the tests: Goldschmid and Bentler report a mean of 6.89 and a standard deviation of 4.85 for the conservation score (1968); Torrance reports means of 27, 33.6 and 20.1 and standard deviations of 14.8, 25.6 and 6.9 for fluency, originality and imagination (1981).

Analysis of the pretest scores indicated the presence of strong correlations (p < .001) between fluency and originality, fluency and imagination, and originality and imagination. This calls into question Torrance's asssertion that these are unique and separate factors in this test. There is also a significant correlation (p < .026) between the imagination factor and conservation. See Table 12 below. This suggests that there is some overlap in that which is measured by the two tests.

Table 12

Correlation Matrix of Pretest Scores

	Fluency	Originality	Imagination
Conservation	.13 (p=.392)	.16 (p=.311)	.34 (p=.026)
Fluency		.90 (p=.001)	.59 (p=.001)
Originality			.61 (p=.001)

On the basis of these findings a principal components analysis of the pretest scores was carried out as a means to reduce dimensionality (Gnanadesikan, 1977, pp. 7-15). Two principal components emerged which accounted for 87% of the total variance in the pretest. The loadings were such that no rotation of the factors was necessary in order to identify variables (see Table 13). The first one corresponded to the three divergent dependent variables, fluency, originality and imagination, and accounted for 62% of the variance. The second one clearly corresponded to the convergent dependent variable, conservation. On this basis it can be concluded that there seem to be only two dependent variables, one being conservation and the other composed of fluency, originality and imagination. From the factor loadings it can be concluded that, these three being basically equal, the second dependent variable is obtained from adding the three scores together as a combined measure for divergent thinking. When the same techniques were applied to the posttest scores, the same pattern of loadings is found.

Table 13

Factor Loading of Principal Components Analysis

	Factor 1	Factor 2	Factor 3	Factor 4
Conservation	.36	.91	.22	.00
Fluency	.91	-.27	.21	.22
Originality	.93	-.24	.[illegible]9	-.22
Imagination	.82	.18	-.55	.01

Two analyses of covariance, with posttest scores as the dependent measure and the pretest scores as the covariate, were then conducted: one on the conservation scores and one on the combined divergent scores. These yielded no significant main effects of treatment for either dependent measure. There were no differences between groups. See Table 14.

Table 14

Analysis of Covariance

Main effects	SS	MS	F	significance
Convergent	5.60	2.80	.304	.739
Divergent	4377.29	2188.65	1.415	.255

The assumption of homogeneity of regression was tested for both the convergent and combined divergent posttests by the procedure suggested by Nie et al. (1981). The test indicated no significant covariate by treatment interactions (F = 0.154 for convergent, F = 0.586 for divergent). The conclusion that there were no significant differences between groups was supported, and it also confirmed the effectiveness of the randomization of subject assignment to treatment groups.

The hypotheses proposed in this experimental part of the study, that computer-assisted learning enhances the general problem-solving skills of kindergarten children, and that Logo enhances the general problem-solving skills of kindergarten children to a greater extent than other forms of computer-assisted learning, cannot be supported on the basis of the results obtained using these instruments and these treatments.

CHAPTER V

DISCUSSION

The Case Study

It is clear that the children were able to work effectively with Logo INSTANT and to learn how to solve some problems specific to their programming. The differences in the behaviour before and after the introduction of structured goal-setting points out the need for some direction for the children if they are to effectively learn to use Logo. The discovery method per se is inefficient for these young learners because of their limited repertoire of skills and experience. What appears to work best here is what Bruner calls "guided discovery" (1966). The children were able to take the tools that were supplied to them, the task cards and the procedure UNDO, and to use these to extend their activities as they were not able to before. They did not need a lot of continuous teacher intervention, merely a little boost applied at an opportune time.

At the beginning it appeared that a lot of the key-pressing behaviour was merely a haphazard psychomotor activity. This was most evident from the number of episodes that resulted in an immediate CLEARSCREEN and the number of redundant commands (Tables 1 and 8). Lack of practice with the commands and unfamiliarity with the keyboard would have been a contributing factor only in the beginning sessions. Many of the simple figures that the children created were single lines of varying lengths, that appeared to be terminated because the child simply did not know what to do next. Then as the children were led to become more attentive to the task and more able to set a goal, there was less haphazard playing with the system. They developed a capacity to recognize a problem, analyze it and then apply a strategy to try to solve it. The increasing number of commands per episode and the decreasing number of episodes per session are further evidence that the children were involved at a deeper cognitive level (Figure 1).

Initially very few children were able to set goals for themselves (Table 3). Of the goals they did set for themselves, many of them were not appropriate for their programming skill level. Children at this developmental level (in transition toward the stage of concrete operations) would have difficulty in setting realistic goals since their perception of the borders between fantasy and reality are indistinct. That they did not improve in their ability to achieve their goals over time (Table 4) is not surprising in light of some characteristics of the problem-solving behaviour of this age group, i.e. that they perceive a solution as all right or all wrong, and that they do not judge the appropriateness of a strategy to solve a problem (Steinberg, 1980a). This also explains why they did not achieve a higher rate of success even when they persisted at length to attempt to solve a problem. Although all of the children were able to solve at least one problem at one time or other, there was a broad range of individual differences with some of them becoming fairly expert and some of them having trouble most of the time.

That the number of identifiable length and angle errors appeared to increase over time (Table 8) is logical within the aforementioned context. (If the programming is haphazard it is not really possible to judge if an error has been committed.) As the children's programming was becoming more task-related there was more possibility of making, and recognizing, an error. The fact that they were not especially proficient at choosing an appropriate strategy to correct a problem would support this conjecture.

The greatest problem the children had in their programming activities was to distinguish right from left (Table 8). This is entirely consistent with their developmental level, and bears out Gregg's (1978) findings about young children's ability to understand the principles of turtle motion. The children in this study were able to understand the differences in the actions produced by the various key-presses, they were able to come to understand that the turn commands caused the turtle to pivot on its axis, but their egocentrism combined with their unclear distinction of the difference between left and right caused them untold confusion. Although the children had been introduced to the concept of "playing turtle" and had enjoyed the idea, they did not make much use of this as a means to help them with their right and left orientation problems. The main reason for this may have had to do with their motivation to be at the computer. They often expressed regret that their computer contact time was so short and so infrequent, and they were loathe to waste one minute of their special time. Also to be considered is an orientation problem of considerable magnitude for children of this age: the turtle functions in an environment that is on a vertical plane with rigid boundaries, i.e. the monitor screen, whereas "playing turtle" involves functioning on the floor, a horizontal plane whose boundaries are vague.

While the children did spontaneously evolve strategies to debug their problems, they were not very efficient at applying them effectively a good part of the time. The idea of reversing commands to correct an error is quite remarkable in the light of their developmental level. Most Kindergarten children have entered a transition phase toward the stage of concrete operations. The principle of conservation is a cornerstone in the foundation of this stage. This principle cannot be assimilated until the concept of reversibility is completely understood (Ginsburg and Opper, 1969). These children were able to understand it and to apply it within this context.

Although some of the children found the UNDO command helpful in debugging right from its introduction, many initially used it as a device to reproduce their figures, and were chagrined when they realized that it was diminishing the figure with each subsequent use. Once they understood this, they were able to apply it appropriately.

The Experimental Study

The strong correlation among the three factors in the creativity test (Table 12), raises once more the question about whether it is possible to measure aspects of creativity, particularly with young children. That young children are creative there is no doubt, but the limitations in their ability to communicate effectively limits our means of assessment. As well, there is controversy around the issue of creativity, and there are those who view it as a dimension of personality rather than of cognition (Willhoft and Lissitz, 1982).

Both tests were chosen for their capacity to reflect very general thinking skills, but they were not specific enough to offer any real information about the children's problem-solving skills in terms of the application of strategies to solve problems. Clements (1985) conducted a study similar to this with grade one children in which a battery of tests measured vocabulary, reflectivity, creativity, metacognition, logical thinking and directionality. The results were also similar to this study in that the differences were not significant overall, although the Logo group did score significantly higher in the creativity test and in the test of metacognition. He suggests that Logo does not affect cognitive development per se, but may affect the way children use the cognitive abilities they possess.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The informal observations of the three treatment groups suggest that there was some very interesting problem-solving behaviour going on in all three groups in very different ways: the incidental activities to add complexity in the CAI group, the spontaneous hypothesizing of the CAL group, and the reversing of commands in the LOGO group. This supports the notion of young children as natural problem solvers (Papert, 1980).

It can be concluded that the hypotheses for the case study were basically supported by the results, while the hypotheses for the experimental study were not supported.

1) Kindergarten children can cope with computers, managing the hardware and working positively with the software. All the children were highly motivated to work with the computer. They all coped very well with the hardware. Indeed, after the initial training they were able to start up the system on their own and to access their particular activities on disk.

2) Kindergarten children will react favourably to certain types of software. The problems with drill-and-practice software (CAI) reflect the difficulties with all types of drill-and-practice activities. Once concepts are mastered, the lack of challenge can have very negative effects in terms of overall learning.

Computer-assisted activities in and of themselves are not necessarily good simply because they involve the computer; they must have relevance for the situation and for the learner.

The structure of the thinking skills software (CAL) allowed the learners to make choices and decisions while being presented with challenge. The children were highly motivated and generally able to achieve success without experiencing frustration.

Although Logo INSTANT is appropriate for use with kindergarten children, because it presents the possibility to manipulate the turtle with single-letter commands, the need for clearer understanding of directionality calls into question the appropriateness of turtle geometry for children at this concrete-operational developmental level. It was a positive experience for the most part for the children, but on the basis of these informal observations, it would appear that CAL materials specifically designed for this age group may be more effective than Logo activities overall.

3) The nature of the children's social intereactions depended partly on the type of software. The behaviour of the children in this study underlines the need for children to work in pairs, or in small groups. It was observed time and again that children working alone at a computer became very helpless during program operation, forgetting even the most familiar commands and losing all sense of purpose.

Then they actively sought support and help from peers and the teacher, even though the initial reaction to being able to work alone at the computer was positive.

Drill-and-practice for the most part elicited negative social interactions, squabbling, boredom and arguing for turns, although some of the attempts to add cognitive complexity to the activity led to some real consultation, sharing and mutual enjoyment. The thinking skills software encouraged and stimulated sharing and cooperation. There was a lot of real consultation and collaboration, and very little interaction that could be construed as negative. The nature of Logo task seemed to create alternating (one child would work at a task while the partner waited to have a turn after) as opposed to sharing as a working style. There seemed to be the most negative interactions in Logo; there was a lot of complaining and arguing.

The observations of the social interactions of the three treatment groups suggest that it would be very worthwhile to observe and categorize the social behaviour of young children working at computers. The possibilities for the development of problem-solving skills in the affective domain as well as the development of group problem-solving behaviour through the computer ought to be examined further.

4) Kindergarten children are able to set elementary programming tasks for themselves while using Logo. The

results on goal-setting clearly indicate that they can do so, but that they may need some direction in order to do so.

5) Kindergarten children will use strategies to solve their programming problems with Logo. The results concerning debugging activities indicate that with encouragement they will attempt to apply a strategy, although they are not always successful.

6) Computer-assisted learning will enhance the general problem-solving skills of kindergarten children, and 7) Logo will enhance their problem-solving skills to a greater extent than other forms of computer-assisted learning. These hypotheses were not supported by the results. Computer activities alone will not lead to the development of problem-solving skills. The author believes that there is a need for direct instruction in the recognition and application of strategies to solve problems, and further that there is a need for explicit instruction in their application elsewhere in order for transfer to occur. This conclusion is consistent with the work of Newell and Simon (1972). This points out a question about the test instruments used for the study. Neither test required the application of a strategy to solve a problem. In both cases, the children were presented with problems to solve, but were given no feedback on the suitability or efficacy of their solutions. There is a need for the development of

suitable instruments that can directly assess the transfer of strategies from one problem area to another.

A review of the current literature by Krasnor and Mitterer (1983) suggests that there is no published evidence of the generalizability of the "powerful ideas" of Logo to date. This is certainly supported by this present study. It must be recognized that the level of programming accomplished in this study would not develop very powerful generalizable skills. The teaching was not explicit enough, for one thing; the children were not taught to recognize and to apply strategies for solving their problems. However, an example of the effect of direct teaching and its consequence could be seen in the learning of a "debugging" attitude after the introduction of the UNDO command. The use of debugging strategies increased overall after the children understood the function of this tool that they had been given (Table 10). The fact that the children significantly decreased the use of an immediate CLEARSCREEN (Table 1) would also suggest that they had developed an adaptive response, that their attitude evolved from one of giving up in the face of failure to one of persisting in the pursuit of a solution to the problem.

Krasnor and Mitterer (1983) suggest that the Logo experience needs to be specifically tailored to facilitate transfer and they cite three central conditions:

1) the selection of appropriate groups of children,

2) the use of techniques to foster the awareness of the general utility of problem-solving heuristics, and

3) the inclusion of the major components of the Logo environment. In view of the lack of transfer of skills in this study, their suggestions, particularly the second one, need to be taken into consideration in any further studies of this nature.

They also emphasize the importance of the social aspects of group problem-solving activities, and suggest that although Logo and other computer experiences may not directly enhance the development of problem-solving skills, the social and motivational aspects make it important to introduce computers to young children.

Very young children definitely enjoy working with personal computers and they can operate them effectively with software appropriate to their developmental level. Certain conclusions can be drawn from the observations of the three groups in this study and certain recommendations for the classroom use of personal computers with Kindergarten children can be made:

1) Software of a drill and practice nature should be used only initially for keyboard familiarization.

2) There is a need for software with structured activities that have objectives that are clear to the children, which require cooperation and which allow for choice and exploration.

3) Logo INSTANT can be used with some success. This is compatible with Boyd's suggestions that pre-packaged drill and practice CAI is of use to novices and that with the right guidance auto-elaborative CAL (such as Logo) may be a powerful educational tool for young children (1983).

If Logo INSTANT is to be used with children of this age, before its introduction as a tool for drawing and as an introduction to turtle geometry, there should be a strong emphasis on activities away from the computer which require cooperation and collaboration, such as playing turtle and manipulating model turtles on a grid. There is a need to provide a kinesthetic component to the computer experience and to integrate computer activities into the whole curriculum. On the basis of the author's teaching experience and from observations with this age group, it is suggested that the INSTANT commands should be introduced through microworlds that gradually introduce the commands and use them to get around the screen in a playful way without the need to "draw" right from the beginning. Only when the children demonstrate a real understanding of turtle motion, particularly the turning, should they then be allowed to explore the possibilities of turtle geometry.

One objective of this study was to find evidence to support the continuing quest of a number of researchers to develop a model for general problem-solving. Tetenbaum and Mulkeen describe this as a project that is "irresistable to the best cognitive scientists despite on-going disconfirming evidence." They call for a moratorium on the attempt to teach generalizable problem-solving skills by teaching Logo in the schools until further research can be conducted. They argue that a moratorium on the widespread teaching of Logo would allow teams of educators to identify and test out the circumstances under which Logo would be a "cognitive amplifier."

Such further research with learners in this age group would require fairly comprehensive effort and long-range study. Longer exposure to thinking skills software and to Logo INSTANT with some direct teaching in such problem-solving strategies as planning ahead, hypothesizing and debugging would probably show differences over time. The author's belief is that the development of a comprehensive integrated curriculum to teach generalizable problem-solving skills should involve not only the computer but activities over a wide variety of environments and in all three domains, cognitive, affective and psychomotor. The computer should be used mainly in ways that are unique to itself and that cannot be effectively and more economically implemented in some other manner in the classroom. Logo INSTANT, Gertrude's Secrets and Facemaker are good examples of software for young children that make use of the remarkable simulation capabilities of the computer. They can make concrete the kinds of activities that the children would otherwise have little experience with, extending the boundaries of the environment that is available for them to explore.

REFERENCES

Almy, M. *Young Children's Thinking.* New York: Teachers College Press, 1966.

Behr, A. L. *Methods and Techniques in Educational and Psychological Resaerch.* Pretoria, South Africa: J. L. van Schaik, 1973.

Bowman, B. T. Do Computers Have a Place in Preschool? Address for the New Mexico Association for the Education of Young Children, Alburquerque, N.M., February 1980. (ERIC Document Reproduction Service No. 231504)

Boyd, G. M. Four ways of providing computer-assisted learning and their probable impacts. *Computers and Education,* 1982, *6*, 3, 305-310.

Brady, E. H., & Hill, S. Young children and microcomputers: research issues and directions. *Young Children,* March 1984, 49-61.

Brown, A. L., & DeLoache, J. S. Skills, plans and self-regulation. In R.S. Seigler (Ed.), *Children's Thinking: What Develops?* Hillsdale, N.J.: Lawrence Erlbaum Associates, 1978.

Bruner, J. S. *Toward a Theory of Instruction.* Cambridge, Ma.: Harvard University Press, 1966.

Case, R. A developmentally based theory and technology of instruction. *Review of Educational Research,* 1978, *48,* 439-463. (a)

Case, R. Intellectual development from birth to adulthood: a neo-Piagetian interpretation. In R.S. Seigler (Ed.), *Children's Thinking: What Develops?* Hillsdale, N.J.: Lawrence Erlbaum Associates, 1978. (b)

Chait, S. An analysis of children's problem-solving in a graphics oriented programming environment. Unpublished Master's Thesis, McGill University, 1978.

Clements, D. H. Logo programming: can it change how children think? *Electronic Learning,* January 1985, 28-75.

Conlin, C. A computer in an infant school. *CAL News,* 1981, *16,* 1-3.

Copple, C., Sigel, I. E., & Saunders, R. Educating the Young Thinker: Strategies for Cognitive Growth. New York: D. Van Nostrand, 1979.

De Bono, E. Children Solve Problems. New York: Harper & Row, 1972.

De Bono, E. Teaching Thinking. Great Britain: Penguin Books, 1976.

De Vries, R., & Kohlberg, L. The Concept Assessment Kit - Conservation. Journal of Educational Measurement, Winter 1969, 6, 4, 263-266.

Flavell, J. H. Metacognitive aspects of problem solving. In L. B. Resnick (Ed.), The Nature of Intelligence. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.

Ginsburg, H., & Opper, S. Piaget's Theory of Intellectual Development. Englewood Cliffs, N.J.: Prentice-Hall, 1969.

Gnanadesikan, R. Methods for Statistical Data Analysis of Multivariate Observations. New York: Wiley, 1977.

Goldschmid, M. L., & Bentler, P. M. Concept Assessment Kit - Conservation. San Diego, California: Educational and Industrial Testing Service, 1968.

Golick, M. Look, Listen, Discover! Montreal: International Cinemedia, 1976.

Greenberg, S. The development and revision of a Logo checklist appropriate for use with ten and eleven year old children. Unpublished Master's Thesis, Concordia University, 1984.

Gregg, L. W. Spatial concepts, spatial names and the development of exocentric representations. In R. S. Seigler (Ed.), Children's Thinking: What Develops? Hillsdale, N.J.: Lawrence Erlbaum Associates, 1978.

Grimes, L. Computers are for kids: designing software programs to avoid problems of learning. Teaching Exceptional Children, 1981, 9, 49-53.

Guilford, J. P. The Nature of Human Intelligence. New York: McGraw-Hill, 1967.

Hines, S. N. Computer programming abilities of five-year-old children. Educational Computer, July-August 1983, 10-12.

Johnson, M. C. A Review of Research Methods in Education. Chicago: Rand McNally College Publishing, 1977.

Klahr, D. Goal formation, planning and learning by preschool problem-solvers. In R. S. Seigler (Ed.), Children's Thinking: What Develops? Hillsdale, N.J.: Lawrence Erlbaum Associates, 1978.

Krasnor, L. R., & Mitterer, J. O. Logo and the development of general problem-solving skills. Unpublished manuscript, Brock University, 1983.

Lauzon, A.-R. An investigation of the effects of using large-letter keyboard and large coloured letter displays on the computer-assisted learning of vocabulary by mentally handicapped children. Unpublished Master's Thesis, Concordia University, 1982.

Lawler, R. In the lap of the machine. Boston Review, June 1982, pp. 8-10.

Moran, J. D., et al. Measuring creativity in preschool children. Virginia Polytechnic Institute and State University, May 1982. (ERIC Document Reproduction Service No. 224584.)

Munro-Mavrias, S, Computer programming by kindergarten children using LOGO. Paper presented at AMTEC/ADATE Confluence '83, Montreal, June 1983.

Nelson, H. Learning with LOGO. onComputing, Summer 1981, 14-16.

Newell, A., & Simon, H. A. Human Problem Solving. Englewood Cliffs, N.J.: Prentice-Hall, 1972.

Nie, N. H., & Hull, C. H. Statistical Package for the Social Sciences, Update 7-9. New York: McGraw-Hill, 1981.

Papert, S. Mindstorms: Children, Computers and Powerful Ideas. New York: Basic Books, 1980.

Piaget, J. Judgment and Reasoning in the child. London: Routledge & Kegan Paul, 1928.

Piestrup, A. Preschool children use an Apple II microcomputer to test reading skills programs. Portola Valley, Ca.: Apple Education Foundation Field Test Report, 1981. (ERIC Document Reproduction Service No. 202476)

Polya, G. How to Solve It. New York: Anchor Books, 1957.

Reisman, F. K., Floyd, B., & Torrance, E. P. Performance on Torrance's Thinking Creatively in Action and Movement as a predictor of cognitive development of young children. Creative Child and Adult Quarterly, 1981, 6, 4, 205-209.

Resnick, L. B., & Glaser, R. Problem solving and intelligence. In L. B. Resnick (Ed.), The Nature of Intelligence. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.

Seigler, R. S. The origins of scientific reasoning. In R. S. Seigler (Ed.), Children's Thinking: What Develops? Hillsdale, N.J.: Lawrence Erlbaum Associates, 1978.

Smith, F. Understanding Reading: a Psycholinguistic Analysis of Reading and Learning to Read. New York: Holt, Rinehart & Winston, 1978.

Smock, C. D. Goldschmid-Bentler Concept Assessment Kit - Conservation. Professional Psychology, Fall 1970, 491-493.

Solomon, C. J. Leading a child to a computer culture. Anaheim, Ca.: SIGCSE-SIGCUE Symposium, February 1976. (ERIC Document Reproduction Service No. 207575)

Solomon, C. J., & Papert, S. A case study of a young child doing Turtle Graphics in Logo. Massachusetts Institute of Technology: Artificial Intelligence Memo No. 375, 1976. (ERIC Document Reproduction Service No. 207578)

Spivack, G., & Shure, M. B. Social Adjustment of Young Children: a Cognitive Approach to Solving Real-life Problems. San Francisco: Jossey-Bass, 1974.

Steffin, S. A. Using the micro as weapon: fighting against convergent thinking. Childhood Education, March/April 1983, 255-258.

Steinberg, E. Evaluation processes in young children's problem solving. Contemporary Educational Psychology, 1980, 5, 276-281. (a)

Steinberg, E. Using PLATO to teach problem solving. University of Illinois: Computer-Based Education Research Laboratory, 1980. (ERIC Document Reproduction Service No. 202455) (b)

Swigger, K. M. More programs for kids. *Educational Computer Magazine,* May/June 1983, 24-25.

Swigger, K., & Campbell, J. The computer goes to nursery school. *Educational Computer Magazine,* July/August 1981, 10-12.

Torrance, E. P. *Thinking Creatively in Action and Movement.* Bensenville, Il: Scholastic Testing Service, 1981.

Travers, R. M. W. *Essentials of Learning,* (2nd ed.) New York: Macmillan, 1977.

Willhoft, J. L., & Lissitz, R. W. A methodological study of the Torrance Tests of Creativity: can creativity be faked? Annual Meeting of the National Council on Measurement in Education, New York, March 1982. (ERIC Document Reproduction Service No. 220458.)

Wise, J. E., Nordberg, R. B., & Reitz, D. J. *Methods of Research in Education. Boston: D. C. Heath, 1967.*

Zeiser, E. L., & Hoffman, S. Computers: tools for thinking. *Childhood Education,* March/April 1983, 251-254.

Ziajka, A. Microcomputers in early childhood education? A first look. *Young Children,* July 1983, 61-67.

APPENDIX A

Checklist of Logo Programming Behaviours

CHECKLIST OF PROGRAMMING BEHAVIOURS

Session ________
Group ________

CATEGORY	EPISODE 1	2	3	4	5	6	7
1. Nature of Goal							
non-existent							
vague, changeable							
fixed, self-chosen							
fixed, teacher-suggested							
related to previous episode							
unrelated to previous episode							
2. Programming Style							
haphazard/exploratory							
contingent							
planned - at computer							
3. Types of Errors							
length							
angle							
right/left confusion							
redundant commands							
4. Debugging Action							
CLEARSCREEN/restart							
reverse commands							
360 rotation							
UNDO							
5. Result							
immediate CLEARSCREEN							
simple							
complex							
achieved goal							
number of commands							

DEFINITIONS

1. Nature of Goal: This had to be deduced from the protocols, since the subjects were not asked if they had a goal. Output judged to have a non-existent goal had a general appearance of scribbling or doodling, with no identifiable pattern. Output judged to have a vague, changeable goal was that in which a pattern could be perceived at some point: sometimes initially, in which case it became obscured by haphazard programming, or sometimes later, in which case the preceding haphazard programming seems to have suggested a goal. If a pair of students announced an intended goal, it was noted and entered on the protocol later. A fixed and self-chosen goal was generally revealed by a clear and recognizable design or pattern. Fixed teacher-suggested goals were those from the introduced task cards. A goal was judged to be related to the previous one if the previous goal was not completed and if the subsequent one resembled it strongly, particularly in the beginning.

2. Programming Style: A haphazard/exploratory style is one in which the child typed in any command which happened to come to mind. A contingent style was being used when the child's choice of command depended on what had already been created on the screen. A planned style was one in which the child indicated ahead of time what needed to be done to accomplish the goal. The latter two could not be easily inferred from the protocols, but related episodes presented clues and relevant observed behaviour was noted on the protocol.

3. Types of Errors: Errors were only counted as such if they were not corrected or debugged. Length errors were those in which the estimation of the length of a line was wrong. Angle errors were those in which the estimation of the size of an angle was wrong. Right/left confusion referred to a difficulty in deciding between turning right or left. Redundant commands were those which did not further the programming activity, such as typing a turning command through more than 360 degrees.

4. Debugging Action: CLEARSCREEN/restart was fairly self-explanatory and involved starting all over again from the beginning. Reverse commands occurred with both length and angle errors, requiring the child to do the exact opposite of what had just been programmed: in the case of angles it simply involved turning left as much as just having turned right and vice versa, and with length of line it involved turning around 180 degrees, retracing the path the right amount and then turning around 180 degrees once

more, a fairly complex manoeuvre. 360 rotation involved doing a complete rotation in order to correct too large an angle. UNDO was a procedure in which the output was erased from the screen and then redrawn with the last step deleted.

5. <u>Result:</u> Immediate CLEARSCREEN occurred when a child typed C (CLEARSCREEN) at the beginning of an episode. A simple figure generally consisted of one or possibly two lines. A complex figure was one in which there were at least three distinct lines and two distinct angles. Number of commands referred to the number of commands typed during the episode.