Styles and Voices

SHERRY TURKLE, SEYMOUR PAPERT

The prevailing image of the computer represents it as a logical machine and computer programming as a technical, mathematical activity. Both the popular and technical culture have constructed computation as the ultimate embodiment of the abstract and formal. Yet the computer's intellectual personality has another side: our research finds diversity in the practice of computing that is denied in action by its social construction. When we looked closely at programmers in action we saw formal and abstract approaches; but we also saw highly successful programmers in relationships with their material that are more reminiscent of a painter than a logician. They use concrete and personal approaches to knowledge that are far from the cultural stereotypes of formal mathematics.

We have studied computers and the cultures that grow up around them in a wide variety of settings ranging from video game arcades to research laboratories of artificial intelligence. In this paper [1] we draw particularly on a long-term line of research on how people enter the culture of programming. Using clinical methods inspired by the Piagetian and psychoanalytic traditions, we built up case studies of children using computers in grade school settings and college students taking a first programming course. We saw many manifestations of the concrete approach, favored in our study by more women than men.[2] We were also able to observe people reacting poignantly to what they felt as a pressure to conform to an officially imposed style. Although the computer as an expressive medium supports epistemological pluralism, the computer culture often does not. Our data points to discrimination in the computer culture that is determined not by rules that keep people out but by ways of thinking that make them reluctant to join in. Moreover, the existence of diverse styles of *expert* programming supports the idea that there can be different but equal voices even where the formal has traditionally appeared as almost definitionally supreme: in mathematics and the sciences.

The computer forces general questions about intellectual style to reveal an everyday face. Even schoolroom differences in how children program computers raise issues that come up in a more abstract form in scholarly debates about scientific objectivity. The computer makes ideas about noncanonical scientific voices more concrete and therefore appropriable because we can relate them not only to the science of the scientists but to our own thinking.

Here we focus on descriptions of a concrete way of knowing; the formal, canonical style is well known and well defended. Yet, our discussion of concrete approaches is implicitly a discussion of formal ones; it contributes to the deconstruction of the canonical style as the only way to think. It also situates it: the supervaluation of the formal approach owes much of its strength within computation to the support it gets in other intellectual domains. Formal thinking, defined as synonymous with logical thinking, has been given a privileged status that can be challenged only by developing a respectful understanding of other styles, where logic is seen as a powerful instrument of thought but not as the "law of thought." In this view, "logic is on tap not on top." As a carrier for pluralistic ideas about approaches to knowledge, the computer may hold the promise of catalyzing change not only within the computer culture but in the culture at large.

Bricolage

Lévi-Strauss used the term *bricolage* to contrast the analytic methodology of Western science with what he called a "science of the concrete" in primitive societies.[3] The bricoleurs he described do not move abstractly and hierarchically from axiom to theorem to corollary. Bricoleurs construct theories by arranging and rearranging, by negotiating and renegotiating with a set of well-known materials.

Lévi-Strauss's descriptions of the two scientific approaches, divested of his efforts to localize them culturally, suggest the variety of ways that people approach computers. For some people, what is exciting about computers is working within a rule-driven system that can be mastered in a top-down, divide-and-conquer way. This is the "planner's" approach which decrees that the right way to solve a programming problem is to dissect it into separate parts and design a set of modular solutions that will fit the parts into an intended whole. Some programmers work this way because their teachers or employers insist that they do. For others, it is a preferred approach; to them, it seems natural to make a plan, divide the task, use modules and subprocedures.

Some of the students we have worked with offer examples of a very different style. They are not drawn to structured programming; their work at the computer is marked by a desire to play with the elements of the program, to move them around almost as thought they were material elements — the words in a sentence, the notes in a musical composition, the elements of a collage.

The bricoleur resembles the painter who stands back between brushstrokes, looks at the canvas, and only after this contemplation, decides what to do next. For planners, mistakes are missteps; for bricoleurs they are the essence of a navigation by mid-course corrections. For planners, a program is an instrument for premeditated control; bricoleurs have goals, but set out to realize them in the spirit of a collaborative venture with the machine. For planners, getting a program to work is like "saying one's piece," for bricoleurs it is more like a conversation than a monologue. In cooking, this would be the style of those who do not follow recipes and instead make a series of decisions according to taste. While hierarchy and abstraction are valued by the structured programmers' planner's aesthetic, bricoleur programmers prefer negotiation and rearrangement of their materials.

For instance, Alex, nine years old, is a classic bricoleur. He attends the Hennigan Elementary School in Boston, the scene of an experiment in using computers across the curriculum. There students work with Logo programming and computer controlled Lego construction materials. The work is both frequent enough (at least an hour a day) and open-ended enough for differences in styles to emerge.

When working with Lego materials and motors, most children make something move by attaching wheels to a motor that makes them turn. They see the wheels and motor through abstract concepts of what they are for: the wheels roll, the motor turns. Alex goes a different route. He looks at the objects concretely, without the filter of abstractions. He turns the Lego wheels on their sides to make flat shoes for his robot and harnesses one of the motor's most tangible features: the fact that it vibrates. When a machine vibrates it tends to travel, something normally to be avoided. When Alex runs into this phenomenon, his response is to make his robot (stabilized by its flat "wheel shoes") vibrate and thus move forward. When Alex programs in Logo he likes to keep things similarly concrete.

In his own way, Alex has resisted the pressure to believe the general superior to the specific or the abstract superior to the concrete. For Alex, thinking about hands as a subset of arms is too far away from the reality of real hands, just as taking a motor that was most striking as a vibrating machine and using it to turn wheels in the standard fashion was too far away from the real motor he had before him. While the structured programmer starts with a clear plan defined in abstract terms, Alex lets the product emerge through a negotiation between himself and his material.

The bricoleurs in our study tend to prefer the transparent style, planners the opaque, but the program's authorship is a critical variable in this preference. Planners want to bring their own programs to a point where they can be black-boxed and made opaque, while bricoleurs prefer to keep them transparent; but when dealing with programs made by others, the situation is reversed. Now, the bricoleurs are happy to get to know a new object by interacting with it, learning about it through its behavior the way you would learn about a person, while the planners usually find this intolerable. The planners' more analytic approach demands knowing how the program works before interacting with it. They demand the assurance that comes from transparent understanding, from dissection and demonstration.

Within feminist scholarship there is a substantial body of literature that challenges the notion that human reason best expresses itself within terms of Western male gender norms. For example, Carol Gilligan's work on moral reasoning calls into question the idea of one privileged, mature way of thinking [4] Gilligan's description of diverse approaches to moral reasoning is analogous to our contrast between the formal, canonical approach to programming and the concrete style of the bricoleur. In the first, justice is like a mathematical principle: to solve a problem you set up the right algorithm, the right black box, you crank the handle, and the answer comes out. In the second, a contextualized arguments is like a concrete argument, one needs to stay in touch with the inner workings of the arguments, with the relationships and possible shifting alliances of a group of actors whose interests need to be negotiated.

Observation of programmers at work calls into question deeply entrenched assumptions about the classification and value of different ways of knowing. It provides examples of the validity and power of concrete thinking in situations that are traditionally assumed to demand the abstract. It supports a perspective that encourages looking for psychological and intellectual development within rather than beyond the concrete and suggests the need for closer investigation of the diversity of ways in which the mind can think with objects rather than the rules of logic.

Objects

Sooner or later in building objects with Lego, children we have worked with run into the need for gears. Looking at their work provides a good example of alternate styles applied to working with the same problem, formal styles that use rules and concrete styles that use objects

The motors in the construction set turn at a high speed with low torque. A car built by attaching these motors directly to the wheels will go very fast, but will be so underpowered that the slightest slope of obstruction will cause it to stall The solution to the problem with Lego cars is the same as that adopted by designers of real cars: use gears. Yet in order to use them effectively, children need to understand something about gear ratios.

If a small gear drives a larger gear, the larger gear will turn more slowly and with greater torque It is the relative and not the absolute size of the two gears that counts. But when we interview children, we find that some of them reason as if the size of only one gear matters, as if they were following a set of rules such as "large gears are slow and strong"' and "small gears are fast and weak." Without the notion of relative size, such rules fail. Other children, and in our study, predominantly the girls, are less articulate and more physical in their explanations. They squirm and twist their bodies as they try to explain how they figure things out; and they get the right answer.

Theorists who look at intellectual development as the acquisition of increasingly sophisticated rules would say that children run into problems if the rules they have built are not yet good enough. The idea of "closeness to objects" enables us to consider a different kind of theory. Our observations suggest that the children who did so well did not have better rules, but a tendency to see things in terms of relationships rather than properties, access to a style of reasoning that allowed them to imagine themselves "inside the system." They used a relationship to the gears to help them think through a problem.

This "reasoning from within" may not be adequate for all problems about gears, but for the kind of problem encountered by the children in our project, it was not only adequate, but much less prone to the errors produced by a too simple set of rules. Relational thinking puts you at an advantage: you do not suffer disaster if the rule is not exactly right.

We have defined bricolage as a style of organizing work that invites descriptions such as negotiational rather than planned in advance, what Warren McCulloch called "heterarchical" rather than hierarchical. The story of the children and their gears serves to introduce another characteristic displayed by many bricoleur programmers. We call this characteristic proximality or closeness to the object. There is little distance between such a programmer and her computational objects. Like the children, who reasoned from within with the gears, she psychologically places herself in the same space as the sprites. She experiences her screens and birds as tangible, sensuous, and tactile. She is down there, in with the sprites, playing with them like objects in a collage. When she talks about them her gestures with hand and body show her moving with and among them. When she speaks of them she uses language such as "I move here."

The object relations school of psychoanalysis focuses on the way development progresses by a process of internationalization of the things and people of the world. They come to live with us; they become the objects with which we think.[5] When psychoanalysts talk about objects they usually mean people. Here we extend the idea of internalized "objects to think with" to the domain of everyday relationships with artifacts. It is not enough to ask whether individuals "like" or "don't like" to program because that puts the question on too high a level of generalization. Liking to program depends on forging a personally meaningful relationship with a computational object, a relationship that "fits." In forging this relationship, there are several dimensions of choice. People can choose among computational objects. For example, some prefer the turtle, its static nature, the fineness in the way it draws. For others, these same qualities are reasons to reject the turtle as constraining, even unpleasant. They prefer the sprites, which move with flash and speed.

People can (and do) choose different ways of approaching the same object. Computational objects, like turtles and sprites, stand on the boundary between the physical and the abstract. You can see them, move them, put one on top of another. Yet, they are mathematical constructions. Canonical programmers treat a sprite more like an abstract entity, a Newtonian particle, while bricoleur programmers treat it more like a physical object, a dab of paint or a cardboard cut-out.

Computational objects offer a great deal to those whose approach to knowledge requires a close relationship to an object experienced as tactile and concrete. Some people are comfortable with mathematical exercises that manipulate symbols on quadrille-ruled paper. For many others, computational objects offer a physical path of access to the world of formal systems. For them, the ambivalent nature of computational objects may make possible a first access to mathematics.

Feminist critics have related the standard notion of scientific objectivity to the social construction of gender: objectivity in the sense of distancing the self from the object of study is culturally constructed as male, just as male is culturally constructed as distanced and objective. From this point of view, a proximal style of programming is countercultural, reminiscent of Evelyn Fox Keller's description of geneticist Barbara McClintock's intimate relationship to the objects of her scientific study.[6] For McClintock, the practice of science was essentially a conversation with her materials. The more she worked with neurospora chromosomes (so small that others had been unable to identify them), "the bigger [they] got, and when I was really working with them I wasn't outside, I was down there. I was part of the system. I actually felt as if I were right down there and these were my friends ... As you look at these things, they become part of you and you forget yourself."

In our research we find a close relationship between bricolage, a style of organizing work, and proximality, a style of relating to the objects of work. Our data are consistent with a model of styles as clusters of characteristics in which bricolage and proximality form the nucleus of one cluster ("concrete thinking") and planning and distality the nucleus of the other ("formal thinking"). These clusters are ideal types: our contention is not that the attributes in each cluster are exactly correlated but that each has internal coherency in the way that a stable culture is coherent.

So, for example, closeness to objects tends to support a concrete style of reasoning, a preference for using objects to think with, and a bias against the abstract formulas that maintain reason at a distance from its objects. Conversely, a distanced relationship with objects supports an analytic, rule- and plan-oriented style. Our theoretical conjecture is that degree of closeness to objects has developmental primacy; it comes first. The child forms either a proximal or a distant relationship to the world of things. The tendency to use the abstract and analytic or the concrete and negotiational style of thinking follows.

Gender, closeness and conflict

Several intellectual perspectives suggest that women would feel more comfortable with a relational, interactive, and connected approach to objects, and men with a more distanced stance, planning, commanding, and imposing principles on them. Indeed, we have found that many women do have a preference for attachment and relationship with computers and computational objects as a means of access to formal systems. yet in our culture computers are associated with a construction of science that stresses aggression, domination, and competition. The cultural construction of science leads to a conflict that considerably complicates our story of how women appropriate technology. In the case of computation, this conflict is particularly acute.

From its very foundations, science has defined its way of knowing in a gender-based language. Francis Bacon's image of the (male) scientist putting the (female) nature 'on the rack,' underscores the way objectivity has been constructed not only in terms of the distance of the knower from nature but also in terms of an aggressive relationship toward it (or rather toward her). From its very foundations, objectivity in science has been engaged with the language of power, not only over nature but over people and organizations as well. Such associations have spread beyond professional scientific communities; aggression has become part of a widespread cultural understanding of what it means to behave in a scientific way. Its methods are expected to involve "demolishing" an argument and "knocking it down" to size. Here the object of the blows is not a female nature but a male scientific opponent. Science as first a rape, then a duel.

The traditional discourse of computation has not been exempt from these connotations. Programs and operating systems are "crashed" and "killed." We write this paper on a computer whose operating system asks if it should "abort" an instruction it cannot "execute." In our ethnographic studies of the social worlds that grow up around computing, we have found that this is a style of discourse that few women fail to note. Thus, women are too often faced with the not necessarily conscious choice of putting themselves at odds either with the cultural associations of the technology or with the cultural constructions of being a woman.

When Lisa, a student in a college-level introductory course, first found herself doing well in her programming course, she found it "scary" because she felt she needed to protect herself from the idea of "being a computer science type." In high school, Lisa saw young men around her turning to computers as a way to avoid people: "They took the computers and made a world apart." Lisa describes herself as turning off her natural abilities in mathematics that would have led her to the computer. "I didn't care if I was good at it. I wanted to work in worlds where languages had moods and connected you with people." Although her classmate Robin had gone through most of her life as a musician practicing piano eight hours a day, she, too, had fears about guys who established relationships with the computer. "To me, it sounds gross to talk about establishing a relationship with the computer. I don't like establishing relationships with machines. Relationships are for people."

In the vehemence with which many women insist on the computer's neutrality, on its being nothing more than a mere tool, there may be something more subtle going on that a clash between culture and personal style—a clash between personal style and sense of self. Many women may be fighting *against* having a close relationship to a computer or to computational objects. For some, there is a clash because they want to belong to the dominant computer culture. But for others, the experience of closeness to the object is a source of conflict with themselves.

Lisa placed herself in the space of the computational objects she worked with, and she tended to anthropomorphize, responding to the computer as though it had (at least) an intellectual personality. Her own style came to offend her because it had led her to what she experienced as a too close relationship with a machine. When Lisa began programming she saw herself as communicating with the computer, but the metaphor soon distressed her. "The computer isn't a living being and when I think about communicating with it, well that's wrong. There's a certain amount of feeling involved in the idea of communication and I was looking for that from the computer." She looked for it and she frightened herself. "It was horrible. I was becoming involved with a thing. I identified with how the computer was going through things."

In our research we find that women express such sentiments with particular urgency. We observe that a conflict fuels their convictions. In many cases, they are most comfortable with a style of thinking in which they get close to the objects of thought. The computer offers them such objects, but the closer they get to them the more anxious they feel. One remedy for their anxiety is denial. The more these people become involved with the computer, the more they insist that it is *only* a neutral tool. Again, their assertion is belied by the vehemence with which it is expressed.

Lisa's conflict with her instructor would be resolved in principle by a greater tolerance for her way of thinking; but addressing internal conflicts about being close to computers requires more than tolerance. It requires profound changes in the culture that surrounds the computer. For instance, if the computer is a tool, and of course it is, is it more like a hammer or more like a harpischord?

The musician Robin is not distressed by her close relationship with her piano, which is also a machine. Lisa, who finds attachment to the computer "unnatural," is not upset by her passion for the beautiful, heavy antique ink pens with which she writes. If Lisa had been in music school, it is most likely that she, like Robin, would not experience as threatening her sense of communicating with her instrument or her emotional involvement with it. Music students live in a culture that over time has slowly grown a language and models for close relationships with music machines. The harpischord, like the visual artist's pencils, brushes, and paints, is a tool, and yet we understand that artists' encounters with these can (and indeed, will most probably) be close, sensuous, and relational. Indeed, the best artists will develop highly personal styles of working with them.

The development of a new computer culture would require more than environments where there is permission to work with highly personal approaches. It would require a new social construction of the computer, with a new set of intellectual and emotional values more like those applied to harpischords than hammers.

Notes

[1] This is an abridged extract from our article, "Epistemological pluralism," *Signs: Journal of Women in Culture and Society*, 16 11 (1990) 128-157; (published by the University of Chicago Press: © 1990 by the University of Chicago - all rights reserved).

[2] We worked with 40 grade school students: 14 out of the 20 girls, but only 4 out of the 20 boys, preferred concrete approaches. In the study of 30 college students, 9 out of 15 women, and 4 out of 15 men, were concrete style programmers. Some specific case studies are described in the article cited in note 1.

[3] C. Lévi-Strauss, *The savage mind,* University of Chicago Press, 1968

[3] C. Gilligan, *In a different voice*, Harvard University Press, 1982

[5] For an excellent overview of the object-relations perspective, see J.R. Greenberg & S.A. Mitchell, *Object relations in psychoanalytic theory,* Harvard University Press, 1983.

[6] E Fox Keller, *A feeling for the organism*, San Francisco: W H. Freeman, 1983, p117