

On Computers, Personal Styles, and Being Human: A Conversation with Sherry Turkle



Sherry Turkle is associate professor, Program in Science, Technology, and Society at the Massachusetts Institute of Technology and holds a joint doctorate in Sociology and Psychology from Harvard. In 1984, her book, *The Second Self: Computers and the Human Spirit*, provided new insights into the relationships that people form with computers and the ways in which these relationships affect values, ways of thinking about the world, and ways of seeing oneself and other people.

LEWIS A. RHODES

In the short time since your book was published, the interest in computers is said to have plateaued.

The problem is that the computer is thought of as the educational treatment and people try to measure its effects. There has been this new rash of surveys reporting that "kids don't like computers," or that schools are losing interest in them. That's a little like saying the child doesn't like pencils; the child doesn't like paper; the child doesn't like reading. Reading what? *McGuffey's Reader*?

Very sensitive, very smart people still treat the computer as though it were a *thing*, and that you simply have to measure its effects, when in fact it's only an element that lets you create different kinds of learning environments. For example, a computer with a word processing program—that allows you to manipulate your words in a kind of sculptural way—has nothing in common with a computer used as a kind of souped-up flash card or a video game learning environment.

The whole art and science of using computers in education is to look at the most exciting, the most forward thinking uses of the computer that could make a difference for kids—and not worry about whether the kids are supposed to like or not like the computer itself.

What would be a better way to think about computers?

We have to see the technology as the material from which we can make personal educational environments. We have to help people focus on the creation of these individual learning environments and have the experience of doing new kinds of manipulations that are synergistic with good learning.

What we've been doing is like telling teachers that we have this new instrument for writing—a pencil—and we are going to get students to use it. But the teachers have never had the experience of sharpening it, of feeling the texture; they don't know how to erase with it.

We're the only institution that provides tools for its clients rather than its workers.

Perhaps that is the fundamental problem. I sometimes talk about a way to start with computers that may sound facetious. I suggest issuing a computer to every teacher. Let them take it home. Then offer lots of workshops, lots of instruction, lots of things available, nothing required, anybody in their family can use it, no restrictions, it's theirs.

I'd give them four pieces of software: a good data base program, a good spreadsheet program, LOGO, and a good word processing program—and a printer. And they should get excellent support without judgment; that is, if they don't know how to use their word processor there should be a hot line. Believe me, I could have used that at various points in my odyssey through the computer culture: somebody to call when *Perfect Writer* didn't do what it was supposed to.

If a school does that, it will get teachers who start to do their writing on the computer, because they have it available when they're doing lesson plans, developing tests and handouts, and writing notes to parents. So while they are meeting these needs, they will also be having the experience of word processing. Then, either they or someone in their family may experiment with some kind of personal budget, and they'll learn a little about spreadsheets. And, if they have time they'll begin to play with the LOGO language. Teachers enjoy it; it's fun to learn new

things, and it's interesting to be able to identify with your students and know what it feels like to learn something new. In other words, give teachers a personal experience of what it's about and relax about the rest.

You seem to be implying "Physician: heal thyself"; that we should look to our own experiences with computers before we can do an adequate job with students.

It's not one or the other. Those two turn out to be the same thing, because one of the most interesting things the computer can do is create an identification of the teacher with the student and the student with the teacher. And it's not just because the teachers are novices with the computer; teachers and students learn together because that's the nature of technology. You can be constantly in the process of creating and learning for yourself. That is something that the teacher and the student share as they work with the computer, even when the teacher becomes much more expert. Every new program you write is new to you. The material fosters the kind of joy that comes from creating something. The most educationally interesting effect is that teachers and students are sharing that experience of creation: of being nervous about getting it done, frustrated when they can't get it right, and satisfied when it all works.

That's why I think so little of this whole debate about "should we or shouldn't we teach students programming?" The reason to teach people programming is that you think when you're committed to creating something. For some people that's a very exciting thinking experience—a good thing to have in your education. The effect on the teacher is not very different from the effect on the child.

Some people say that to better understand how to integrate computers into the classroom you must look for their effects on children. Are you saying that we can find some of that information by looking within ourselves?

Absolutely. Teachers who get involved with programming are having an experience of working with a new

expressive medium. Some people don't like the medium and that's okay. Some people don't like clay, because it gets under their fingernails; they would rather work with watercolor. Here too, they are working with an expressive medium and they can therefore identify with the ways in which the kids who are in contact with it will be feeling as they too work with this medium.

And they can identify with what I'm most interested in, which is that different kids are going to use the medium in their own way. When teachers understand that they won't slap the wrists of kids who don't do it the way it is in the books. They may even encourage people to use different styles and different personal learning strategies because good teachers know that's the way their own learning happens.

Two frequently asked questions are should computer literacy be mandated, and should every child be taught to program.

You can put hardware in and enforce educational programs from the top. You can get children to be able to do some elementary programming, and you can get children to know what a programming language is. You can force them to know what a disc drive does. You can force all of this from the top. You can get high school students knowing how to write simple programs in BASIC or PASCAL. But I think that having done that, you haven't done much of great educational consequence.

Ask yourself, what's the computer revolution? And how do you get that to happen? That requires a base of experience, for both adults and children, of a personal medium, a personal learning environment. The reason this has to come from the bottom—call it grass roots if you want—is not because it's a political movement that needs organizing with teacher networks, and so on. It's that this is an expressive medium. You can't teach poetry writing if you've never written a poem; you have to have an experience of it in order to have any kind of sense of what can be done.

I think this experience involves doing some programming—but not be-

"There has been this new rash of surveys reporting that 'kids don't like computers,' or that schools are losing interest in them. That's a little like saying the child doesn't like pencils; the child doesn't like paper."

cause programming is something that you want to test as a competency. It's because programming is an experience that can lead to self-reflection about learning and about your style of learning. The more people know about their style of learning, the better they are at learning a lot of different things. The issue is not "what does programming generalize to?" What's being generalized is not the skill of programming. What potentially could be exciting to have is an experience of yourself as a learner. That makes you both more indulgent of your difficulties and more appreciative of where you have strengths.

I don't want to put down all the "good things" that can be imposed from the top, such as minimum competencies or a sense of participation in the future that can come from feeling comfortable with a computer. But the critics are not wrong in saying "Is this really what education needed? People who can do a little programming?" And the utopians are not wrong in saying, "This is irrelevant to all concerned." Because insofar as I'm a utopian, insofar as I believe that something big can happen here that has very important implications for thinking, feeling, for how children think about themselves, it's just not enough. People don't sense new feelings of empowerment because they can write a program in PASCAL. It's nice, it's

good, I don't want to put it down. But that's not what the revolution is about. But that's not where the potential for seeing something revolutionary happen in education will come from.

If you could develop the ideal plan that would facilitate this revolution, what would it be?

I have a two-pronged "plan" that expresses the principles I've been talking about. First, as I said before, spend your money on giving computers to the teachers. There should be enough computers around the school so that the teachers can use them there, too, if they want to. But more important is that they get to take them home.

The second prong is that while this is happening over perhaps a year or two, computers would be in the classrooms for *word processing* and all the children would use them. That is one of the more important uses of the computer. It's not hard, and many schools find it turns children into writers; you get an *explosion* of writing.

And the children meanwhile are learning how the computer operates and becoming comfortable with it as a tool that they use to empower themselves. It's not a toy; it's a productivity tool; they are using it to get their work done more effectively. And that's the meta-lesson.

Then, depending upon what the resources in the school are, there should be support for children to do programming. The number of children and the degree to which they learn some LOGO should depend upon the individual. With the computers around some children will become programmers—because they will find a manual. I would let the natural process of diffusion happen.

But, as a first pass, I wouldn't do much more. The teachers have the computers, they are doing word processing, they are experimenting at home, they are going to be teaching their own children LOGO, teaching themselves. They are going to be meeting together to share problems and learnings, they will be getting support, and they will start to experiment with the computer, too, to see what they can do. And then after two years I would then be in a position to

think about curriculum and organizational implications.

In summary, according to the "Turkle plan," everybody gets to use the computer as a productivity tool and gets to feel comfortable with the computer as a tool for personal expression.

And because teachers will know from their own experience that everybody does it differently, including themselves and their three kids, they are going to be expecting that and facilitating that among the children. Then when the girls do it differently from the boys, or the "hards" or the "softs," or the people who like transparency or the people who like opacity, teachers aren't going to be alarmed, dismayed, and shocked and tell people how they should really do it "right."

You spend quite a bit of time in schools. What good things do you see?

When I go into the schools and talk to teachers and look at students, I'm not seeing the kind of simple effect that most writers seem to say we should be looking for; I'm seeing consequences. I see teachers who are turned on by teaching something new, and students who are empowered because they can make something happen. I see new kinds of collaboration, too.

I also think that computers, unlike television, increase one's attention span. This is a medium where you can't be passive and you have to be constructive. The computer takes up

"Teachers and students learn together because that's the nature of technology. You can be constantly in the process of creating and learning for yourself."

Blocks and Computers

Parents who want schools to prepare their children to live in a computerized world would like to see computers in every classroom—beginning with kindergarten. Yet some child development specialists who believe that kindergarten children need to use concrete materials to investigate logical principles may recommend keeping computers out of the classroom—at least temporarily. As a kindergarten teacher listening to this debate, I often wondered if I should choose between computers and sand, blocks, and paint.

On Wednesday a computer was delivered to my kindergarten classroom for our use during the day. When I went to load the Concentration game disk, I realized that there was no disk drive. A few minutes later a disk drive was delivered, and the class watched as it was installed inside the computer. Then I loaded the program. For the rest of the day the 21 children played Concentration two at a time. This was their second hands-on experience with the school's computers. By popular request the computer was brought back the following day.

On Friday I went into the block area and noticed that three children had filled some hollow blocks with unit blocks. I suggested that this limited the number of blocks available and that it made clean up difficult. The children agreed that we could empty some of the blocks.

Then they pointed to two constructions. "We can't empty these. They are our computers. You know they have to have stuff inside them." Two large hollow blocks standing on end were filled with an assortment of smaller blocks. In front of the large blocks were "keyboards" constructed of small blocks.

As I looked inside their "computers" another child said, "The computer needs that long thin block in there."

"Why?"

"Because that's the disk that tells the computer what to do."

These children do not have computers at home and their parents do not work with computers. But like every five-year-old they are learning about the world around them and translating it into experiences they can control. I now feel certain that a computer belongs in my kindergarten classroom along with the blocks, sand, and paint.

—By Sue Rasala, kindergarten teacher, Lincoln-Eliot School, 191 Pearl St., Newton, MA 02158.

your attention because there's so much that's being called for, that you have to put in.

I'm also seeing a lot of very interesting uses of the computer as a vehicle for fantasy. It frees up some children to make up extravagant stories that express their fears and their fantasies in a very constructive way. They can get a sense of control and mastery over the fantasies by, in a sense, imbedding them in a movie or in a computer-generated little filmstrip that they feel in control of. This is a very interesting—almost paradoxical—use for technology to free up imagination and feelings.

Another psychological and cognitive effect reported in your book you called the "computer-as-Rorschach": the way computer use can reflect personal style, even to the point of becoming a "Second Self."

I'm still increasing the depth of my understanding of the computer as a personally expressive medium for different styles. When people appropriate the computer in ways that allow the machine to be integrated into their sense of identity, I argue that that's the biggest payoff educationally and personally.

I have been working in a school project in an inner city school involving creation of a very computer-rich environment. We want to understand and document how a variety of teaching and learning styles can be supported using the technology. We hope to show that even the teacher who wants a very structured classroom can have an experience of personal mastery. In this school the teachers involved are not just those who bought open education and *Mind Storms* if not at all. Some of these teachers have very strong feelings about children's need for structure.

And as part of the culture of the classroom, I'm very interested in the way children develop a sense of their own aptitude and inaptitude, and the ways they label themselves. This means I've become increasingly interested in the issues of gender and equity. My observations since the book reinforce the idea that different people use the computer in different ways, and the issue of *control* is central. Not surprisingly, in our culture girls tend to have different control orientations than do boys. The approach of, "*Ab, a machine: let me dominate it!*" is not something that many young women share in quite that form.

In my book I noted the fallacy that programming is only one kind of thing. I had observed two styles I termed "hard mastery" and "soft mastery." The hard masters, usually boys, use conventional programming style. They make a plan, conceive the program as a whole, and then break it up into manageable pieces. It's a top-down, "divide and conquer" strategy. The "soft masters," in contrast, have a style in which the aesthetics of the graphics are more important than the elegance of the plan. It is a style in which things are not done by advance planning, but through negotiation and experimentation with the machine. While hard mastery is the imposition of will over machine, soft mastery is more interactive: try this, wait for a response, try something else, let the overall shape emerge from an interaction with the medium—more like a conversation than a monologue.

You enter into more of a negotiation than a relationship that puts one priority on your imposition of will over the program. It sometimes can look like fooling around, but it's not. When the reports come in from the schools around the country saying that girls are getting so turned off, it's because computer mastery is being defined as a certain thing, and a lot of girls don't like that thing. They need to be allowed to do it their way.

That's why I'm interested in the inequity that comes from not appreciating the computer as an expressive medium and not allowing different styles to flourish. I know empirically that not creating environments where

“The reason this has to come from the bottom—call it grass roots if you want—is . . . that this is an expressive medium. You can’t teach poetry writing if you’ve never written a poem.”

there is permission for different styles to be legitimate leads to discrimination against women.

There’s a personal story I sometimes tell when I don’t think people know how to program and I want to explain these styles of mastery. I am a product of two-culture education—in France and in the United States. In a French composition class, the teacher allowed a week to do the outline and two weeks to do the composition—and in that order. I couldn’t do that because I write my composition by having it all over the floor and doing parts out of order and moving things around. So I wrote the composition my way by moving things around on the floor, and then I wrote the outline based on that composition—all this in the first week. I handed in the outline and then sat on the paper and explored Paris for two weeks while all the others were doing their compositions.

The point of the story is not that I was so very smart, but that I was basically fighting against an educational system designed to weed out people like me. The result was that I thought I couldn’t write; I thought what I did was bad. It had a profound effect on me, and the idea that I have written two books boggles my mind because it

was so deeply ingrained in me that I didn’t do writing right. Not only did I have a sense of incompetence; I had a sense of fraudulency for what in fact was my way—not just of writing but of thinking, of getting ideas together.

And the reason I always tell the story when talking with people who don’t program is that I’m arguing the same thing for programming. If you slap the wrists of the ones who work “associatively,” you’re telling them that they don’t do it right—and you’re keeping them out of the computer culture.

In *The Second Self* you noted what might be called “metaphysical” implications of the computer for children. Have you followed up on this?

I’m continuing to pursue the effect of the computer on the child’s concept of what is life and what is not life. One of the most interesting things that came out of my work is the way in which involvement with the computer causes kids to think through what is special about being a person. Because the computer sits on the boundary between a living thing and a not living thing, this causes the child to reflect on the difference between “thing” and “person”: what is special about being a person? Children always used to define the specialness of people in relationship to people’s nearest neighbors. When the nearest neighbors were the animals—the pet dogs and cats, or the horses—they believed the animals had desires, but the people were special because they had reason. So in a certain sense the Aristotelian definition of man as a rational animal was good even for the youngest child. Today, however, if computers are the nearest neighbors, children believe people are special because they *feel*. Computers “think,” but people “feel.”

I have documentation now that this new way of thinking about people is not a transitional first computer generation phenomenon. Because I am now able to look at children who grew up assuming that a computer was something you naturally had in your house, I’ve been able to be a lot more confident that the effects I found were not transitory.

In fact, I’ve been very interested in what’s happened in children’s toys and

games in these intervening years. Look at television on Saturday and Sunday morning and see the toys the kids are playing with: toys that walk that same edge between what is a machine and what is a person—these Transformer toys, the Gobots, the cars that turn into robots that turn into people, or people that can transform themselves into robots. Children are involved more and more in their child culture with objects that play through some of the issues I wrote about in the book.

Do you see any connection between what you’ve just said about children seeing humans as emotional beings and the increasing number of social observers such as Peters and Waterman, Benis, and Goodlad who perceive a “new paradigm,” one within which people’s “feelings” become an important factor in institutional management?

You know, people smile about the “high tech—high touch” metaphor, but it’s an extremely good way to capture what I’m saying. One of the things that high technology does is confront us with the fact that *that’s* really not enough to fulfill most of our human purposes. It makes us think about what is special about being human and therefore what is special about human organizations. It makes us see that life is a lot more than pressing buttons.

I’m not sure that what I’m talking about with children is the same issue, but there certainly is a similar paradoxical effect. I remember one child who talked with me about robots: “I guess they’ll do a lot of the jobs but I guess people will still have some things to do. I think they’ll cook the food, have the babies, be in families, love each other. I guess they’ll be the only ones who will go to church.” In other words, faced with the world of the robots, the child is thrown back on what in fact is special about being a person.

Copyright © 1986 by Sherry Turkle

Lewis A. Rhodes is manager, Media and Technology, Association for Supervision and Curriculum Development, 125 N. West St., Alexandria, VA 22314.

Copyright © 1986 by the Association for Supervision and Curriculum Development. All rights reserved.