DOCUMENT RESUME

ED 249 932 IR 011 360

AUTHOR	Mawby, Ronald; And Others
TITLE	Structured Interviews on Children's Conceptions of Computers. Technical Report No. 19.
INSTITUTION	Bank Street Coll. of Education, New York, NY. Center for Children and Technology.
PUB DATE	Feb 84
NOTE	41p.
PUB TYPE	Reports - Research/Technical (143)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	Cognitive Development; *Concept Formation; Elementary Education; *Microcomputers; *Models; *Programing; Programing Languages; *Student Attitudes; Vocabulary
IDENTIFIERS	*LOGO Programing Language

ABSTRACT

The terms and concepts children used to explain their beliefs about computers before and after classroom exposure to microcomputers were studied to identify misconceptions about computers that could interfere with computer-based learning. Children in each of two classrooms at the Bank Street School for Children were interviewed individually on their conceptions of computers at the beginning and end of their first year with computers in the classroom (fall 1981-summer 1982), which included work with the LOGO programming language. Their experience with computers inside and outside of school were also examined to provide a context for the following overall issues: (1) children's understanding of programming computers in LOGO; (2) their broader conceptions of computers beyond their function of programmable devices, such as cultural objects which have general properties and various functions; and (3) their interpretations of person/computer systems. By the end of the year, most of the children had some rudimentary skill with the computer; however, the progress of even the most advanced students could be hindered by their inadequate mental models of LOGO. The biggest problem seemed to be the lack of two concepts: a program as an algorithm applied to data, and the central processing unit as the functional core of the computer. Eight references are listed. (LMM)

CENTER FOR CHILDREN AND TECHNOLOGY

Structured Interviews on Children's Conceptions of Computers

Ronald Mawby, Catherine A. Clement,
Roy D. Pea, and Jan Hawkins

Technical Report No. 19

Bank Street College of Education 610 West 112th Street NY, NY 10025

Structured Interviews on Children's Conceptions of Computers

Ronald Mawby, Catherine A. Clement,
Roy D. Pea, and Jan Hawkins

Technical Report No. 19

February 1984

CENTER FOR CHILDREN AND TECHNOLOGY
Bank Street College of Education
610 West 112th Street
New York, NY 10025

STRUCTURED INTERVIEWS ON CHILDREN'S CONCEPTIONS OF COMPUTERS

Ronald Mawby, Catherine A. Clement,
Roy D. Pea, and Jan Hawkins

Introduction

Knowledge about computers, beginning in the elementary years, has come to be viewed as a fundamental aspect of education in our nation. Uses of computers in schools are widely heralded as promising individualized instruction in the humanities as well as the sciences, and as powerful tools for children to use for learning and problem solving. Computers have been characterized as technological innovations that will change thinking, communication, and the processes of education in ways at least as profound as the advent of the technologies of writing (Ong. 1982) and printing (Eisenstein, 1979). Just as the widespread availability of texts came to democratize the process of education and made learning available beyond the academies, so computers may enable educators to motivate interest in learning, and to adapt instruction to the specific needs and misconceptions of individual learners in ways never before deemed practically possible.

With all this excitement and the rapid entry of computers into the nation's schools come great concerns and profound questions. Some of the most frequently raised questions are: How can computers be used effectively in schools? For what purposes should they be used? What new concepts and skills will children need in order to use computers effectively? Are there developmental obstacles to certain kinds of computer-supported learning for children of different ages?

With so many questions of immediate practical import pressing on researchers and educators, we are in danger of overlooking a crucial fact. Children, as well as educators, must come to terms with computers in the classroom. How do children view the computers that are being introduced into the classroom? What is their conception of this new machine? How do they think about it? What is it for? Do they expect to like it? What do they think they will need to know in order to use it?

When computers enter the classroom, children face a novel experience. For researchers and educators, children's construal of this experience is crucial. How children interpret the computer will be a

major determinant in both their interaction with it and the subsequent effect it has on them. In addition, children's initial intuitive conceptions of computer functioning will provide the starting point for their further learning. We know that effective teaching addresses itself to the students' current level of knowledge and experience. Thus, in order to develop and teach curriculum concerning computational concepts, we need to know the outlines of children's conceptions of computers. If child novice models of computer functioning are badly flawed, the models will impede rather than support their learning about and with computers; that is, children may acquire low-level skills, but the deeper conceptual understanding that allows skills to develop and generalize may elude them.

Because of the importance of the child's construal of computers, we sought to document, through exploratory structured interviews, the level of children's computer understanding. As part of a major project on the social and cognitive influence of computers in school, funded by the Spencer Foundation, we have been engaged in longitudinal studies (fall 1981-summer 1983) of classroom computer use by children from eight to twelve years of age. There were six microcomputers (three Apple II and three Texas Instrument computers, all 64K) in each of two classrooms at the Bank Street School for Children (8- and 9-year-olds; 11- and 12-year-olds). With 25 students and one teacher for each class, these classrooms are rich computer-resource environments by today's standards; on the average, schools have only one microcomputer per 185 children, and most schools have only one per classroom (Center for Social Organization of Schools, 1983). The children work with Logo, a programming language designed to allow learning about such powerful computational ideas as variables, procedurality, recursion, and list structures, yet which is very accessible to children (Harvey, 1982; Papert, 1980).

We interviewed the children individually on their conceptions of computers at the beginning and end of their first year with computers in the classroom (fall 1981-summer 1982). Our goal was to discover the terms and concepts children used to explain their beliefs about computers before and after their classroom exposure. We hoped thereby to identify misconceptions about computers that could interfere with computer use and learning, and that could be addressed by instruction.

We view the findings as suggestive, revealing provocative trends, but with the limitations inherent in any qualitative study. We have not done systematic pre-post comparisons for individual children (or across ages) to track specific shifts of understanding about computers during the year of computer use. Subsequent studies by others with much larger populations and standardization techniques will be neces-

sary to establish the validity of our observations. Our more limited aim is to give a qualitative account of children's computer conceptions, and to impress on educators and researchers that this information is crucial if computers are to be effectively integrated into schools.

We were surprised by the children's awareness of the uses of computers throughout society. We were impressed by the children's critical comparisons of human thinking and computer operations, a topic in which they showed great interest. Certain remarks made by the children alerted us to important misunderstandings that needed correction, such as confusions about the concept of "program," which indicated some naivete on their part about the powers of the computer.

Finally, we view the insights children give us into their thoughts about computers--their functions and limits--as of some historical interest. In the not-too-distant future, computer use will be so pervasive in our society that the idea of a computer-naive child will seem antiquated, no more understandable than a school-aged child who does not know about books. These discussions thus shed some light on children's knowledge about computers at a critical watershed, before these powerful symbolic devices enter nearly every aspect of life, including school.

In the discussion below, we first describe the children's experience with computers in and outside of school. This provides a context for addressing three overall issues. First, we wanted to probe children's understanding of the particular type of computer activity they were engaged in throughout the year--programming computers in Logo. Since computers can be used for many different functions (e.g., word processing, data management, simulations), it was necessary to understand children's mental models of the computer in terms of their own mode of engagement with the technology. Second, we analyze children's broader conceptions of computers, beyond their function of programmable devices, such as cultural objects which have general properties and various functions. Third, we examine children's interpretations of person/computer systems. We were interested in children's understanding of people and computers as possible "partners" in problem solving, and in their perspectives about the relationship between human thinking processes and the computational processes of the machines.

Analyses of these three issues offer a broad picture of children's conceptions of computers, both before and after programming experience over the course of a school year.

Computer Experience

Experience with Computers Before Interview 1

Important background information for the interviews on children's conceptions of computers is the extent of children's computer experiences both before and during the research period. Had they themselves used computers, and where in their lifespace had they seen computers?

The Logo computers (Apple II Plus and Texas Instruments) were first introduced to the Bank Street School for Children for six weeks in the spring of 1981. During that time, several of the 20 students interviewed for this study had access to them in their classroom. Nearly all the children we interviewed had done something with the computer during that 6-week period, though usually only during a few lunch periods. Our knowledge of the classroom use of computers during that pilot period is through teachers' recollections since no records were kept at that point. For the classroom year between Interviews 1 and 2, discussed below, there is a record of the time each child spent on the computer.

Extent of Experience

Aside from their hands-on classroom experience during the previous spring's pilot period, the 20 children came to their first full year of classroom computer work in the fall of 1981 with large differences in exposure to computers outside of school.

Three of the younger children, but none of the older children, had what they described as computers at home--a calculator, a Magnavox Odyssey II game computer, and an Atari game computer. Two of these children had used a computer somewhere other than at home or school: one had seen and used an Atari computer at his friend's house; another had used a TRS-80 computer in a store. We use the proviso "what they described as computers" advisedly, since children's definitions of computers varied a great deal, and they had no doubt widely used calculators and arcade videogames.

Four of the older children had used "computers" outside of school. One played computer games on a friend's computer. Another used a Pet Commodore at a computer school to learn BASIC, and also used computers at a science museum. A third had used a friend's calculator. A fourth child played computer games with friends, and owned a battery-operated computer game and a digital watch.

Apart from these users, only a few children of each age mentioned that someone they knew closely used a computer, usually at work. The "someone" was usually a parent (4 cases), friend (2 cases), or close relative (1 case).

In addition to classroom experience, children reported exposure to computers through media, such as television or magazines, and in places such as libraries, offices, grocery stores, banks, police stations, and hospitals. Every child gave at least one example of a computer, from Apples to electronic games to calculators. Only one child insisted she had no computer experience or knowledge.

Computer Experience Between Interviews 1 and 2

Logo classroom context. The children gained computer experience in their classrooms at the Bank Street School for Children, a private school on Manhattan's upper west side. One classroom included 25 (11 boys and 14 girls) 8- and 9-year-old children (third and fourth graders), while the other consisted of 25 (10 boys and 15 girls) 11- and 12-year-old children (fifth and sixth graders). The children represented a variety of ethnic and socioeconomic groups and a range of achievement levels. Many of the children were above national norms in school achievement and came from upper-middle-class and professional homes.

Each classroom had six microcomputers during the 1981-1982 school year. Both the younger and older groups had three Apple II Plus computers and three Texas Instruments (TI) 99/4 computers. In each class children were learning Logo, a programming language designed to be easily accessible to children and to encourage the development of thinking skills (Papert, 1980; Byte, 1982). For the Apple computers, a widely distributed MIT Artificial Intelligence Laboratory version of Logo software was used; for the TI computers a commercially available TI version of Logo was used. Teachers received extensive training in Logo classroom use before the school year, and the computer programming activities during the year were intended by the teachers to be largely child-initiated so as to encourage the Piagetian discovery-learning pedagogy advocated for Logo by Papert. While teachers gave the children some simple instruction in Logo during the first several weeks and occasionally held group sessions to introduce new aspects of Logo during the year, their self-defined role was principally one of responding to questions and problems from students as they came up. Students were encouraged to create and develop their own computer programming projects. The children's computer activities included drawing pictures in turtle graphics, playing with the TI Logo sprites, and writing and playing game programs.

Teachers scheduled computer use for students in their classrooms so that everyone would have equal time--about two 45-minute work periods per week. There were additional optional times for computer use throughout the day--before school and during lunch period--when computers were available on a first-come, first-served basis.

Children varied tremendously in the amount of class time they spent working with computers. For the 26 children whose records were kept over the year, the mean number of hours spent on the computer was 26, the standard deviation was 14.5, and the range was 3 to 55 hours. Of the 20 children interviewed at the beginning and end of the 1981-1982 school year, nine spent 20 hours or less at the computers (two spent as few as 3 hours), and eleven spent 20 hours or more (six with 40 hours or more).

Computer experience outside school between Interviews 1 and 2. Between the fall of 1981 and the spring of 1982, the 20 children had some additional computer experience outside of school. Only one child acquired a computer at home. One child played computer games in computer stores. A second used a friend's Atari computer to learn BASIC programming, and another friend's Apple II to make a maze game and to program music. A third child's father sometimes brought a computer home from work on weekends so that they could play computer games together. Four other children had new opportunities to use computers--primarily to play videogames--at the homes of friends or relatives. One of these children started using multiplication/division software on his aunt's Atari computer.

Qualitative Aspects of Computer Experience

In Interview 1, if the children had used a computer before, we asked whether or not they liked it and why. Younger children generally answered yes, they liked it because they had fun moving graphic shapes around or typing. Computer use was definitely identified as fun and not typical of school. Older children often described computers as fun, especially because they could draw with the turtle graphics, but they sometimes had negative associations. One girl said: "It's just like work, since you have to remember what to tell it." Another boy noted that "it can get boring because it's too much of one thing."

In Interview 1 we also asked the children whether some kids would like the computer more than others, who would or wouldn't like it, and why. The younger children mentioned novelty and familiarity as two reasons why kids would like the computer. With respect to novelty, they thought that people who hadn't used the computer before would be the ones who would like it. The kids said things

like: "A computer isn't something kids have every day so they'll love it": or "Many new people who haven't seen the computer would like it more." The children thought that familiarity with computers would lead to knowledge about them. They believed that some expertise or the ability to gain expertise is necessary for liking the computer. One child told of a classmate who hadn't liked the computer, but "when he learned how to do programs, he liked it better." One said that "older people will like them 'cause they know more and could understand more easily," and that "kids who know how to use them will like them more. Others who know less will still have fun but won't be able to do as much with the computer. Some will find them boring but I'm gonna like them because I've used them before."

For the younger children, then, novelty makes computers initially attractive. Their remarks suggest, however, that sustained interest requires that the first spark of interest be fanned by good pedagogy. Competence becomes the source of continual novelty and interest.

Although a few of the older children mentioned novelty and know-how derived from familiarity, they focused on the match or mismatch between computer activities and an individual's general interests. They related computer activities to similar noncomputer activities and predicted that children's interest in that class of activities would manifest itself in their like or dislike of the computer. For example, some kids would like the computer because they like typing, or drawing shapes, or playing computer games, or figuring things out, or just because they are more "interested." Others predicted that some kids wouldn't like the computer because they like to answer questions, think for themselves, or play outside ("I'd rather play baseball").

By Interview 2, after a year of computer exposure in school, children had much more elaborate accounts of their computer experiences accompanied by reasons for their affective responses.

All of the younger children said they liked working with computers; for example:

> I like making games, and other stuff on them.
>
> I like to fool around with the computer and experiment with making different programs. We learned exciting and fun things, like how to make different games and harder, more complicated programs.

Many of the older children said they enjoyed working with computers. Two recurrent themes were that computers were fun and that the

children were learning new things. One important factor in making it fun was their control over the activity--they were doing their own games, their own projects, things that interested them. Some exemplary quotes are:

> It's fun and adventurous and you're learning more. You figure out stuff you never knew before.
>
> I like making games. You can do things you're interested in.
>
> Interesting and fun. I made a doctor program that would ask you questions and give you advice.
>
> It was very, very fun. I learned a lot from computers. They showed me things I never knew you could do on a computer, like doing PRINT to make conversations. I loved learning all the new things about drawing and making sprites.
>
> They're fun because you can do whatever you want, make your own games on it.
>
> I liked them. It's made me understand something new.

Negative reactions to computer experiences came mainly from girls. The themes of not learning, and not being as much fun as real life are noteworthy:

> I got tired of the things I know and I don't learn anything.
>
> I don't like it so much. I don't know why I didn't. It's not something I enjoyed. If I had free time I wouldn't work at the computers, but I'd read or do homework.
>
> I'm not crazy about computers, but it was a new experience. They're neat, but I don't think they should take over my work. It's much more fun to do in real life than in computers.

Summary

Exposure, experience, and interest in computers varied widely in the children we observed. Most of them were from middle- or upper-middle-class families and thus, relative to the school-aged population at large, had frequent opportunities to interact with computers.

Children liked the novelty of the computer and seemed especially enthused by controlling the computer on projects of their own choice.

Children's Knowledge of Computer Programming Concepts

What is a "Program"?

Computers are devices that can be used for many purposes. Novices can experience computers through different kinds of activities. The particular domain of exposure may influence the organization of understanding about the machines. As we have seen, most of the children we interviewed had considerable exposure to computers, especially in learning to program in Logo. We wondered if that exposure was sufficient for them to acquire some fundamental computer concepts. One such core concept is "program." A program may be thought of as a list of instructions for the computer to execute or as an algorithm applied to a data structure. Understanding the concept of program is at the heart of computer literacy. Teachers need to know if children really understand this concept after a simple, general introduction to computers and a few dozen hours of Logo programming experience.

In Logo, screen drawings can be made directly from the command mode without using programs. Logo programs are reusable modules of code which may be used both to record individual procedures and as part of larger programs. In Logo, programs are written in the edit mode and then run by being called from the command mode.

The children gave some interesting answers to our question: "What is a program?" Before he worked with the computer, one 8-year-old believed that "to program meant that I would reach in and fiddle with putting wires together." Some answers were very good, such as:

> A list of instructions to the computer.
>
> It's something you make, TO P, then you're making a program called P. When you type P and then "enter," the computer does whatever you told it to in P.
>
> It's a thing you write things in and save it.

Above all, many children defined a computer program as a "time saver," as a way of easily entering a list of commands each time you want to use them again:

> It's when you have to tell the computer something if you want to be able to do it over and over again.

It saves time typing it over each time. You can use it over.

It makes things quicker to do.

It is striking that only one of the 20 children noted the powerful modular property of Logo programs--that they can serve as building blocks for increasingly complex programs. One older child said: "You can make a program inside a program inside a program, and so on." For most children, it was clear that a program was viewed as a local device for saving keyboard entry time, not as a way of building a storehouse of tools, utilities, or useful programs for incorporation into future programs.

Of most significance for conceptual diagnosis was the children's overly general definition of programs. A quarter of the children defined a program as anything entered into the computer at the keyboard, making no distintion between the command mode when commands may be entered and executed but not saved, and the edit mode when commands may be entered and saved as programs. Examples of this overly general definition are:

A program is whatever information you put in.

You have to make a program to get stuff to come on the screen.

The computer won't understand unless you tell it what to do in a program.

The prevalence of these remarks leads us to make several pedagogical points. First, if children are to use the modular procedure powers of Logo to make complex programs, it may be well to place greater emphasis on the "building block" and "toolbox" metaphors for these procedures. Second, educators should be aware of and work to overcome children's tendencies to lump together the command and editing modes. All the information one enters at the keyboard is <u>not</u> a program. If children believe that anything entered at the keyboard is a program, they will be disappointed when their commands are not stored and reusable.

<u>Computer Process Understanding</u>

In Interview 2, we asked a series of Logo comprehension questions designed to tap children's knowledge of Logo and internal computer operations. We felt that knowing how much Logo the children understood would help us to interpret their responses to other questions,

and perhaps expose conceptual flaws that might interfere with their learning.

In our view, continually expanding mastery of computers requires both know-how (practical knowledge or skill to achieve particular goals) and knowing-that (propositional or declarative knowledge about the structure of the domain). For many Logo concepts, skilled know-how rests on a base of propositional knowledge: effective action is mediated by a mental model of Logo. However, even with very limited understanding one can make any number of interesting screen effects. We asked the children several questions in order to determine their grasp of Logo. We asked (for different cases) what screen effects would occur after typing commands at the keyboard, and what would happen inside the computer. We also asked questions about information storage and retrieval processes: how "save" and "read" operations are carried out, and what happens inside the computer when these things are done. The precise location of the information inside the computers was an issue that concerned many of the children.

Screen Effects of Keyboard Entry/What Happens Inside

Our most extensive queries for four different examples were what screen effects would occur after a series of commands was typed at the keyboard. These examples were selected to be representative of simple commands and sequences encountered by the children in the course of their Logo experience. We then asked what would happen inside the computer in each case to create the screen effects. Answers to the "what happens" questions were comparable across examples and will be presented together. The children's predictions for screen effects of the four examples are presented in turn.

1. Red ball. In TI Logo, one can call one or more "sprites," or forms, from memory, and assign them different shapes by telling a specific sprite number to "carry" a variable shape name (e.g., truck, ball, car, rocket). A variable color is assigned through the command "setcolor." The sprite can also be assigned a speed and a heading.

We presented the command sequence:

Tell 1 carry :ball setcolor :red home

This example tells sprite number one to carry the shape "ball," to set its color to "red," and to position itself at "home" (the center of the screen).

All but three of the 20 children correctly predicted that the screen would show a motionless red ball at the center of the screen. Two of these three children, both from the younger class, assumed that the red ball would be moving (i.e., would have a speed and heading) even though it had not been commanded to do so--probably because the children usually created moving sprites. One child did not know what would happen on the screen.

Children's explanations of what happens inside the computer when the commands are entered were of two basic types. About half the children of each age said they didn't know. Most of the remaining half of the groups described processes involving "memory" (younger group) or "instruction understanding" (older group). Younger children mainly alluded to memory in their accounts:

> What you type goes in memory and memory tells it put this ball on the screen.
>
> Memory understands it.
>
> The light comes on inside and it goes to memory.
>
> It goes onto your disk.

Other mysterious processes were described, such as:

> It goes through the computer brain.
>
> Like a watch they have to move around; each part moves a different part.

Older children focused mainly on understanding instructions:

> The computer understands the instructions.
>
> Logo lets the computer know what to do with the commands.
>
> Computer does the equivalent of thinking about it and then shows what the commands mean.

2. Box drawing. In both Apple and TI Logo, children could direct the graphics turtle to leave traces, in effect drawing lines on the computer screen. They could issue these commands directly in command mode or write programs in edit mode to be run later. For this example, we presented the command sequence:

```
REPEAT 4 [FORWARD 40 RIGHT 90]
```

With this command, the graphics turtle begins at the center of the screen, draws a square box 40 units per side by moving forward 40 units while tracing its path, turning right 90 degrees, and repeating this sequence of actions three more times.

Most children correctly predicted the effect described above, and gave explanations of the inner workings of the computer to create these effects that were basically restatements of those quoted above.

3. Makeshape. In TI Logo, children often edited shapes that were stored in memory and could be assigned to the various sprites, in effect, allowing children to create their own sprite shapes. To get a shape to work on, one types MAKESHAPE # (# from 0-25; some numbers have assigned shapes, others are blank) and a shape appears on a 16 x 16 grid of "tiles." By moving the cursor over the grid with the SHIFT key pressed, one blacks out the tiles to create the new shape.

Nearly all the children correctly predicted that MAKESHAPE 8 would recall a shape that could be edited once it appeared on the screen. Again, few gave informative accounts of internal processes, although a number of children correctly noted that "the shape [you are working on] will be stored in memory until you turn off the computer; it will carry your shape. You can save it on the disk."

4. Variable tail recursive program. Unlike the previous three cases where command sequences were hypothetically entered into the computer in command mode, here we presented a program. This program requires relatively sophisticated understanding of the use of variables and recursion in Logo.

```
TO J :LINE
FORWARD :LINE RIGHT 120
FORWARD :LINE RIGHT 120
FORWARD :LINE RIGHT 120
J :LINE + 10
END
```

In this program, ":LINE" represents a variable called LINE which must be specified when the program is run. For example, one would type "J 10," and the program would be called and run with all occurrences of the variable LINE equal to 10. A triangle is thus produced by constructing three lines at 120 degree angles to each other. The penultimate line (J :LINE + 10) exemplifies what is called "tail recursion." There the program J creates a copy of program J, which executes beginning with its first line, this time with a value for LINE of 20 (original value of line plus 10). When this second version of

program J reaches its penultimate line, it will create a third copy of program J, this time with a value for LINE of 30, and so on. Each time J runs from beginning to end, it produces a triangle and creates another copy of J. The visual design produced would be a triangle with sides of 10, followed by a larger triangle with sides of 20 around it, and so on, indefinitely (one stops the program execution with a specific combination of keys).

What did the children predict would happen? Most children did not know what the program would do. Some children knew that a design would be drawn, but not what it would be. One younger child and three older children correctly noted that triangles of increasing size would be drawn. There were two interesting responses from younger children. One knew that the shape drawn would increase in size and keep growing, but thought the shape was a square instead of a triangle. Another child made an error common in other studies we have done with children in these classrooms. This child thought that the shape would be drawn 120 times, as if it were "REPEAT 120"--numbers are not always correctly assigned to their category of meaning in the program. As for the internal process mediating the screen effects, only one child noted that "when the computer sees 'LINE,' it will put in the number."

Understanding of "Save"

Almost all the children could explain how to save a program, which involves putting the diskette on which one wants to save the program into the disk drive, and typing SAVE and the name of the file. Most children could not explain the internal process by which this was accomplished. Half the older children did not know, and the other half observed only that the computer puts the program on the disk. Most younger children did not know, but two noted the role of memory:

> The computer has to memorize it. We have to memorize the name, but the computer puts it in its brain.
>
> The program goes into memory on the disk.

Understanding of "Recall"

Almost all the children could explain how to recall a program from a disk. This involves putting the diskette from which one wants to recall the program into the disk drive, typing RECALL, pressing the space bar until the name of the file to be recalled is found, and then pressing return to load the program into working memory. Most children could not explain the internal process by which this was

accomplished. Half the older children did not know, and the other half noted only that the computer takes the program and puts it "in the memory bank" or "into the computer." Most younger children did not know, but two also mentioned memory: "The computer has to recall it; it's already in its brain"

One younger child was the only student who appeared to understand that when a program is recalled, a copy of it remains on the disk: "The disk has the program on it. It's giving it to the computer with RECALL, but still keeping it."

Understanding "where the information is" by forming an adequate functional model of the filing system is important in guiding operations with peripheral devices. Our limited results suggest that this information is not spontaneously induced by children who are just beginning to work with computers.

Summary

These children varied greatly in their knowledge of Logo, perhaps as a function of time spent on the computer. Many children knew what to do to make certain things happen: to get sprites to flash across the screen, to save programs and then recall them from the disk, to draw simple geometric figures. But most of the children seemed to have poor models of how Logo works. As we will see, despite their ability to use the computer for certain purposes, the children's lack of certain basic organizing conceptions limited their understanding of computers (e.g., lack of an adequate mental model of Logo).

Unpacking Children's Concepts of Computers

Children's Descriptive Typology of Computers

In addition to discovering children's mental models of programming in Logo, we wanted to probe their organization of information about computers in general. Novice understanding of computers will probably be drawn by analogy from familiar devices, and may be inadequate in essential respects. If we want to improve their concepts and sharpen the analogies (and disanalogies) between computers and other devices, we need to uncover children's terms of interpretation.

General computer literacy will incorporate certain basic features: the distinction between hardware and software; the recognition of information capacity as a relevant dimension along which computers differ; types of software available for different categories of use. Do children know the difference between hardware and software, or is "the computer" taken as a globally fused whole? Do children classify

computers in terms of a sophisticated functional system or in superficial perceptual terms?

In this section, we discuss responses concerning the differences among computational devices; the processes by which computers in general work; and the functions of computers. In Interview 1, we asked the children what computers are, whether there are different kinds of computers, and whether computers could be different even if they looked alike. The last question probes the distinction between hardware and software. Two computers with identical hardware but different software are functionally different even though they are perceptually alike.

In response to the first question, children listed an extended class of objects as computers. One listed "Apples, digital watches, arcade games, electronic games, tape recorders, typewriters, telephones, and stoplights." Calculators were included by two children, and a ditto machine was mentioned by another.

All the children described computers in physical terms, such as keyboard and TV screen. A few children also referred to how the computer works--"It's a box you put programs in." In general, older children gave more detailed and sophisticated answers. The responses of younger and older children are presented below.

Younger children. Several children classified computers in terms of physical size and visible parts, such as the presence or absence of a printer, or the number of buttons. One child said: "There are big computers with typewriters and screens, medium ones that look like tape recorders, and little ones like calculators."

Others classified computers by use or the places where they are used. Most of these children made the global statement that different kinds of computers do different things. One child specified "computers for games, computers in schools used for teaching, and computers in business and science used to store information."

A few children pointed out that computers could be programmed to do different things; another said that different computers could have different brains. It was not clear whether children distinguished programs from computers. One child did mention a difference in capacity: "Some computers store more information than others."

Few children gave specific responses to whether computers could be different even if they looked alike. Several children answered with a simple "yes" or "I don't know." One said that different computers which looked alike could be programmed to do different things, such

as math or history. She did not say whether one computer could be programmed in two different ways. Another child said that two computers could be made by the same company, look alike, and be used in different places: "A computer at business could look like that Apple over there and they could be made by the same company." One child said that if computers were different, they probably would not look exactly alike.

Older children. When asked if there were different kinds of computers, several older children responded by naming different brands. Others mentioned perceptual attributes like physical size and arrangement of buttons (claiming, however, that all computers "have to have the same way of getting information in...have to have buttons to type").

Several children said that computers could be programmed for different purposes, such as "to figure out math, or another one would be programmed to figure out social studies and could not do math." Another said computers could be used for "math, the stock market, scientific problems, and world matters like planning and peace treaties." Users of computers included "hospitals, for manufacturing, in school offices and arcades."

A few children noted that computers can vary in capacity. One said that there are "large complicated computers, smaller ones which are easier and convenient." Others mentioned "expensive sophisticated computers" or "computers with many connections with other computers."

In response to the question--"Could computers be different and look alike"--one child said that looks didn't matter, computers could have different programs. Others said that computers could look alike and have different data in them, or "one could make noise while another makes words or draws." One said that whether two computers were different depended on what was inside. Another child said: "Two Apples could have totally different minds--one could be dumb, another stupid, depending on how they were programmed."

Summary. Here, again, there is a wide range of knowledge among the children. At the low end, some children speak of "big and little" computers, and the arrangement of buttons on the keyboard. In the middle range, we hear of different computer functions, though it is not clear whether these children distinguish hardware from software. At the upper end, some children clearly distinguish "computers" (hardware) from "programs" (software) and know that information capacity is a relevant dimension. Older children had a more differen-

tiated notion of the computer, one less tied to perceptual features and user actions.

How Computers Work

We probed further into the children's conceptions of computer parts and processes. The essence of a computer is that it computes; that is, it manipulates symbols according to rules that preserve meaning. This abstract concept of a "semantic engine" seems far removed from the external, perceptual features of a computer. While a keyboard may be likened to a typewriter, a monitor to a TV, a disk drive to a cassette player or phonograph, the central processing unit is more like a brain. Like the brain it is not normally visible, and is best known to us by what it does. Since it is out of sight, will children even mention it as a part of the computer? Or does their conception of a computer not involve computation?

In order to address these questions, in interviews 1 and 2 we asked children: How do computers work? What are the different parts of the computer? What does each of the parts do?

Younger children. A few children barely responded to the Interview 1 question about how computers work. One child said it "runs by electricity." Fuller answers focused on the user's actions, the external computer parts, or the computer's internal workings:

> It works by plugging it in. You press buttons, put things in and take them out of slots.
>
> You put a program in.
>
> There are lots of wires and batteries.
>
> There are engines inside and a computer brain. The brain lets the computer know what to say like our brain lets us know what to say.
>
> There is an electronic brain inside and you have to teach it.
>
> There is a little piece like a mind, and it reads information typed out on the screen and does what it's told, or takes information off the disk and does what it's told.

Two children referred to the computer's memory bank "which remembers like our mind." One of these children gave a detailed account of computer operation in terms of switches and a binary computational

system. Several of the children who mentioned the computer's memory or brain pointed out that people put information into the computer brain, usually through disks or cartridges.

In Interview 2, we asked three questions: What are the parts of the computer? Do you know how the inside of the computer works? How do the different parts work?

When asked to name the parts, more than half of the younger children mentioned only the external visible parts: the TV screen, the keyboard, and usually the disk drive. Students who did mention the inside of the computer said there was a brain or a memory; one said the brain was a "regular piece of metal." One child noted that "the memory is the computer. It would just be a TV or typewriter without it."

When asked if they knew how the inside of the computer works, only three children said anything other than "I don't know." One said; "It works by the wires inside." The other two children referred to TI Logo sprites, saying they work "by lights flashing across the screen." One of these two said the sprites were controlled by the memory which "does everything, and it works by being programmed when the computer is made."

When asked to describe how the different parts of the computer work, a few children focused only on the external parts. Two children accurately described the disk drive, screen, and keyboard, but appeared not to recognize that anything else is necessary for computer functioning:

> The disk drive saves and recalls. You type what you want on the keyboard, and this shows on the screen...the disk drive and keyboard work into the screen.
>
> The disk puts information in, you type information in on the keyboard, and the screen shows your typing.

These children seemed to think of the computer as a typewriter where work is shown on a TV screen and saved on a disk instead of on paper. They did not mention any process intervening between what they typed and what appeared on the screen.

Other children referred to the internal parts of the computer. One revealed a minimal understanding: the user "presses keys and there is a memory for saving things." Others described the functions of parts and mentioned the memory:

> You get information in with the typewriter, the TV shows you, and the disk drive saves and recalls...the memory works the whole computer.

> One part has to think, one part has to make it move, I think it's the brain (though I'm not sure if there is a brain). The parts move together like cooperating.

One child pointed out that when the machine is turned off, everything in the memory is erased except what was "already in there." This child was one of the few who mentioned any permanent memory. In general, the younger children appeared to have little awareness of the internal parts which operate on the information fed into the computer.

Older children. In response to Interview 1 questions, older children described users' actions, the interactions among the parts of the computer, and the computer's internal workings. Most of the older children understood in more detail than the younger ones how computers work, although one said only that they work through electricity, and another only knew that various computers have different languages.

Two children described the flow of activity between the user and the external parts of the computer:

> I think you push buttons and information comes out on the screen and you write it down.

> I give information to the cartridges, you type something in, it goes onto the cartridges and onto the screen.

The last child seemed to think that the user interacts with the cartridge rather than with the computer. Another child referred to the computer memory:

> You type in something, push buttons, and [it goes through] a circuit and prints, and through the memory banks and comes out on the screen.

The older children, like the younger ones, described the inside of the computer with varying detail and sophistication:

> There is a whole bunch of stuff and little things inside.

> The machinery inside makes them move.

[There is a] memory, open it up and put something in, it would be in memory and you could draw...like LT 90.

> You start programming. It's like another mind in the classroom or house...it's like having another mind not really a person. Different computers have different minds; a watch can only do a few things.

Two children had a more sophisticated albeit partial understanding:

> Computers have memory banks to store information, and they can be programmed to give information when keys are pushed.

> They have memories called RAM [random access memory] and if you tell it something you have to tell it in a certain language like BASIC 'cause it doesn't understand the whole English vocabulary. Computers work by giving it commands and telling it to remember something (which you would do by pushing "print" or something) and they would put it in their memory.

A few students mentioned that people program the computer. One said: "First you have to teach it stuff. Inside, the computer has its own brain. You teach it stuff and then it would have two brains. People teach it in addition to what the computer already knows."

In Interview 2, most of the older children, in contrast to the younger, did not describe computers simply in terms of the observable parts with which users interact. When asked to name parts, most mentioned the inside of the computer as well as the keyboard, screen, and disk drive. Some children merely acknowledged that something was there--"I don't know what the things inside are called"--while others mentioned "memory banks," and two children mentioned "chips."

Again, the question asking how the inside of the computer works elicited mostly "I don't know" responses. One child said that the language is loaded into memory chips called ROM [read only memory], and another said that part of the computer is programmed to understand the language and everything else the user programs.

For the most part, older children were aware that something intervenes between the input and output. Descriptions of the process varied. A few children gave vague responses, such as:

> The keyboard prints into the computer; the computer knows everything that's on it.

> The screen shows what you are saying to the computer and what the computer is saying to you.
>
> Information is recalled from the disk into the computer.

A few mentioned the memory banks or chips which store input. Most children recognized that information saved and recalled from disks is not permanently in the computer:

> You make your program on the computer inside. If you turn off the computer you would lose it.
>
> Computer gets information off the disk, and without the disk drive the memory bank wouldn't know what to do.

As described above, one child was aware of the distinction among the hardware, the firmware (language chip), and the users' program.

Summary. In general, the inner workings of computers are largely unknown to these children, thus substantiating our observations of children's understanding of Logo. Younger children tend to have only vague ideas about the existence of a "black box" that they call "memory." The older children know the "black box" exists, and some have a sense of its functional architecture. It would appear that the children know nothing of the computer as an "abstract computational device."

Functions of Computers

General computer literacy implies a principled understanding of the powers and limits of computers. Knowledge of the nature, power, and limits of computers is essential for their judicial use. It is of paramount importance for people in an "information age" to be wise consumers of information technologies.

We approached the issue of computer power and limits by asking children about the functions of computers. In both interviews 1 and 2, we put the questions: What are computers used for? What do they do?

In general, the children gave three types of answers: computers are used to store and manage information; they are used to solve problems; and they are used for fun. Because children know that computers are used to "figure things out," we asked them what kinds of problems computers can solve. Since there were age trends, we will discuss the older and younger groups separately.

Younger children. Responses to interviews 1 and 2 were similar. The children gave a number of global answers to the use question: "computers are used to help people"; "to help you learn"; and "to answer questions." In addition to the helpful functions of computers, children said they were used for fun, as in playing games and drawing. After the year of classroom exposure, children tended to name more instances of computer use. The only new types of use mentioned were related to classroom activities, such as writing, printing, and drawing shapes in Logo graphics.

The prototypical computer problems for nearly all the children were mathematics and, to a lesser extent, spelling. One child mentioned computers in scientific discovery--"helping solve problems that could not be solved before." One child said computers could help solve crimes. Another said that computers could do whatever they were programmed for. Computer problems ranged from math to banking to helping people stop drinking alcohol. Computers were said to be unable to help people stop fighting, get dressed, or heal cuts.

Older children. Older children gave a greater number of, and more specific, responses. They seemed to have a sense of computers as powerful tools: they are fast, can manage large quantities of information, and can solve important problems. Like the younger students, the older children mentioned general categories of computer use such as helping people, helping people to learn, answering questions, and solving problems. As with the younger children, they noted the use of computers for fun, games, drawing and, writing. The role of computers in different professions was often mentioned: computers are used for "controlling things" such as space flights and air traffic; they help secretaries, answer telephones, and are used for "simulating what would happen with nuclear waste." One child said that computers could help you figure out how to rule the world.

The older children emphasized that computers can store, manage, and make available a large base of information:

> Humans have so much to do. Computers can give you information, store it, and communicate with other computers or memory banks to exchange information.
>
> They can save people piles and piles of paper.
>
> Computers can keep track of things.
>
> Keep track of gasoline prices--calculate the average of these.

Keep business records.

Store recipes and phone numbers.

Keep track of stock.

Send mail.

Older children tended to say that computers could solve problems only if "you know how to program it," "give it answers," or "give it information." Most children had the important insight that any programmable problem could be solved. The specific types of problems most often cited were mathematics, spelling, and science. One child thought computers should only be used for fun, "to solve boredom." By Interview 2, some children seemed disillusioned. One child said computers could solve problems only if you already knew the answer. Another didn't think that computers helped since "they only say back what you tell them."

The range of computer problems varied. One child said computers could stop child abuse; another said computers couldn't predict future events like plane crashes and deaths. While one child denied that computers could solve emotional and family problems, another said a computer could act just like a psychiatrist if programmed to do so. Beyond the limitation that computers must be programmed, computers "can't solve fighting problems" or "eat or make a sandwich."

Summary. The older children had a more adequate sense of the powers of computers. They were aware, especially by the year's end, that computer usefulness depends upon programming. However, because of confusions about the nature of programs children tended either to overestimate or underestimate computer power. This will be discussed further below.

How Computers Help People to Think and Solve Problems

Finally, we were interested in children's understanding of computers as parts of systems that can help people to solve problems. As noted above, many children focused on the computer as problem solver. In order to illuminate their understanding of this, we asked the children how the computer helps people solve problems.

Younger children. Although younger children knew that computers were used in problem solving, many did not know how this was done. In Interview 1, the question drew responses such as "you push buttons" and "the answers come on the screen." One child said the computer solved problems with its brain, if it had a brain. Another

was sure the computer had a brain and memory banks, and thought about whatever you asked it to think about.

Similarly, in Interview 2, many said they did not know or that you just "type it in." A few made vague mention of programs and memory. One child said you can tell computers how to solve problems using a language they understand.

Older children. As compared with the younger group, older children were more apt to say that computers solved problems with "answers in memory." Some said that people teach computers to solve problems. One child had a sophisticated model in which the computer manipulates different kinds of information, and then outputs it to people who relate it to the problem to be solved. Some older children, like the younger ones, said that they didn't know or "you press keys."

Interview 2 was less informative: some children didn't know; one said computers store information in memory using a language; one mentioned disks; another said people "program in" solutions. One child said computers help solve problems by remembering for you.

From both younger and older children, then, we heard a range of responses from "you push buttons" to "you program the computer in a language it understands." The difference in level of description is striking; the second reveals a far richer conceptualization. By Interview 2, most of the older children had grasped the importance of programming for problem solving.

Since problem solving is one kind of thinking, we probed further by asking: "Can computers help people think?"

Younger children. For the most part, the younger children said, yes, computers can help people think because the computer can give answers to questions:

> Yes, you can ask them questions and they can show you the answer.
>
> Yes, in math might be able to help a person, if you would program something.
>
> If you have a problem but can't figure it out, pull a switch down and it'll tell you.
>
> Yes, like Blackjack--if you had 15 you would have to think if you should hit another or just stop.

Others denied that providing information helped people to think:

> No it can't help people think; it can only tell them something.
>
> Sometimes it solves problems for you, in numbers, but you wouldn't be thinking.

Older children. The older children responded much like the younger ones:

> Yes, it can give you information or ideas to do problems.
>
> If you put in pieces of information and then ask a question the computer can give an answer.
>
> Yes, solving problems if they are already solved, like math problems.
>
> Computers give information. It'll trigger a person's memory to make him remember similar problems he did before. The computer might solve half the problem, the human the other half.

Several mentioned the limits of the aid computers can give:

> It depends on what people want to think about.
>
> They can't help people think whether they like a person or how they feel about themselves.

This theme of different aspects of thinking is one we will return to.

What Do Computers Know?

In Interview 2, we probed further into children's conceptions of the relation between human problem solving and computer processes by asking three questions: What kinds of things do computers know? How do they know these things? Do computers know some things that people don't tell them?

Younger children. When asked what kinds of things computers know, younger children mentioned "certain kinds of words," "programs," "how to make sprites," "games," "math," and "they know what you tell them."

As for how the computer knows these things, the children responded:

From stuff inside them.

When they're in the computer shop men computerize them.

They just do--like I'm a person, how do I know? I just do.

When we asked--"Do computers know some things that people don't tell them"--the younger children's responses were split between positive and negative:

Yeah, I think so, like MS ["make shape," a primitive command in TI Logo].

Yes. If I didn't know math, they would know it. If I didn't know a program, they would know it.

* * * * *

No, never. A man or lady always has to tell them.

No, because computers don't know anything unless you program them to know it.

Older children. When asked what kinds of things computers know, older children mentioned "sprites," "what to do with certain words," "commands," "a lot of things if you tell them," "how to make programs you teach them," "whatever you teach them," and "only what they are programmed to know."

When asked how the computers know these things, the children said:

It just knows it. I'm not sure how it works.

I don't know how they know. It's been programmed in.

It's something to do with the way they function.

When the older children were asked if computers know things that people don't tell them, they agreed that computers know the commands that are already in them when the machine is turned on:

When you turn it on, it knows things that are already there.

They know FORWARD and BACK [turtle graphics primitive commands].

If you type out something and make a mistake, they'll know what they're doing even though you made a mistake.

But most children said that the ultimate source of the computer's knowledge is the people who do the programming:

They got knowledge from people who made them.

Computers have to be programmed.

They have to be told everything.

User Knowledge Required

In order to probe their understanding of the human factor via-a-vis the operation of computers, we asked children what a person must know to use a computer. Their answers were directly related to their knowledge of the machine: a child who answers "push buttons" has an impoverished conception. Children recognize that things must be known in order to use the computer. Their views of what these things are constitute their initial ideas of expertise and, by implication, what they need to know to use the computer--their personal curriculum. Answers given on Interview 2, after the year of classroom exposure, tell us what children felt were the important things they had learned and what they must learn more about.

Younger children. In Interview 1, most of the younger children listed what one must do to operate the computer: plug in and turn on the machine, and "press buttons" to make it work. Others answered vaguely, "You have to know how to use it." A few recognized the importance of programming in a language the computer understands:

You have to know how to talk to it.

If you go to a machine and start pressing buttons, it would just say, "I don't know what you're talking about."

You can't just write "computer put a box on the screen"--you have to do it differently.

A couple of children mentioned thinking, math, and reading as necessary skills.

In Interview 2, many more of the younger children talked about programming and knowing a language--"typing what you want it to do." Many children mentioned the importance of knowing what not to

do, such as not pushing buttons which would cause the program to crash. Some continued to respond at the level of "you press buttons" or "you have to know how to use it"--answers that reflect either superficial or diffuse understanding. Occasionally children mentioned typing, reading, and spelling skills as important. One child said nothing special was needed: "Anyone can use it--smart, dumb, rich, poor."

Older children. The older group gave more diverse responses on Interview 1 than the younger group. Many said only: "You have to know how to use it." Some mentioned "pushing buttons"; others mentioned programming and computer languages. Several children talked about access and hands-on experience: "You need to get one"; or "To really learn it you can't just read the book." A few talked about mastery of the system's components, such as saving and reading from the disk drive and using the printer. The general cognitive skills mentioned included reading, writing, spelling, and knowing "the alphabet, numbers, angles, and right from left." Two children spoke about attitudes of patience and "nonviolence" toward the computer.

By the time of Interview 2, nearly all the older children mentioned "making programs" or "knowing commands" or "knowing the language." Some still said "press buttons"; some remarked on knowing how to type. Like the younger group, many stressed knowing what buttons not to push and knowing not to touch the hole in the disk. Several children mentioned the importance of a tutor or teacher to learn from, and one child said you need books to learn from.

Summary. After a year of exposure the children had much clearer and more specific ideas about what one needs to know to use a computer. The transition from "plugging it in" to "knowing the language" is a significant advance. The importance of spelling (the computer will give error messages if commands are misspelled), arithmetic (turtle geometry demands it), and reading (the computer doesn't talk, so you must be able to read) were noted by children and suggest that a desire to use the computer may motivate children to learn these subjects.

Summary of How Computers Help People to Think and Solve Problems

We pause here to gather our results. We asked children what kinds of problems computers can solve, how they help people to solve them, whether computers can help people to think, what things computers know, and what users must know to operate the computer.

The pattern of responses shows that children benefited from the year of classroom exposure. Their comments were less global and vague on Interview 2 than at the start of the year. However, even at year's end many seemed to lack clear concepts of "program." One fruitful conception, understood by very few children, characterizes a program as an algorithm applied to a data structure. Of course, we would not expect children to know the term "algorithm" without teaching, but the concept could have been drawn from their work with Logo. Lacking this concept, the children tended to misjudge the power and limits of computers. Too often, they spoke as if computers know specific facts, such as the product of 23 times 45, rather than having general algorithms that generate specific answers to specific questions.

Many of the younger children seemed almost to view the computer as a natural object, which "just knows" things and has the intrinsic ability to answer questions. These children overestimated computer power because they did not understand it to be conditional on programming.

Most of the older children, especially after their year of experience, remarked that computer power is conditional on programming. Some tended to be disappointed because computers "only say back what you tell them." A portion of these children underestimated computer power. They did not grasp the fact that an individual user need not write all his own programs and, more importantly, did not distinguish the algorithm from the class of problems the algorithm can solve. Some thought that because a person must program in the algorithm, there was nothing left for the computer to do. They failed to see that computers can apply algorithms to data in ways far beyond the capacity of humans. Many other children thought that a new program must be written for each specific problem in a class of problems. This error supported, and was supported by, their undeveloped programming practices--they tended not to write modular Logo procedures that could be flexibly reused in several programs, or used to solve many specific problems in one general class of problems.

Children's Comparisons of Human Thinking and Computer Operations

Since its invention, the computer has been identified as a new kind of tool because, instead of increasing our physical power, it augments our intelligence. The early terms "thinking machine" and "electronic brain," like the current "artificial intelligence," express the powerful idea that the computer is created in the image of the human thinker. We probed children's conceptions of human thinking and computer operations by asking three questions: Can computers think? What does it mean to think? Are computers like brains?

Children's understanding of the relation between thinking and computer operations will affect their views on the powers and limits of computers. Their conceptions of the similarities and differences between people and computers will influence their interactions with the computer. Also, since proponents of Logo say that it is a language with which one may talk about thinking (Papert, 1980; Nickerson, 1982), we should see what aspects of thinking children believe can be embodied in computer operation. Finally, as the analogy between the human brain/mind and the computer becomes part of our general culture, we wished to know if children see any disanalogies between human thinking and computer operation.

Can computers think? Many children (like many philosophers; see discussions in Dennett, 1978) are not sure whether or not computers think. Most of the remarks quoted here were prefaced by indicators of uncertainty: "maybe"; "I'm not positive"; "I think so"; "I don't know." Affirmative and negative answers were given with nearly equal frequency: "They [computers] think for you"; and "A person really has to think for them."

Computers seemed to provide children with a "black swan" phenomenon--factors in nature which are usually found together are now empirically separated. This forces a distinction between essential and nonessential features of the concept of thinking. Thus, computers in the classroom may give children a natural setting in which to develop a more explicit and articulate conception of thinking.

Younger children. There were few differences between the responses on interviews 1 and 2 among the younger children. Many of them denied that computers think because computers are neither alive nor human:

> They're not alive, don't have a brain, just wires and things to make them work.
>
> I think you need a human brain to think.
>
> They are machines, they don't have brains.
>
> They are not people, they don't have brains.
>
> Not really [think]. A man or a woman would make the computer's brain; it's not so strong.
>
> Computers are not flesh and blood.

Others, however, considered mechanical thinking a variant type of thought:

> Yes, they think in the way that machines think. They connect wires to make it think.
>
> They don't really think. Well, I guess they do, with little gears and they look back, look through files--"that's what she wants so she gets it."
>
> They think, but not like we do.
>
> Yeah, they know how to think real good.

Often children contrasted the self-initiated, spontaneous thought of humans to externally elicited computer operations:

> They can talk only if programmed to talk. They can't talk whatever they want to talk.
>
> They can't do anything without a person making them do it.
>
> Yes [they think], but not like human brains. Humans can talk, have the information in their heads, they don't get it from somebody else. Computers wait for us to type in something.
>
> They can sort of [think]. They take stuff from what somebody programmed and they sort of pass it on.
>
> If you type something in, it will think about it. If the computer has information in its memory bank, it will think about it and give you the answer.

Several children argued that computers must think because of what they are able to do:

> I think yes [computers think], because how would they do everything if they didn't think?
>
> How could the ball [a ROM graphic shape in TI Logo] get in color if the computer didn't think?
>
> They're like humans, they can talk to us.

Older children. When asked if computers think, older children also showed little change from Interview 1 to 2. Like the younger children, several of the older group denied that computers think:

> No, they don't have a brain.
>
> No, brains can think.
>
> No, they don't have a mind.
>
> No, not really, they're just machines that people made. A computer couldn't make another computer.

Many older children noted that computer operation is conditional on programming:

> Not the computer itself [which thinks]; you need to program it in.
>
> They don't have brains. It looks like thinking but it's not because they're programmed by a thinking person.
>
> They only think as much as you tell them to. Thinking is knowledge in the brain--you could put your knowledge in the computer.
>
> Not really [do computers think], only it will know a thing if you program it into it.

Several children asserted that computers do think and are like brains:

> Yes, they think when you make them talk.
>
> They think what they're supposed to write on the screen ...computers think in more detail than people.
>
> Yeah, you can put stuff in them. You can teach somebody how to draw, and tell the computer how to draw.
>
> Yeah, they have a brain because in Logo there is certain stuff they already know.
>
> Yes [computers think], because if a person who runs them knows how to run it, they can make it think.

One child was especially articulate about the computer's inability to understand the meanings of the symbols it manipulates:

> It can remember and it can figure out, but it can't really think about something. Like, if you asked, "Why do you think I'm sad," it wouldn't understand the question. It wouldn't know how to think about it. If you printed in, "What does sad mean," it could tell you; but if somebody asked me what sad means and I picked up a dictionary, it wouldn't mean I was thinking.

In interview 1, we probed further by asking: "What does it mean to think?"

Younger children. Several younger children said thinking means "using your brain." They may have intended this phrase as a physiological account, but it seems more likely that it was used as an idiom for reflective activity, as in "[thinking means] to use your brain, not just say anything."

Several children linked thinking with problem solving:

> [Thinking means] to work on a problem.
>
> Your eyes see things and your brain takes it all in. The brain sometimes stores it and sometimes forgets. When something similar comes up, it takes the information and answers the question.
>
> Say you have a math problem and don't know it, you don't just write down, "I don't know." You use your brain to think...you would study it.

Some children remarked on the awareness that accompanies thought:

> You think what you're doing. When you pick up a pencil, you know that you're picking up a pencil.
>
> [Thinking] is using your mind. If you're reading, you're thinking about the words or if you're writing, you're thinking about what to write. Right now, I'm thinking about the questions.

One child used this aspect of thought to deny that computers think:

> They [computers] take in all the information, but they would never stop there to think. It could never say, "What am I doing here," because it's a machine. The computer wouldn't know what it was doing.

Several children again noted the self-initiated property of human speech as distinct from the evoked responses of computers:

> You can either give the answer or hold it inside.

> We can say anything inside without buttons--they have to have something done to them.

Older children. Responses of the older children to the probe "What does it mean to think" were much like those of the younger group. Several of the older children mentioned the awareness that should accompany speech, and the human choice to express thought in speech:

> Think means to have something in mind you might say or keep to yourself.

> [Think means] to understand what you're thinking. When you say something, you have to understand and know what you're thinking.

Many spoke of thinking as entailing mental independence and going beyond the information given:

> To be able to figure something out without needing to have all the information.

> You try to use your own mind; try to figure out by your own knowledge of the subject what the answer would be.

> You get an idea and you keep on thinking about it...it's like an idea that can be changed into many ideas.

> To get an idea in your head and be able to build on it.

Several children listed types of thinking:

> Thinking is remembering or wondering about something.

> [There are] different ways to think, like imagine, learn, dream even. Dreaming is picturing something in sleep;

learning is when you concentrate on one specific thing and put it into the memory bank.

Some children in the older group described the similarities between human thought and computer operation using computer terminology:

It's like you have your own computer in your head.

Your brain is a memory bank, like a computer; it doesn't have a screen.

Memory bank is like the nervous system in the brain and all the keys are like the nervous system. When you hit a key, it sends messages to the memory bank in the brain.

Other children contrasted computer functioning with human thought in order to define thinking:

It [the computer] can't solve problems unless someone already solved it.

[Thinking means] think about a concept, its different sides. There could be a question that doesn't have a right or wrong answer...computers can't turn something over and look at it.

Summary. These responses provide some insights into the children's concepts of computer functioning and human thinking. When children reflected on the similarities and differences between computers and people with respect to thinking, four themes emerged. Each theme suggests an aspect of the children's concept of human thought that is challenged by the computer. Human thinking is an activity of the brain; it is self-initiated and controlled; it implies self-awareness and self-reflection; and activities like solving novel problems, using language, and answering questions require thinking. For certain children, some or all of these features are essential. Since only some of the features apply to computers, the children are unsure about whether or not computers think.

Conclusion

By the end of the year, most of the children had some rudimentary skill with the computer. However, the progress of even the most advanced students could be hindered by their inadequate mental models of Logo. The biggest problem seemed to be the lack of two concepts: a program as an algorithm applied to data, and the central processing unit as the functional core of the computer.

Thinking of a program as an algorithm applied to a data structure is especially useful for problem solving. The algorithm captures, in a definite sequence of steps, the essence of the solution to a problem. Differentiating the algorithm from the data entails distinguishing (1) the defining character of a class of problems, (2) the range of specific problems falling within the class of problems, and (3) the general form of the solution to every problem in the class. This articulation of the problem-solving situation clarifies problems and fosters intelligent use of previously formed solutions. If children fully understood programs as algorithms plus data structures, they would grasp these problem-solving concepts in explicit and general form.

While Logo is a language in which powerful problem-solving strategies can be articulated and practiced, it is unclear how children may begin to attain these important insights. If children had a thorough understanding of Logo, they might exploit Logo's modular structure as a support in problem solving. Reciprocally, if children generally employed explicit high-level strategies such as problem decomposition, they might discover in Logo a powerful problem-solving environment. But the children we studied had neither deep knowledge of Logo nor explicit problem-solving strategies. Since interesting screen effects can be generated from simple Logo programs, free exploration of the computer does not tend to move children to explore the powerful problem-solving ideas embodied in Logo.

The computational core of the computer is not visible, and children have few good analogues with which to grasp it. The keyboard, screen, and disk drive are more salient and familiar parts of the computer, and children initially focus on the perceptual and user-action features of these components. The conceptual essence of the computer as a rule-governed, symbol-manipulating device largely escapes children. While they understand that the computer "has information" or "answers" inside, they seem to have no idea of how the computer transforms that information in the course of its operations. Thus, children lack any solid idea of the computer as a machine which computes.

The consequences of children's inadequate concepts of program and central processor are manifold. Some children fundamentally misconceive the problem-solving power and limits of the computer. Some treat the computer as a display typewriter with disk save which cannot help people solve problems because the user must type in all the answers. With a better grasp of program and processor, children would have a better idea of how computers can be used in problem solving.

The children had a healthy grasp of the issues involved in the question: Can computers think? Even with undeveloped models of computer operation, they were reflective and insightful about human thinking and its contrast with computer operation. Computers in the classroom may further children's awareness of their own thinking by providing a contrast case which challenges their categories.

Children need adequate computational concepts in order to be competent users of computers and competent judges of their power and limits. Without computer skill, children cannot employ these powerful tools for their own purposes. Without knowledge of the power of computers, children may not be motivated to learn about them. Without knowledge of the limits of computers, children may become mere technocrats, ignorant of the proper place of computers in human life.

References

Byte. Special issue on Logo, August 1982.

Center for Social Organization of Schools, Johns Hopkins University. School Uses of Microcomputers, 1983, Issues 1, 2, 3.

Dennett, D. Brainstorms. Montgomery, VT: Bradford Books, 1978.

Eisenstein, E. The printing press as an agent of change: Communication and cultural transformation in early-modern Europe (2 vols.). New York: Cambridge University Press, 1979.

Harvey, B. Why Logo? Byte, August 1982, 163-193.

Nickerson, R. S. Computer programming as a vehicle for teaching thinking skills. Thinking, The Journal of Philosophy for Children, 1982, 4, 42-48.

Ong, W. J. Orality and literacy: The technologizing of the word. New York: Metheun, 1982.

Papert, S. Mindstorms. New York: Basic Books, 1980.