Good to Think With:
Educational Visions and the Materiality of Computing

Janet Abbate, Virginia Tech

Paper presented at the 2015 Annual Meeting of the
Society for the History of Technology, Albuquerque, NM

Teaching children to program computers has become a hot topic in US education in recent years, with various "coding" events held around the country, attempts to include CS in high school curricula, and the declaration by Congress in 2009 of the first week of December as Computer Science Education Week. I am interested in unpacking the assumptions about computer science that underlie these policy efforts. Why is computer use seen as an essential skill? What is it, exactly, that we hope children will learn?

These are not new questions: educational computing efforts date back to the early 1960s, a formative moment when what we would now call "STEM" education became a national priority. Taken off guard by the 1957 launch of the Soviet Sputnik satellite, many Americans feared that the US had fallen behind their Cold War rival in science and technology. In 1967 President Lyndon Johnson, guided by his science advisory committee, became an advocate of using computers to improve education and directed the National Science Foundation to begin funding research on educational computing.

While some of the early attempts at using computers in schools were simple "drill and practice" systems, others envisioned using the power of computers to transform pedagogy.[1] Two major projects were PLATO at the University of Illinois and Logo at BBN and MIT. PLATO ran on a supercomputer and used advanced graphical displays and a network to deliver educational experiences to students from elementary school to college. [**IMAGE**: Plato terminal.] Logo was a programming language that was designed to provide an intuitive way for children to program a computer; later Logo systems allowed children to control the actions a simple robot called a "turtle." [**IMAGE**: Logo turtle.] Both PLATO and Logo were widely adopted and extensively discussed in the pedagogical literature, and versions of these

[1] Patrick Suppes at Stanford was a major researcher on drill-and-practice systems based on behavioral psychology. Patrick Suppes, "Computer-Assisted Instruction at Stanford," in *Psychology and Education Series* (Institute for Mathematical Studies in the Social Sciences, 1971).

systems continued to be used into the 21st century. These systems helped set expectations for educational computing that continue to influence policy.

These systems were predicated on the idea that the new computing technology could transform teaching. Yet the technical choices made in the two projects created very different computing environments, whose capabilities and constraints shaped the learning experiences offered by Logo and PLATO. I suggest that the *materiality* of educational computing—the hardware and software that teachers and students actually encountered—had unintended effects on pedagogical aims and methods. In this paper I will examine 1) How was the choice of technology for each project shaped by particular pedagogical visions? and 2) How did those technical choices in turn *reshape* popular and professional beliefs about how computers could improve education? I will conclude by asking what lasting influence these early projects have had on current discourse about educational computing.

PLATO

PLATO (Programmed Logic for Automatic Teaching Operations) was created at the University of Illinois at Urbana-Champaign in 1960. The project grew out of the Coordinated Sciences Laboratory (formerly Control Systems Laboratory), which had a strong engineering focus. It was initiated by physics professor Daniel Alpert and designed and built by Donald Bitzer, a 25-year-old engineering PhD student and lab assistant who went on to become a faculty member. [**IMAGE**: Bitzer]

PLATO's designers saw it as both an educational project and an engineering project. As an *engineering* project, PLATO's goals were strongly shaped by its creators' consciousness of the limitations of available technology. They spent much of their energy devising new and better hardware to support the use of multimedia content in classes. As an *educational* project, PLATO was presented as a radical shift from computer-controlled "drill and practice" lessons to more open-ended experiences that would allow students to develop "intuition and insight" (Alpert and Bitzer 1970, 1584). But beyond this, PLATO's creators did not promote a specific pedagogical approach. Alpert and Bitzer described their research method as building a prototype system with sophisticated technology and then asking, "What is educationally possible [with such a system]?" (Alpert and Bitzer 1970, 1583). They left it up to individual teachers to provide answers to that question, noting that the PLATO "made it easy for educational innovators to use their intuitive notions to develop wholly new sets of teaching or testing strategies" (Alpert and Bitzer 1970, 1585). As two early adopters at UI

commented in a 1976 article, "The character of PLATO lesson material varies greatly since the computer system does not impose a pedagogical structure on the authors of the materials" (Smith and Sherwood 1976, 344).

Bitzer saw the physical interface as a crucial factor for the success of computers as an educational tool. The usual interface at that time was a teleprinter—basically a typewriter—which could only display one line at a time, and the output was fixed on paper rather than displayed dynamically on a screen. Bitzer's initial version of PLATO substituted a television and a small custom-built keyboard wired to the university's ILLIAC computer. PLATO II, built in 1961, had two major advances: remote access and support for multiple users. These were aimed at making the system available to large groups at remote locations, a necessary step if the ultimate goal was widespread use of computers in schools. Bitzer arranged a high-profile demonstration of the system, with two terminals being used at once, as part of a educational conference presided over by the university President. [**IMAGE**: PLATO II demo.] The demonstration and subsequent improvements generated support for further development and distribution of the PLATO system. William Norris, head of the computer manufacturer Control Data Corporation, became interested in the project and donated a CDC 1604 computer to run PLATO III, which came out in 1963. In 1966 the university established a Computer-Based Education Research Laboratory, with Bitzer as director, to house the project. In 1968 CERL make a successful proposal to NSF to fund the fourth and final version of PLATO, which was finished in 1972.

Plato IV had many impressive technical features. To support instructional material with rich graphics, Bitzer and two PhD students, H. Gene Slottow and Robert Willson, invented the plasma display panel in 1964; commercial prototypes of the terminal were produced in 1970 by Owens-Illinois of Toledo, Ohio (Alpert and Bitzer 1970, 1588). The display had two glass panels with gas enclosed between them; a grid of electrodes could stimulate selected pixels, making them glow orange. [**IMAGE**: plasma display.] This allowed the system to display two-dimensional graphics and animations. (This was the basis of today's flat-screen plasma TVs, which earned the three inventors an Emmy Award in 2002.) But there was more: the glass could also be used as a screen on which photographs could be projected from the rear of the display. The photos were stored on microfiche and specific images could be displayed using program commands. [**IMAGE**: PLATO IV display diagram.] Still photos or slides of text could thus be superimposed on dynamic computer-

generated images for a richer visual effect. A final innovative feature was a touch-screen, which allowed students to select actions or answers to questions by pointing. [**IMAGE**: touchscreen.] In other words, the Plato IV display had many of the features of the tablet computers being used in schools today—but about two decades before tablets became commercially available.

Another radical hardware improvement came in sound production, to support music instruction as well as composition. In 1972, Sherwin Gooch created the Gooch Synthetic Woodwind, a synthesizer that could be used to create and play up to four musical voices (later expanded to 16). In keeping with the PLATO philosophy of having teachers create their own courseware, Gooch provided not just the synthesizer hardware but also a programming language that PLATO authors could use to easily control the synthesizer and a set of supporting programs, debuggers, and music files (Gooch 1978).

On the software side, zoology graduate student Paul Tenczar created the TUTOR programming language for PLATO in 1967, designed "specifically for use by lesson authors lacking prior experience with computers."[2] TUTOR's command set was tailored to course preparation and the easy display of graphics. The TUTOR manual for teachers emphasized that *they* were experts and that PLATO would allow them to fulfill, rather than replace, their pedagogical goals: "The instructor is to some extent an expert in teaching his material. He generally has at least a notion of an ideal method or methods by which this material could be presented to maximize learning. Computer-based education should be considered as merely another medium which might allow some of these methods to be used in practice" (Avner and Tenczar 1969, 7).

Another software innovation that turned out to be very influential was created almost by accident. PLATO had a program that allowed users to report system bugs. In 1973, David R. Woolley expanded this feature to create PLATO Notes, which allowed multiple people to respond to a bug report. This permitted an ongoing conversation, similar to today's listservs. Woolley broadened the system to allow conversations on other topics

[2] R. A. Avner and Paul Tenczar, "The Tutor Manual," (CERL, University of Illinois—Urbana, 1969)., 1. "Teachers ... can begin to prepare, edit, or modify lesson materials after a few hours of familiarization with the Tutor language" (Alpert and Bitzer 1970, 1585).

besides system bugs, and it quickly became popular.[3] (Woolley later went to work at Lotus Corporation and created the commercial product Lotus notes based on PLATO Notes.) A variation of the program called Personal Notes allowed one-to-one conversations; other programs called Talkomatic and Term-talk created multi-user chat rooms. These programs added a social dimension to the PLATO system that was not part of the original educational vision. As Joy Rankin has argued, PLATO and other time-sharing educational projects inspired "social computing" practices long before the World Wide Web (Rankin 2014).

PLATO was influential both directly, through use of the system, and indirectly, through adoption of its techniques by other computer projects. For example, researchers from Xerox PARC were given a tour of the PLATO system and incorporated some of its ideas for graphics and communications programs into their own personal computer system, the Alto, which in turn was a model for Apple computers. Control Data licensed the right to sell commercial versions of the PLATO hardware and software from 1970 to 1983 (University of Illinois archives 2011). In 1985, CERL staff members launched University Communications Inc. as a for-profit business to market PLATO; their system, called NOVANet, used satellite technology to distribute PLATO programming (University of Illinois archives 2011). NOVANet continued to operate NOVANet until 2015. Several other companies were licensed to manufacture or sell PLATO technologies.

How did the opportunities and limitations of PLATO technology shape educational uses?

While PLATO did not dictate a particular teaching approach, its novel capabilities did shape the educational visions and methods of instructors and the PLATO staff who supported them. A key technique that emerged from PLATO was the use of simulation and dynamic models to support an investigative, problem-solving approach characteristic of the natural sciences. Two science educators who were early PLATO adopters described this type of learning as essential, yet hard to achieve in the traditional classroom: "Since much of science is based on the results of experiments, it seems important to have students learn to design experiments and interpret the experimental data. However, many of the key experiments in the development of important concepts cannot be carried out by large

[3] These developments parallel similar events in the history of the Internet, where email was introduced in 1972 as a minor feature and quickly became the most popular use of the network. (Abbate 1999, chapter 3).

numbers of students because of the lack of adequate equipment, time, and experimental technique" (Smith and Sherwood 1976, 345). PLATO's plasma display, combined with the calculating power of a supercomputer, made it possible to create elaborate simulations including animated graphics. Students could change variables in the computer simulation and then "see" the results of their experimental actions on the screen (Smith and Sherwood 1976, 345). [**IMAGE**: chemistry lab simulation]. As the TUTOR manual explained, "each student can collect his own data without fear of damage to himself or apparatus. Since the time scale of a model laboratory can be shortened, the student does not have to wait hours, days, or even years for actual experimental conditions to occur" (Avner and Tenczar 1969, 2). By allowing students to do calculations on the computer, the system also eliminated drudgery: "Freed from tedious calculations, the student can rapidly explore the important relationships among elements of a problem" (Avner and Tenczar 1969, 4). Simulation could thus allow students to experience what it was like to think and behave like a scientist, without the years of training or advanced equipment they would have needed to achieve those results in an actual laboratory.

PLATO also facilitated student-driven learning, even when used within a standard curriculum. Teachers could present an open-ended problem, such as identifying an unknown chemical substance, and give students the initiative to decide what steps to take to solve it (Alpert and Bitzer 1970, 1584). The student would type in a question or request an action and the computer would answer the question or run a simulation of the specified test and present the results. Typically the student would go through several rounds of questions or tests in order to solve the problem; different students could take different paths to a correct solution, working at their own pace and guided by their own knowledge and intuition. There was also a *constructive* element, as students could create their own graphical objects as part of a lesson. For example, a lesson on optics might allow the student to specify the shape of a lens, then display it on the screen so that the student could see the lens as well as a simulation of how it would refract light (Several other companies were licensed to manufacture or sell PLATO technologies, including Owens-Illinois, Electronic Information Systems, Carroll Manufacturing, and Global Information Systems Technology [University of Illinois archives 2011, 5]).

PLATO's advanced graphics also gave teachers the ability to structure lessons as games to make learning more engaging. For example, a math game challenged children to combine

three random numbers arithmetically to produce a total that would move them ahead toward the finish line and avoid obstacles; two players could compete to see whether the train or the stagecoach would win (Solomon 1988, 41). The influence of PLATO games went beyond teaching: users created for their own amusement some of the first multiplayer online games, drawing on the system's graphics, time sharing, and communication features. [**IMAGE**: Avatar game, created 1979.]

Yet the very sophistication of the PLATO technology was also a constraint, because it raised the cost of the system beyond what many schools could pay. The terminal alone cost about $6000 in 1976 (Smith and Sherwood 1976, fn 19), and the price never came down as much as its creators hoped—though even if it had, their predicted $1800 price would still be too much for schools to buy more than a few terminals for their students. In addition, schools would have to pay for computer time, which was expensive if purchased from a commercial provider, as well as the cost of phone connections to the computer. A study in the mid-1970s that questioned the cost-effectiveness of PLATO let to cuts in funding for grade-school experiments. In the absence of an explicit educational philosophy focused on young children, the PLATO designers tended to confine their vision for PLATO in elementary schools as merely "drill and practice in arithmetic and .... reading skills," in contrast to their excitement about its "particularly valuable role... at the undergraduate level" (Alpert and Bitzer 1970, 1586).

Logo

Logo differed from PLATO in having a strong pedagogical vision and in its specific focus on young children. The Logo project was started by computer scientist Wallace Feurzeig at Bolt Beranek and Newman, a Cambridge acoustics and computing firm with close ties to MIT. Feurzeig joined the company in 1962 and worked on computer-aided instruction (CAI) systems under J. C. R. Licklider. Feurzeig collaborated with psychologist John Swets on an educational program called "Socratic System" (Feurzeig 2011). Unlike the usual drill-and-practice programs where students were asked to answer questions, the Socratic System allowed students to *ask* questions in order to explore and solve a problem. Their showcase application allowed doctors to practice differential diagnosis by asking questions about a hypothetical patient's symptoms.

Two technical developments in 1964 changed the direction of Feurzeig's research from

training systems for adult professionals to learning environments for children. One was time sharing operating systems, which reduced the cost of interactive computing by allowing many people to use the computer at once; this made it economically feasible to provide interactive computing in schools. The second was the introduction of so-called 'conversational' programming languages, which were interactive, interpreted languages that gave immediate feedback to the programmer. The new languages inspired Feurzeig to try using programming itself as an educational experience.

Feurzeig's project began with a specific focus on "making mathematics more accessible and interesting to beginning students" (Feurzeig 2011, 290). In addition to the post-Sputnik emphasis on STEM education, the so-called "new math" curriculum had been introduced in the early 1960s, disrupting existing teaching methods and making math education an obvious target for intervention. Feurzeig's group was funded by the U.S. Office of Education to experiment with teaching children to write programs in a 'conversational' language to solve problems in arithmetic and algebra. After this successful experiment, Feurzeig initiated a project in 1966 to create a new language specifically designed for teaching, which he would name Logo.

Two members of his team had done work at the MIT Artificial Intelligence laboratory, and they suggested to Feurzeig that he bring in as a consultant the AI Lab's co-director: Seymour Papert (Yazdani 1987, 11). Papert was a native of South Africa who earned a doctorate in mathematics at Cambridge 1958. As part of his doctoral work, Papert spent a year at the Poincaré Institute at the University of Paris, where he met the Swiss psychologist Jean Piaget and became interested in his theories of learning. After Papert completed his PhD in 1958 Piaget invited him to work at his International Centre of Genetic Epistemology at the University of Geneva. Papert spent four years with Piaget and "became passionately interested in children's thinking" (Papert 1993, 32-33). Piaget had developed a theory of learning called *constructivism*, which held that, rather than passively receiving knowledge, children actively constructed their own theories about the world through their physical interaction with it. Papert later developed his own version of the theory, which he called *constructionism* to emphasize that the optimal way for children to build *mental* models was through constructing *physical* or computer-generated models (Papert and Harel 1991).

Papert met Marvin Minsky in 1960 when they both attended the London Symposium on Information Theory. [**IMAGE:** Minsky & Papert, 1971.] Minsky invited Papert to visit

MIT later that year, where Papert got his first experience programming a computer (Papert 1998). Papert was amazed that the computer allowed him to quickly solve a complex mathematical problem that he had been struggling with. As he later recalled,

> "There I was euphoric with this sense of empowerment that this intellectual tool had given me. And since at the time my regular job was at the University of Geneva teaching and researching with Piaget, children were close to my mind and an obsession was born. The obsession was that children should get that intellectual power and that sense of thrill that I got" (Papert 1998).

In 1962 he joined Minsky at MIT as a mathematics professor and co-director of the AI Group (replacing John McCarthy, who had recently left MIT for Stanford). Given Papert's newfound interest in computing for children, he was a natural choice to join the Logo project. Though Papert and Feurzeig had different backgrounds, both had collaborated with psychologists and were committed to making computer systems that were "good to think with."[4]

Feurzeig and Papert guided the overall design for Logo. Rather than create a language from scratch, they decided to base Logo on the LISP programming language, which had been created by John McCarthy and was in use at both MIT and BBN. LISP was a 'functional' language, which made it easy to write modular code: a set of commands to perform a particular action—for example, drawing a square—could be defined as a unit of code called a "function" that could then be reused in other programs. This would allow children to create simple building blocks of code and then combine them to generate complex images or behaviors. But while LISP was powerful, it was also considered difficult to learn (Feurzeig 2011, 291), so Logo was envisioned as a simplified version that would be more accessible to children. The Logo system at this point had no special hardware: it used ordinary teletype terminals connected to a time-sharing computer at BBN (Papert and Harel 1991).

The first version of Logo was created in 1966-1967. After some field tests and

[4] The expression "good to think with" is from Levi-Strauss, *Totemism*, referring to the use of animals as symbols or metaphors for human society. Papert referred to the Logo turtle as "good to think with" in *Mindstorms* (11). Papert's later collaborator, Sherry Turkle, referred to the computer as "good to think with" in *The Second Self: Computers and the Human Spirit* (18).

improvements, the BBN team got an NSF grant to do a year-long test in 2nd- and 7th-grade math classes in 1968-1969 (Feurzeig 2011, 291). After successful field tests, the efforts of Feurzeig and Papert diverged somewhat. Feurzeig and his group at BBN continued to improve the technology used to support Logo, as well as embarking on other educational projects. Papert relocated his efforts to MIT, founding its Logo Laboratory in 1970, and became the public face of Logo, publishing the popular book *Mindstorms* in 1980 to introduce the world to Logo and the educational philosophy behind it.

Papert described that philosophy as "creating the conditions under which intellectual models will take root"; he saw the computer as a tool that could allow children to build and test mental models (Papert 1980*,* forward). Programming also let children *externalize* these mental models in the form of code, so that so that they could discuss, assess, and improve their own thought processes, and consciously apply strategies that they had learned through programming to other intellectual problems (Papert and Harel 1991). Thus programming was seen as a way to teach "general thinking skills" (Yazdani 1987, 11). This was quite different from learning a *curriculum*, as in traditional education. Rather than absorbing a set body of knowledge, children in Logo classes would decide for themselves how they wanted to use the computer, and would build new theories spontaneously. [**IMAGE:** Logo classroom.] Papert believed that once children had developed the fundamental thinking skills fostered by Logo, it would be relatively easy for them to learn any specific knowledge that they needed.

Logo's main hardware innovation was added in 1971: a computer-controlled robot called a "turtle." Children could use program commands to make the turtle move across the floor and draw with a pen. [**IMAGE**: Logo turtle.] A virtual version of the turtle that drew images on a computer screen was added in 1972. For Papert, the turtle played an important pedagogical role, because he adhered to Piaget's theory that children learn from concrete experience. The child programmer could imagine him- or herself as the turtle and get an embodied sense of how the program should work. Papert called the turtle a "transitional object" that helped children translate concrete experiences into abstract ideas (Papert 1980). The turtle soon became iconically associated with Logo.

In addition to the charismatic Papert's lectures and books, an important factor in popularizing Logo was the work of BBN and MIT programmers to implement it on widely-used computer systems. In the early 1970s, BBN software engineers created a version of

Logo for the widely-used DEC PDP-10 computer and made it available for free to "over 100 universities and research centers" (Feurzeig 2011, 294). As microcomputers were introduced in the late 1970s and 1980s, other engineers adapted Logo for this new wave of computer technology. In 1980 the MIT Logo group spun off a commercial arm, Logo Computer Systems, Inc., with Papert as chairman, which created popular Logo products that are still produced and sold today.

How did the opportunities and limitations of Logo technology shape educational uses?

As with PLATO, technical design choices for the Logo system influenced how and by whom it was used. Unlike PLATO, Logo did not rely on expensive special-purpose hardware: its only non-standard component was the turtle robot, which was optional, and the Logo software did not require a powerful computer to run. This allowed the developers to switch in the early 1980s from using large time-sharing computers to personal computers, which made Logo affordable for schools. Not only were PC prices rapidly falling, but having the Logo software run locally on the PC meant there would be no telecommunications costs or time-sharing computer charges to pay. Moreover, personal computer manufacturers donated machines to many schools in order to increase their user base. Feurzeig could note with satisfaction that "The advent of micro-computers, with their wide availability and affordability, catapulted *Logo* into becoming one of the world's most widely used computer languages during the 1970s and 1980s" (Feurzeig 2011, 294).

But Logo could be difficult to use in other ways. While teachers using PLATO could begin writing courseware after an hour of training, teachers who adopted Logo had to substantially change their pedagogical approach. Rather than teachers introducing a concept and having students practice it, students were supposed to take the lead in computer explorations that would lead them to construct mathematical concepts. Yet paradoxically, this process would not be effective without carefully planned activities and skilled intervention by teachers in the learning process. As a meta-study of Logo research reported in 1997,

> "'exposure' [to Logo] without teacher guidance often yields little learning. ... Studies that have shown the most positive effects involve carefully planned sequences of Logo activities. Teacher mediation of students' work with those activities is necessary.... They need to (a) focus students' attention on particular aspects of their experience, (b)

> reduce informal language and provide formal mathematical language for the mathematical concepts, (c) suggest paths to pursue, (d) facilitate disequilibrium using computer feedback as a catalyst, and (e) continually connect the ideas developed to those embedded in other contexts" (Clements and Sarama 1997).

While some teachers found the new methods exciting and liberating, to embrace Logo as an intellectual tool for children—and not simply as a programming exercise—took a commitment of time, effort, training, and institutional support that not all teachers had.

As noted earlier, the Logo turtle—though not part of the initial plan for the system—quickly became one of the most recognizable and enduring aspects of the system. The popularity of the turtle and its successors led to a subtle shift in emphasis from building fundamental mathematical concepts to controlling robots. In the 1990s the MIT Logo group collaborated with the Lego company to incorporate a small computer into a large Lego brick, allowing children to build elaborate machines and robots out of Legos and control them with Logo programs. A commercial kit called "Lego Mindstorms" as released in 1998 and continues to be extremely popular, described by its manufacturer as "the best-selling product in the LEGO Group's history" (http://www.lego.com/en-us/mindstorms/history). While Lego Mindstorms can be used for the kind of exploration and learning envisioned by Papert, the commercial versions—like other Lego products—steer the user toward building pre-specified figures. Commentators have also noted that these kits are targeted toward boys.[5] [**IMAGE:** Lego Mindstorms web page.]

Conclusions and Open Questions

Summary and comparison of cases:

In this brief overview of two early educational computing projects, I have tried to uncover the co-construction of computer technology and pedagogy. Both projects pioneered the use of computers for student-driven, creative learning rather than rote drill. Both projects believed in the power of visual, animated feedback—whether a moving turtle-robot or a simulated chemical reaction—to help students understand scientific concepts and to

[5] Audrey Watters. "Lego Mindstorms: A History of Educational Robots." Blog post,10 Apr 2015. http://hackeducation.com/2015/04/10/mindstorms/; MIT Media Lab professor Leah Buechley, quoted in Bobbie Johnson, "ETech: Getting Girls into Computing with Mit's High-Low Tech" http://www.theguardian.com/technology/blog/2009/mar/12/research-gadgets.

give students a feeling of hands-on involvement and agency.

They differed in their vision of the teacher and the learning process, which guided the design of the language used in each system. PLATO was in many ways a teacher-centered system: it was designed to give teachers new capabilities, to allow them to create their own software without being dependent on programmers, and to encourage collaboration among courseware authors. Alpert & Bitzer believed that teachers knew how to create meaningful learning experiences, and PLATO's TUTOR language was designed to give teachers the tools to do so. Papert, in contrast, was convinced that teachers needed to adopt a new, student-led approach to education (or at least mathematics), and the Logo language was designed for children to use. The very role of a programming language was fundamentally different. In the Logo world, students would learn not by *interacting* with a program but by *writing* a program. The logical structures of computer code were seen as general-purpose intellectual concepts that students could apply to other areas. Learning to think like a Logo programmer was the same as learning to think like a mathematician—which was, by implication, the same as learning to think.

These examples also show the influence of disciplinary paradigms. Papert's focus on acquiring fundamental concepts of mathematics downplayed the importance of memorizing facts. In the PLATO worldview, where natural science was the paradigm for learning, education meant mastering a body of knowledge and procedure as well as developing intuition. This justified the emphasis on high-quality graphics, since much of the information in science is visual. [**IMAGE**: PLATO showing fruit fly mutations.] Likewise, animated simulations allowed students to practice investigative procedures so that they could internalize the culture of laboratory science.

To return to my initial question: What did the advocates of these projects hope children would learn by using computers? For PLATO, the answer ranged from "the usual curriculum" to "a much richer laboratory experience." For Logo, it ranged from "fundamental mathematical concepts" to "general thinking skills, and an inclination to use them." How were they actually used? I have tried to discern how the initial educational goals were refracted through the lens of available technology, emerging with a change in direction that may have gone unacknowledged.

Open questions for today's educational computing:

Historical legacy: To what extent is current thinking about the role of computers in education still influenced by ideas that crystallized during the 1960s? A large number of people were exposed to these projects, either as teachers or students; how did that experience shape the ideas of the next generation of educational thinkers?

Technology: Both of these projects started with the assumption that it was necessary to build new technology in order to effectively use computers for education. Specific design choices in each project reveal their agendas for shaping both the role of teachers and the experiences of students. What specialized technology is being developed today, and what agendas are embedded in it?

Policy: A final open question concerns the role of policy. In the background of both projects was a public policy imperative to improve the performance of American children in scientific subjects. But neither project aimed at teaching children computer skills as an end in itself—as a way to make a living and keep the US competitive with its geopolitical rivals. Today we again hear calls for STEM education, but there is much more emphasis on teaching coding as a marketable skill that will benefit both the student and the US economy. How has this change in attitude affected the way today's educational computing projects are conceived and carried out? While coding activities for young children may be playful, they are also seen as a way to entice kids to learn a utilitarian skill. Perhaps the computer has become less "good to learn with" and more "good to earn with."

References

Alpert, D., and D. L. Bitzer. "Advances in Computer-Based Education." *Science* 167, no. 3925 (1970): 1582-90.

Avner, R. A., and Paul Tenczar. "The Tutor Manual." CERL, University of Illinois, 1969.

Clements, D. H., and J. Sarama. "Research on Logo: A Decade of Progress." In *Logo: A Retrospective*, ed. C. D. Maddux and D. L. Johnson (The Haworth Press, 1997), 9-46.

Feurzeig, Wallace. "Educational Technology at BBN." In *A Culture of Innovation,* edited by D. Walden and R. Nickerson, 281-351. Waterside Publishing, 2011.

Gooch, Sherwin. "Plato Music Systems." In *Annual Meeting of the Association for the Development of Computer Based Instructional Systems*. Dallas, Texas, 1978.

Papert, Seymour. "Child Power: Keys to the New Learning of the Digital Century." (1998), http://www.papert.org/articles/Childpower.html.

———. *The Children's Machine: Rethinking School in the Age Ot the Computer*: Basic Books, 1993.

———. *Mindstorms*: Basic Books, 1980.

Papert, Seymour , and Idit Harel. "Situating Constructionism." In *Constructionism*: Ablex Publishing, 1991.

Papert, Seymour, and Idit Harel. *Constructionism*: Ablex Publishing, 1991.

Rankin, Joy. "Toward a History of Social Computing: Children, Classrooms, Campuses, and Communities." *IEEE Annals of the History of Computing*, April-June (2014): 86-88.

Smith, Stanley G., and Bruce Arne Sherwood. "Educational Uses of the Plato Computer System." *Science* 192, no. 4237 (1976): 344–52.

Solomon, Cynthia. *Computer Environments for Children*. Cambridge: MIT Press, 1988.

Suppes, Patrick. "Computer-Assisted Instruction at Stanford." In *Psychology and Education Series*: Institute for Mathematical Studies in the Social Sciences, 1971.

University of Illinois archives. "Computer-Based Education Research Laboratory: Historical Note." (2011) http://archives.library.illinois.edu/

Yazdani, Masoud. *Artificial Intelligence and Education: Learning Environments and Tutoring Systems*: Intellect Books, 1987.