THE LOGO LINEAGE

by Wallace Feurzeig

Wallace Feurzeig is a division scientist in information sciences for Bolt Beranek and Newman. His favorite implementation of Logo is still Logo-S for minicomputers.

Logo was created in 1966 at Bolt Beranek and Newman, a Cambridge research firm. Its intellectual roots are in artificial intelligence, mathematical logic and developmental psychology. The first four years of Logo research, development and teaching work was done at BBN.

During the early 1960s, BBN had become a major center of computer science research and innovative applications. I joined the firm in 1962 to work with its newly available facilities in the Artificial Intelligence Department, one of the earliest AI organizations. My colleagues were actively engaged in some of the pioneering AI work in computer pattern recognition, natural language understanding, theorem proving, LISP language development and robot problem solving. Much of this work was done in collaboration with distinguished researchers at M.I.T. such as Marvin Minsky and John McCarthy, who were regular BBN consultants during the early 1960s. Other groups at BBN were doing original work in cognitive science, instructional research and man-computer communication. Some of the first work on knowledge representation (semantic networks), question-answering, interactive graphics and computer-aided instruction was actively underway. J. C. R. Licklider was the spiritual as well as the scientific leader of much of this work, championing the cause of on-line interaction during an era when almost all computation was being done via batch processing.

Time for Interaction

My initial focus was on expanding the intellectual capabilities of existing teaching systems. This led to the first "intelligent" computer-assisted instruction (CAI) system, MENTOR, which employed production rules to support problem-solving interactions in medical diagnosis and other decision-making domains. In 1965 I organized the BBN Educational Technology Department to further the development of computer methods for improving learning and teaching, and the focus of our work then shifted to the investigation of programming languages as educational environments. This shift was partly due to two recent technological advances: the invention of computer time sharing and the development of the first high-level "conversational" programming language.

The idea of sharing a computer's cycles among several autonomous users, working on-line simultaneously, had stirred the imagination of programmers in Cambridge in 1963 and 1964. BBN and M.I.T. teams raced to be first in realizing this concept, with BBN winning by days and holding the first successful demonstration of computer time-sharing in 1964. Our initial system, designed by Sheldon Boilen, supported five simultaneous users on a DEC PDP-1, all sharing a single CRT screen for output. Seeing dynamic displays from several distinct programs, simultaneously and asynchronously ("out of time and tune"), was a breathtaking experience.

Time sharing made feasible the economic use of remote distributed terminals and opened up the possibilities of interactive computer use in schools. We had recently implemented TELCOMP, one of the new breed of high-level interactive programming languages. TELCOMP was a dialect of JOSS, the first "conversational" (i.e., interpretive) language, developed in 1962-63 by Cliff Shaw of the Rand Corporation; its syntax was similar to that of BASIC, which had not yet appeared. Like BASIC, TELCOMP was a FORTRAN-derived language originally designed for numerical computational applications. Shortly after TELCOMP was created, we decided to introduce it to children as a tool for teaching mathematics and in 1965-66, under U.S. Office of Education support, explored its use as an auxiliary resource in eight elementary and secondary schools served by the BBN time-sharing system. Students were introduced to TELCOMP and then worked on standard arithmetic, algebra, and trigonometry problems by writing TELCOMP programs. The project strongly confirmed our expectation that the use of interactive computation with a high-level interpretive language would be highly motivating to students.

My collaborators in this research were Daniel Bobrow, Richard Grant and Cynthia Solomon from BBN and consultant Seymour Papert, who had recently arrived at M.I.T. from Jean Piaget's Institute in Geneva. The idea of a programming language expressly designed for children arose directly from this project. We realized that most existing languages were designed for doing computation and that they generally lacked facilities for nonnumeric symbolic manipulation. Current languages were inappropriate for education in other respects as well: they often employed extensive type declarations that got in the way of students' expressive impetus; they had serious deficiencies in control structures; their programs lacked procedural constructs; most had no facilities for dynamic definition and execution; few had well-developed and articulate debugging, diagnostic and editing facilities, so essential for educational applications.

Educational Expression

The need for a new language designed for and dedicated to education was evident. The basic requirements for the language were:

1) Third-graders should be able to use it for simple tasks with very little preparation.

2) Its structure should embody mathematically important concepts with minimal interference from programming conventions.

3) It should permit the expression of mathematically rich nonnumerical as well as numerical algorithms. Examples of nonnumerical problems include translating English into pig Latin; making and breaking "secret codes"; word games such as testing whether a word is symmetric (a palindrome), finding words within words, writing words backwards; question-answering and guessing games (such as "twenty questions" and "buzz"). Our strategy was to introduce mathematical

ideas through experience with these familiar and meaningful problems and projects.

Incredibly, the best model for the new language (which was to be as simple as possible) turned out to be LISP, the lingua franca of artificial intelligence, often regarded (by non-LISP users) as one of the most difficult and formidable of languages. Of course, the syntax of Logo is much more familiar and accessible than that of LISP Essentially, though, Logo is LISP and is thus both an easy and a powerful language. The power is not evident in most existing microcomputer implementations, mainly because of their small memory and restricted performance.

The initial design of Logo came about through extensive discussions in 1966 between Seymour Papert, Dan Bobrow and myself. Papert developed the overall functional specifications for the new language, and Bobrow made extensive contributions to the design and did the first implementation. Subsequently, Richard Grant made substantial additions and modifications to the design and implementation, assisted by Cynthia Solomon, Frank Frazier and Paul Wexelblat. I gave Logo its name.

This first version of Logo was pilot-tested in the summer of 1967 with fifth- and sixth-grade math students at the Hanscom Field School in Lincoln, Massachusetts, under support of the U.S. Office of Naval Research. In 1967-68 we designed a new and greatly extended version, which was implemented on our DEC PDP-1 computer system by Charles R. Morgan. Michael Levin, one of the original implementors of LISP, contributed to the design. From September 1968 through November 1969, the National Science Foundation supported us in the first intensive program of experimental teaching of Logo-based mathematics in elementary and secondary classrooms. The teaching experiments demonstrated in principle that Logo can be used to provide a natural conceptual framework for the teaching of mathematics in an intellectually, psychologically and pedagogically sound way.

In 1970 Seymour Papert founded the Logo Laboratory at M.I.T. Logo-based turtles, erroneously thought by many to be essential to the Logo teaching enterprise, were introduced around 1971. Several hardware implementations and teaching experiments followed during the decade of 1970s at M.I.T., BBN and elsewhere.

Then came microcomputers. Their wide availability and affordability catapulted Logo into becoming one of the world's most widely used languages. At this point in its history, future development of exemplary teaching ideas and materials is crucial to support the needs of users. Our work on Logo development is continuing and continues to be compelling. Logo is an idea whose time is now.