CHAPTER 7

Control and Power in Educational Computing[1]

PETER H. KAHN, JR. and
BATYA FRIEDMAN

> The [high school] computer center is overseen by a student group, the Computer Users Society. Members have keys to the room, and may gain access at any time, day or night, weekends. One Sunday morning, Harvey [the high school teacher] recalls, he came in around 10 a.m. thinking he would get some work done while the place was quiet. He could not find a free terminal.
>
> —Dormer, 1981

> The Computer Center User Society . . . holds regular meetings to decide the policies and rules of the computer program. Managing the computer system are six students, called "superusers," who oversee the use of the computers. . . . [One student] explained that although superusers have access to everything, by the time a student gets to the point of being a superuser, immoral actions such as looking at grades are "the last thing we'd have to worry about. This place is so special . . . it would be such a downfall that no one would want to risk it."
>
> —Appel, 1985

These educational computing practices formed part of the computer education that Harvey (1980, 1983) initiated in an otherwise traditional high school. In this paper, we want to say more about the direction in which Harvey points. It is toward student participation, cooperation, and interest, and a school culture that is imbued with using computer technology to foster democratization.

More broadly, we seek to convey an account of educational computing based on the primacy of human agency. Toward this end, we distinguish between control and power. In our use of these terms, control occurs when people direct coercively another person, or direct the action or process of a non-person (e.g., a machine). The key here is that within social relations control is usually unethical because it undermines others' autonomy. Power, in turn, occurs only in social relations. It can be coercive, and in such cases power refers to how people control others. But power can also refer to non-coercive relationships within which people influence, organize, and lead others. After developing this distinction, we take up several questions: How do we teach students that humans, and often the students themselves, control computer technology and are responsible for the consequences of computer-mediated action? How do we teach future computer scientists to design systems that foster such understandings? And how do we teach students that by controlling computer technology—and choosing wisely—people have power to effect meaningful and ethical change in educational settings and beyond?

Computers: Objects or Others?

Consider this brief event: A friend of ours sought to make an airline reservation and called the airline company to charge her ticket. In the process of writing the ticket, the reservationist requested a home phone number. Now, it happened that our friend was between academic jobs and in transit for the summer. Thus she did not have a home phone. Although the reservationist listened sympathetically to this explanation, he insisted that he be given a phone number: "Our computer program has to have a phone number or it won't work. I can't even get to the next screen. It's the computer's fault." How many times have we heard that? "It's the computer's fault." Well is it? Can computers be at fault?

Such questions arise because computer technology often appears volitional and intelligent. Computer systems, for example, can "track" credit histories and "decide" to reject loan applications. Computer-guided missiles, as noted by Dawkins (1976), often appear to "search actively" for their target, to "predict" and "anticipate" the target's evasive moves. Medical expert systems "diagnose" illness and "recommend" cures. In terms of educational software, intelligent tutors "decide" which problems a student will work on, "correct" students when they are wrong, and "judge" when a student is ready for more advanced problem sets. Tracking. Deciding. Searching. Anticipating. Recommending. Tutoring. Correcting. Judging. Such terms would seem to imply that we believe computers are not so different from humans and that they have, to varying degrees, intentional states such as thoughts, desires, consciousness, free will, and the capability to make intelligent decisions. How did we come to believe such things about computers?

One response is that we do not believe such things. Another response is that we believe such things because they are true. Both responses are worth our attention. In the first, it might be granted that in conversations people readily talk about "intelligent tutors" or "smart missiles." Or, like the airline reservationist noted earlier or a student who hands in a late term paper, one might say "it is the computer's fault." But, it could be argued, what appears as sincere instantiations of attributing agency to computational systems are nothing more than superficial verbal responses, and that people do not really conceive of computers as human or humanlike.

Research bears on this first response. Turkle (1984), for example, interviewed children about their experiences with an interactive computer game called Merlin that plays Tic-Tac-Toe. Her findings indicated that children attributed psychological characteristics to Merlin: that Merlin, for example, was capable of cheating. Similar findings appear with adults (cf. Rumelhart and Norman, 1981; Weizenbaum, 1976). Kiesler and Sproull, for example, have shown that when the interface presents the computer as a persona, users engage in cooperative behavior with the computer (Kiesler and Sproull, in press) and attribute some personality traits to the computer (Sproull, Subramani, Kiesler, Walker, and Waters, 1996). Along similar lines, Nass, Steuer, Tauber, and Reeder (1993) tested computer-literate college students using different types of computer-based tutoring and testing systems. Their findings showed that when a computer mimics even a minimal set of characteristics associated with humans (such as simple textual cues) that users attributed to the computer human-like qualities, such as friendliness. More recently, based on their research, Nass, Moon, Morkes, Kim, and Fogg (in press) have suggested that people respond positively to flattery from a computer, are attracted to dominant or submissive programmed characteristics embedded in the interface of a computer, and, most generally, treat computers as social actors.

Another study by Friedman and Millet (1995) employed social cognitive methods to assess individuals' reasoning about computer agency and computer capabilities, and individuals' judgments of moral responsibility in two situations that involved delegation of decision making to a complex computer system. One situation involved an automated computer system that evaluates the employability of job seekers and rejects a qualified worker. The second situation involved an automated computer system that administers medical radiation treatment and over-radiates a cancer patient. Computer science majors from a research university were interviewed. Results showed a complex pattern of results. On the one hand, most of the students attributed either decision making or intentions to computers, and over one-fifth of the students explicitly held computers morally responsible for error. On the other hand, most of the students judged computer decision making and computer intentions to be different from that of humans. Reasoning focused, for example, on rule-based systems (e.g., "[the computer] can decide in a sense that somebody has programmed rules, which it follows, and in that sense it chooses a course") and algorithmic processes (e.g., "[the computer] is deciding based on a very clear strict algorithm . . . it's a decision but not an open-ended one"). Moreover, virtually all (96 percent) of the students' reasons for blaming the computer referred to the computer's participation in the sequence of events that led to harm. In contrast, most (80 percent) of the students' reasons for blaming people (such as the system designers, human operator, or administrators) referred to failing to meet some commonly expected reasonable level of performance, including appeals to inadequate design of the computer system, inadequate testing or debugging of the program, inadequate monitoring, and personal carelessness. Thus, while this study does not support the stronger claims by Nass and his colleagues that people treat computers as social actors, it does suggest that in certain respects people attribute agency to computer technology, and that such attributions go well beyond superficial use of language.

If people attribute agency to a computer, it could be argued—and this is the second response—that this attribution occurs quite simply because computers have agency. Historically, the military has been committed to this view philosophically and financially. The military has sought to develop "smart" and "autonomous" computer-based weaponry and to use computers to train military personnel. In turn, the military has had an enormous impact on shaping the fields of computer science and educational computing. For example, according to Thomborson (cited in Noble, 1991, p. 13) 70 percent of all academic research in computer science has been funded by the Department of Defense. Similarly, according to Noble (1991), "military agencies have provided three-fourths of all funding for educational technology research over the last three decades, and within government agencies, the military spends seven dollars for every civilian dollar spent on educational technology research" (p. 2). Thus, most of the research in computer science and most of our nation's computer-based educational applications comes out of a tradition that believes that computer technology can mimic, if not duplicate, human agency. Is such a belief warranted?

Perhaps the most sustained and incisive critique of the idea that computers can be intelligent in the sense of having intentional states has been advanced by Searle (1980, 1984, 1990, 1992). While this is not the place to go in depth into Searle's thinking, it is worthwhile to review the skeleton of his well-known Chinese room argument. Through it, we stake out our philosophical position toward technology, which, in turn, provides a cornerstone for our educational practice.

Searle's thought experiment goes something like this: Imagine that you do not understand Chinese writing or speaking, and you are put in a room with a basketful of Chinese symbols. You are then given a rule book in English that tells you how to match up certain symbols with other symbols, based only on their visual configuration. For example, a double-bent squiggle might get matched with a half-bent squoggle. Now, imagine that a Chinese person from outside the room writes you a question to try to determine if you know Chinese. You receive the written question, which to you are only meaningless Chinese symbols, and according to the rules of your English rule book, you match up the symbols with other appropriate symbols, and then send out the response. With a good rule book, the Chinese person will not be able to tell the difference between your answers and the answers given by a Chinese-speaking person in that room. The outside questioner might ask with Chinese squiggles, "Would you like a Big Mac or a Whopper?" You might respond with Chinese squoggles, "A Whopper, for I find Big Macs just a tad too greasy." But now we ask the question, would you understand what you just said? Or, more generally, when you correctly manipulate the symbols, do you *understand* Chinese? Searle argues, absolutely not. After all, when you send out an answer, you do not know what you are sending out. All you are doing is matching symbols with other symbols based on the rule book. That is, there is nothing in the formal symbol manipulation (the syntax) that provides the understanding of Chinese (the semantics). So, too, with computers. Because computational systems are purely formal (syntax), and because purely formal systems have no means to generate semantics, Searle argues that computational systems do not have the properties that are central to human agency.

Searle's position has generated a great deal of debate, starting with twenty-six rebuttals in the journal where his initial article first appeared (see Searle, 1980). For example, one rebuttal (often referred to as "the systems reply") to Searle goes like this: "Searle has confused levels of mental organization. The rule-follower in the Chinese room represents a piece of the human brain, and the entire room represents the person. Thus, while it is true that the rule-follower in the Chinese room does not understand Chinese, the entire room does." In Searle's (1981) reply, he asks that we go ahead and assume that the person in the Chinese room internalizes all of the elements of the room: the rule book, the data banks of Chinese symbols, everything. The person can then do the calculations in a location outside the room. All the same, the person understands nothing of Chinese. Thus neither does the room, because there is nothing in the room (the system) that is not in the person.

In our view and that of others, Searle has defended himself well against his critics. This is not to say that humanlike agency might not someday be realized in material or structures other than biological brains. It is to say that computers as we can conceive of them today are not such materials or structures. Thus while humans may (sometimes with valid justification) delegate decision making to computer technology, fundamentally it is humans who control technology.

In the last handful of years, some members of the Artificial Intelligence (AI) community appear to have backed off from their strong philosophical claims (see, for example, the editorial by Chandrasekaran, 1992, editor-in-chief of the artificial intelligence journal of the IEEE). Their current position goes something like this: Regardless of whether or not computers actually have intelligence, computers act as if they do, and thus we can rightly design and interact with them as if they do. On the one hand these AI researchers give up a good deal of philosophical ground to those like Searle. On the other hand, they keep their research agenda fully in place—to build intelligent-like computer systems. But if computers continue to be designed as if they have agency, and if in their interactions with computers people act as if computers have agency, then it seems likely that people will think so as well. Thought and action cannot be so easily compartmentalized. Thus we have argued elsewhere (Friedman and Kahn, 1992) that increasingly sophisticated computer systems should be designed not to mimic human agency, but to support it. For example, computer interfaces should not be designed to intercede in the guise of another "agent" between human users and the computational system, but to "disappear" such that the user is freed to attend directly to, and take responsibility for, the tasks at hand (see, also, Shneiderman, 1987; Winograd and Flores, 1986). The same holds for the emerging work on software agents.

Here is where we have been heading. We suggest that children and adults in various ways mistakenly attribute agency to computer technology. And we say mistakenly because it is our claim that computers as we can conceive of them today in materials and structures do not have agency. Moreover, we suggest that people should control technology, take responsibility for the consequences of computer-mediated action, and minimize if not eliminate control within relations that involve power between people.

To develop these last points a little more fully, consider the following example. Drawing on Arnstine (1973), imagine the hijacker who points a gun at an airline pilot and says, "Fly me to Havana or I'll blow your brains out." The gun could be said to coerce the pilot. After all, without the gun the threat would have little force. However, contrary to some theorists who argue that objects can have a social life (Appadurai, 1988), we maintain that the gun is fundamentally an inanimate tool mediated by a person (see also Scheffler, 1991, chap. 8): in this case a tool used for the purpose to establish control over another person. In this explicit psychological sense, we agree with the statement, "guns don't kill people, people do." It is also the case, however, that once people build tools, the tools have built-in features such that people usually use the tools in certain ways, toward certain ends (cf. Bromley, 1992). Thus with a gun, or with a computer-driven weapon, people often exert power to control other people.

During the Persian Gulf War, United States Patriot missiles were used to detect and intercept Iraqi Scuds. However, the precision of a Patriot's calculations to predict the location of a Scud while in flight depended in complex ways on the continuous running time of the Patriot's computer system. The longer the system was running, the greater the imprecision. This systematic software error resulted in the death of at least 28 Americans in Dhahran (US General Accounting Office, 1992). In such situations, it is easy to say, "it's the computer's fault," not unlike what the airline receptionist said whom we mentioned earlier. We have argued that it is not. Rather, the responsibility for harm incurred through computer-mediated action lies ultimately with people. Depending on the specific situation, responsibility will perhaps rest with one or more of the following: the programmers who wrote the software, the administrators or other individuals who chose to implement the particular hardware or software, the designers or manufacturers of the hardware, or the operators of the computer technology.

Constructivist Education

In terms of educational computing, our distinction between control and power motivates several questions. How do we teach students that humans, and often the students themselves, control computer technology and are responsible for the consequences of computer-mediated action? How do we teach future computer scientists to design systems that foster such understandings? Finally, how do we teach students that by controlling computer technology, people have power to effect meaningful and ethical change in educational settings and beyond? Our answers build on a constructivist account of education. Thus, before highlighting specific computing activities, we should like to say a few words about this perspective by elaborating on four constructivist principles proposed and practiced by DeVries and her colleagues (DeVries, 1988; DeVries and Kohlberg, 1990; DeVries and Zan, 1994).

From Instruction to Construction

Many people believe that for students to learn, teachers must instruct, by which it is meant that learning depends on a teacher who correctly sequences curriculum content, drills students on correct performance, corrects mistakes, and then tests for achievement. Granted, one might note a few sidewise embraces of critical thinking and cognition. But if push comes to shove—if, for example, test scores go down—the call is clear. Back to basics. Instruction in the Three Rs. In contrast, in the move from instruction to construction, learning involves neither simply the replacement of one view (the incorrect one) with another (the presumed correct one) nor simply the stacking, like building blocks, of new knowledge on top of old knowledge, but rather transformations of knowledge. Transformations, in turn, occur not through the child's passivity, but through active, original thinking. As Baldwin (1897/1973) says, a child's knowledge "at each new plane is also a real invention. . . . He makes it; he gets it for himself by his own action; he achieves, invents it" (p. 106).

Think of it this way. On a daily level, children encounter problems of all sorts: logical, mathematical, physical, social, ethical. Problems require solutions. The disequilibrated state is not a comfortable one. Thus the child strives toward a more comprehensive, more adequate, means of resolving problems, of synthesizing disparate ideas, of making sense of the world. Constructivist education, therefore, centrally involves experimentation and problem solving, and student confusion and mistakes are not antithetical to learning, but a basis for it.

From Reinforcement to Interest

Traditional educators often seek to shape student behavior through four types of reinforcement procedures: positive reinforcement, punishment, response cost, negative reinforcement (Rohwer, Rohwer, and B-Howe, 1980). Each procedure builds on a conception that children learn through stimulus-response conditioning, and that for effective instruction the teacher needs to strengthen, weaken, extinguish, or maintain learned behaviors through such reinforcement procedures. In contrast, as we have argued, humans have not just syntax, which leads to a conception of the child as a computer, to be programmed, but semantics, meaning. It is not surprising, then, that children construct meaning more fully when engaged with problems and issues that they find meaningful, that captivate their interest. Thus, from a constructivist perspective, teachers find out what interests their students and build curriculum to support and extend those interests. They allow students to help shape the curriculum and the freedom to explore, to take risks, to make mistakes. Indeed, it can be argued that many of the behavior problems that traditional teachers try so hard to suppress arise precisely because students find the curriculum drudgery.

From Obedience to Autonomy

Construction and interest do not thrive in an environment in which the teacher is the authority demanding obedience. Moreover, obedience leads to conformity and to the acceptance of ideas without understanding. Thus, from a constructivist perspective, teachers should move away from demanding obedience and toward fostering the child's autonomy. Now, by autonomy we mean in part something like independence from others. For it is only through being an independent thinker and actor that a person can refrain from being unduly influenced by others (for example, by neo-Nazis, youth gangs, political movements, and advertising). Autonomy in this sense is necessary for an individual to control technology. But by autonomy we do not mean a divisive individualism, as constructivist autonomy is often said to be (Hogan, 1975; Shweder, 1986). Rather, within a constructivist framework (Baldwin, 1897/1973; Kohlberg, 1969, 1984), autonomy is highly social, developed through reciprocal interactions on a microgenetic level, and evidenced structurally in incorporating and coordinating considerations of self, others, and society. In other words, the social constrains or bounds the individual(ism), and vice-versa.

From Coercion to Cooperation

In some sense, the movement from coercion to cooperation reflects the flip side of obedience to autonomy, but more from the student's and not the teacher's standpoint. As does autonomy, cooperation entails incorporating and coordinating one's own feelings, values, and perspectives with those of others. Given that adults' relationships with children are laden (often necessarily so) with coercive interactions, peer relationships are centrally important. Through them, concepts of equality, justice, and democracy flourish (Piaget, 1932/1969), and academic learning is advanced (Vygotsky, 1978).

These four constructivist educational principles comprise part of a larger social-cognitive research tradition (e.g., Arsenio and Lover, 1995; Damon, 1977; Friedman, 1997; Kahn, 1992; Kahn and Friedman, 1995; Kohlberg, 1969; Laupa, 1991; Nucci, 1996; Piaget, 1932/1969; Selman, 1980; Smetana, 1995; Turiel, 1997; Wainryb, 1995). These principles are also, in many respects, compatible with other educational theories, for example, those that are progressive (Dewey, 1916/1966) and experiential (Wigginton, 1986). We prefer, however, the constructivist label for two reasons. First, it highlights that no matter how social the discourse, elaborate the organizational structure, or co-constructive the reciprocal interaction, fundamentally it is a self (a morally responsible agent) who makes meaning in the world. Second, constructivism allows best for the modern as opposed to postmodern epistemology which guides our work and which provides a foundational basis for checking abuses of power (Kahn, 1991, 1993, 1995; Lourenco, 1996).

Constructivist Computing in the Classroom

Educational computing can be enhanced by software designs which shift more of the control over technology, and responsibility for learning, back on the student. Historically, Papert's development of the LOGO microworld is perhaps most well known along these lines. Through programming computers with turtle geometry, Papert (1980) showed how a child "both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building" (p. 5). More recently, this orientation toward the technology and learning is embedded in computer construction kits. *Interactive Physics II* (Baszucki, 1992), for example, allows the user to access a large number of physical components (springs, ropes, blocks, and disks) and properties (forces and gravity) to model and investigate two-dimensional physical systems. Through such construction kits, the user engages actively in a design, receives immediate data by which to judge its success, and then is positioned to rework a better design, and thus continue the generative process.

Educational software can also be designed to allow students to explore hypothetical social situations (not just mathematical or physical) from multiple perspectives, and to follow different courses of action of each. For example, in the computer simulation *Our Town Meeting* (Snyder 1985a), students working in groups are assigned to one of three town agencies, each with its own agenda. Through the simulation, students find that cooperative strategies often provide the most effective means for resolving disputes and achieving agency-specific goals. Cooperation is also encouraged in another computer simulation, *The Other Side* (Snyder, 1985b), where students work in teams to build the economy of their own country while working toward building a bridge between their own and a rival country.

Such software moves educational computing in a good direction. However, we have seen many cases where "traditional" teachers use such software while remaining largely unaware of how they constrain the software's use. For example, Friedman (1988) conducted an ethnographic evaluation of a computer education program in an elementary school known for successful innovation. Friedman found that fourth and fifth graders were taught graphics programming largely by rote: "After two years of instruction, students still

primarily drew pre-determined images on the computer screen using a single well-drilled algorithm" (p. 4). Teachers seldom asked or even allowed students to seek novel solutions or define their own problems, although the technology allowed for such possibilities.

One compelling way teachers can support such possibilities is to have students use computer technologies as tools to generate, share, and argue about ideas. Consider, for example, a student-run newspaper. Word processors and desktop publishing tools support students' writing. Spelling checkers help the presentation. Spreadsheets can help students to manage the financial aspects of running the newspaper, and databases to organize and access relevant information. Thus, students learn not only how to use a wide range of computer applications but that, as students, they control computer technology and are responsible for the consequences of their computer-mediated action: namely, for their published ideas.

Electronic mail and more generally accessing the World Wide Web has of course been providing many classes with an even wider community for discourse (cf. Cummins, 1988; Horowitz, 1984; Martinez, 1993). Not uncommonly, for example, classes conduct science experiments in consort with other classes across geographical locations, using electronic mail to share their data and interpretations. Other schools have combined efforts across geographical boundaries to write a joint newspaper. Educational networks like Apple Global Education support such computing practices, and more (see, e.g., Kearsley, Hunter, and Furlong, 1992, chapters 5 and 8). In ways like these, computer technology can be used to enhance communication and understanding between students who might differ on the basis of geography, culture, race, age, and economic standing.

Students can also be engaged with computing issues which emerge within organizations. Here, for example, is an idea for a sustained project with older students. Have students choose an organization—McDonald's, a bank, a manufacturing plant, the school attendance office, most any will do—and have them research how the computer technology was brought into the organization, how it supports the organization's goals, and how it shapes social interactions. Students can collect "data" through observations, surveys, and extended interviews. Their data can inform on such questions as: Who decided that computer technology would be good to have in the organization? Who decided on the specific hardware and software? To what extent did workers and consumers participate in this process? What do managers and workers think about the technology's current use? Were there any unanticipated consequences from using the technology?

Such projects were successfully conducted by undergraduate students at our own institution. One group of students, for example, in an administrative science course, elected to study computer use in a small accounting firm. The firm used a computerized tax preparation program chosen for its ease of use and compatibility with existing hardware. In addition to these features (and unnoticed by the firm members at the time of purchase), the software kept a running tally of the number of tax forms completed by each accountant. This information on each accountant was revealed at the end of the tax season. Office discord followed. In response, the firm made a collective decision to "hide" this unsolicited information in subsequent years. Through reflecting on findings like these, students can identify situations where workers feel either controlled or supported by technology, and can be better positioned to propose changes for the work place.

Who Controls the Computer?—Democratic Communities

Alexander Randall has told of his experience walking the streets of places like Warsaw, Moscow, and Kiev during their recent struggles for independence (Lewis, 1992). Randall talked with people in various organizations, with dissidents, and with mainstream journalists. He asked, "What is the next step to make sure that freedom of speech and free government survive?" Their answers surprised him. They all said they wanted desktop publishing equipment. It dawned on him. "What is freedom of the press if you don't have the press?" For such reasons, Randall founded the East-West Foundation. This nonprofit organization solicits used (and by many standards outdated) personal computers from Westerners and provides them free of charge to dissidents and journalists in the Eastern European countries. In Randall's words, "Our old XT's and AT's and Mac 512's are awesome tools to people who don't have them. . . . The fall of Communism did not axiomatically mean the rise of democracy. This is our people's response to insure that the forces trying to make democracy happen will ultimately succeed" (p. F12). In such ways, people can control computer technology to enhance political freedom.

If we accept, moreover, that children develop understandings about political freedom and of how to create societies in which such freedoms thrive, then it follows that such understandings and practice need to be made an integral part of children's education. Toward this end, we suggest that schools be organized to increase student self-governance (e.g., Kohlberg, 1980, 1984, 1985; Power, Higgins, and Kohlberg, 1989). In such schools, students determine many of the policies that regulate their own classroom and school activities, and thus gain experience with democratic and consensus decision-making processes. The teacher's primary role is to guide the process. Teachers may help students become aware of and understand issues when they arise, and ensure that all students have an opportunity to voice their views in policy-setting meetings. Since teachers are themselves members of the self-governing community, they too may act as advocates, influencing the selection of issues for which students set policy. Teachers may also need to limit the range of acceptable agreements, particularly when majority agreements conflict with minority rights and welfare, or when community decisions conflict with legal requirements on the societal level.

Such "authoritarian" interventions in self-governing classrooms and schools occur less frequently than perhaps would be expected. Recall, for instance, the educational practices quoted at the outset of this paper that Harvey initiated in his high school computer center. Students held regular meetings to decide their policies and rules. By this means, students delegated "superusers" who oversaw the use of the computer systems and had access to most all of the files (including grades). One student "explained that although superusers have access to everything, by the time a student gets to the point of being a superuser, immoral actions such as looking at grades are 'the last thing we'd have to worry about. This place is so special . . . it would be such a downfall that no one would want to risk it'" (Appel, 1985). In this situation, one would perhaps like to see more consideration for the privacy of others (vis-a-vis grades), and the psychological harm that can occur when such privacy is violated. Yet what is admirable with Harvey's students is that they bounded their own actions ethically by recognizing that harmful repercussions would otherwise fall on their community as a whole.

Friedman (1986, 1991) provides further suggestions for how self-governance can be implemented with somewhat younger students. For example, in one situation Friedman (1986) taught eight- to eleven-year-olds in a classroom in which each student did not have his or her own computer. Thus, along with most teachers nationwide, Friedman faced a common problem: how should the computer resources be allocated? Instead of dictating a solution, Friedman turned the "problem" itself into an educational experience. She divided the students into small groups of four or five and asked them to come up with a solution by consensus. In addition, she imposed the requirement that any solution could not prevent a student from completing class work. Initial proposals ranged from flipping a coin, to relay races, to "those who want to share can share," to "people who shared yesterday don't have to share today." The class then discussed the proposals and voted to implement one. Not surprisingly, students often implemented many imperfect or less than fair solutions. Thus Friedman further guided the process. She asked that the class live with the implemented solution for a week, at which time the policy would be assessed and the process repeated. Accordingly, no policy was binding for an unbearable period of time. Moreover, students had rich opportunities to increase their understandings of resource allocation and participatory democracy.

True, even in such "self-governing" classrooms, teachers have power. But here is a crucial point. Power can be used well—wisely, ethically, not to control students, and not to undermine their autonomy but to promote it. Moreover, as we noted earlier, by autonomy we do not mean a divisive individualism but a commitment to cooperative interactions that coordinate considerations of self, others, and society. Thus, for example, Friedman guided her students through a consensus-building process and allowed students the freedom to make mistakes. Similarly, Harvey allowed his students the freedom to choose which programming languages they wanted to learn and, indeed, the freedom to learn no programming language at all. Harvey (1980) writes that educational freedom means first of all that students can make significant choices, not trivial ones invented by a teacher. Self-governance allows for such significance and thus embraces an educational freedom that seems to us but part and parcel of political freedom.

Educating Computer Scientists

If it is important to teach students across ages and academic fields about the social implications of computer technology, then all the more so for computer science students. These are the students who in the coming years will design and build the computational systems that shape our computing environments. Teaching such students about the social responsibilities that come with their technical expertise has received increasing attention in the literature (Bynum, Maner, and Fodor, 1992; Denning, 1992; Friedman and Kahn, 1992; Johnson and Nissenbaum, 1995; Kling, 1996; Ermann, Williams, and Gutierrez, 1990; Friedman and Winograd, 1990; Gotterbarn, 1992; Johnson, 1985; Miller, 1988; Parker, Swope, and Baker, 1990; Perrolle, 1987; Winograd, 1992). It has also become part of the core curriculum recommended by leading computing organizations ("A Summary of the ACM/IEEE," 1991) and by which computer science departments are judged for national accreditation. Thus we wish to describe several means to integrate social concerns into students' computer design experiences, such that when students define and implement computer systems, standard issues include not only technical ones (such as "What data structure should I use?"), but social ones (such as "How does my system impact the intended users?").

Recent analyses of bias in computer system designs lead toward one such means. Friedman and Nissenbaum (1996) have identified three overarching categories of bias (systematic unfairness to individuals or groups): preexisting social bias, technical bias, and emergent social bias. Preexisting social bias has its roots in social institutions, practices, and attitudes. It occurs when computer systems embody biases that exist independently of, and usually prior to, the creation of the software. For instance, a legal "expert" system was written to offer advice to immigrants seeking citizenship in Britain. Some have argued (Berlins and Hodges, 1981), however, that the British immigration laws are themselves biased against certain nationalities and people of color. To the extent that they are, such biases also became embedded in the expert system. In contrast to preexisting social bias, technical bias occurs in the resolution of technical design problems that often arise due to limitations of the programming tools or algorithm. For instance, in the above legal expert system, the programming language, Prolog, sometimes went into infinite loops and thus failed to prove theorems that were logically implied by the axioms. Due to this technical limitation of the programming language, it follows that the system systematically would fail to identify individuals who were otherwise entitled to British citizenship. Finally, emergent social bias emerges in the context of the computer system's use, often when societal knowledge or cultural values change or the system is used with a different population. For instance, since the early 1970s, the computerized National Resident Medical Match Program has placed most medical students in their first jobs. In the system's original design, it was assumed that only one individual in a family would be looking for a residency, and the system thereby inadvertently discriminated against couples. At the time, such an assumption was perhaps not out of line since there were few women residents. But as women have increasingly made their way into the medical profession, marriages between residents are now not uncommon. Such bias against couples only emerged when the social conditions changed.

These three categories of bias can be used pedagogically within computer science courses. In a course on data structures, for example, students were asked to design and implement a computer dating program. The technical material focused on the use of records and linked lists. The issue of bias arose when students determined issues like who would be included in the database and how individuals in the database would be searched. Some students, for example, assumed only heterosexual users, and their programs were critiqued by other students on the basis that the program resulted in the unfair exclusion of homosexuals, due either to oversight or to the programmer's preexisting bias against homosexuals. Other programs searched for matches with a first-entered first-searched strategy, and thus unfairly favored those individuals who joined the database earlier over those who joined later. This instance of technical bias commonly arose because students used a linear linked list. To remove the bias, some students redesigned the program with a circular linked list. Our point here is that the social import of system design, and the designer's role and responsibility for the design, can emerge compellingly from students' own design experience.

Moreover, the social dimensions of computing become more sensible to students when they design computer software for use by real people in real settings. Such design makes salient the need to take seriously debugging, prototyping, field testing, interface issues, and user satisfaction. Based on our experience, students have successfully pursued a variety of design projects. For example, one student wrote a customized program for helping a baseball coach keep track of his team statistics. Another student built an interactive videodisk of selected artwork and performances for an art history teacher. Yet another student designed a program for the college's radio station that would identify the daily records played and the station's top ten hits. It is also often desirable for students to work on such projects collaboratively. In this context, which is akin to that of a business environment, students must decide what program modules to create, who will take major responsibility for each module, how to fit the modules together, and, once together, how to remove the errors. Thus, to be successful, students must be not only capable intellectually and able to assert themselves on intellectual ground, but able to discuss, cooperate, and build with others.

We are well aware that educational computing can be hindered by the coercive power wielded by organizations beyond the immediate scope of any single individual or school. When this occurs, other organizations need to respond; and though it is beyond our purposes here to say much about how this response can best be organized, we do suggest that effective response can build on our distinction between control and power. Here is an example. In choosing the standard by which to measure academic computing achievement among high school students, the Educational Testing Service (ETS) selected a single programming language, Pascal, for the Advanced Placement Exam in Computer Science. (Note: ETS says that it expects to change to C++ by 1998, or later.) There is undoubtedly an interesting story about why ETS chose specifically Pascal. But what we find especially remarkable is that ETS chose only one programming language. After all, in the field of computer science there is no standard programming language. For example, on the college level, computer science departments in their introductory courses often choose C, C++, Scheme, Modula-2, or Java instead of Pascal. In turn, in advanced high school computer study, teachers and students with interest in artificial intelligence would find most useful the languages of LOGO or LISP, while those interested in operating systems with access to UNIX would find most useful C. Thus why only Pascal on the Advanced Placement Exam? We suspect that part of the answer hinges on the increased costs that arise from creating and administering exams in multiple languages. But, whatever the reason, the end is the same. ETS uses its power to coerce high schools into teaching Pascal (and in the next decade C++), and they thereby unreasonably dictate what counts as advanced computing at the high school level.

How might the educational community address this problem? Minimally, those in secondary and higher education need to discuss these problems with ETS, such that a choice of programming languages is provided on the Advanced Placement Exam. But perhaps this process, even if it is successful, misses the larger point, for it still compels high schools to teach to standardized tests if their students are to be competitive in their college applications. Another solution is possible, which would also solve the problem that arises if ETS is unable or unwilling to add additional programming languages to the Advanced Placement Exam. Namely, in their admissions process, colleges and universities can go beyond standardized testing to assess the depth, scope, and integrity of student's high school computer science education. In this way, standards still matter, as they should. Excellence counts. Indeed, as exemplified in Harvey's computer center, excellence thrives because high school educators and students gain increased freedom to make significant choices to pursue advanced computer science as best suited to their interests, talents, and resources.

Conclusion

We have argued that students should learn that they themselves can control computer technology, and by choosing wisely they have the power not only to promote their academic learning but to effect meaningful and ethical change in and around their lives. Toward these goals, we offered a constructivist account of educational computing and a range of educational activities for both users and future system designers. We should be clear, however, that many of these activities encourage users to think about design considerations and designers to consider the social impact of their work. Our goals here fit within some of the broader conceptions in the field of human-computer interaction, which seeks not only to create better designs sensitive to the users' needs but to enfranchise and validate users in the design process (Bodker, 1991; Greenbaum and Kyng, 1991; Namioka and Schuler, 1990; Shneiderman and Rose, in press; Winograd and Flores, 1986). Through such considerations, power can be apportioned more equitably. Moreover, the idea of the "technocrat" as keeper and definer of the technology can give way to a more embracing conception of the designer and user as essentially linked to help create the infrastructure within which we live and work.

ized textbooks. Many teachers, especially young women, rightly felt exploited by low pay, poor working conditions, and an expanding curriculum for which they felt either ill prepared to teach or, more usually, had insufficient time to prepare quality lessons for. The standardized text was one way to solve parts of this dilemma, even though it may actually have undercut some of their emerging autonomy at the same time. Some elements of this story are told in Marta Danylewycz and Alison Prentice (1984), "Teachers, Gender, and Bureaucratizing School Systems in Nineteenth Century Montreal and Toronto," *History of Education Quarterly* 24 (Spring), pp. 75–100.

26. See, for example, Carol Gilligan (1982), *In a Different Voice*, Cambridge: Harvard University Press. For a general discussion of the issue of gender and experience, see R. W. Connell (1987), *Gender and Power*, Stanford: Stanford University Press; Leslie Roman, Linda Christian-Smith, and Elizabeth Ellsworth, eds. (1988), *Becoming Feminine*, Philadelphia: Falmer Press; and R. W. Connell (1995), *Masculinities*, Cambridge: Polity Press. We do *not* want to essentialize women, however; nor do we want to claim that all women are uniform. We instead are making a conjunctural argument that at this historical time, in this situation, such resources and strategies tend to come into play.

27. Sandra Acker (1988), "Teachers, Gender and Resistance," *British Journal of Sociology of Education* 9 (no. 3), p. 314. See also, Apple, *Teachers and Texts*; and the extensive review of the research on gender and teaching in Sandra Acker, "Gender and Teachers' Work," in Michael W. Apple, ed. (1995), *Review of Research in Education Volume 21*, Washington: American Educational Research Association, pp. 99–162.

28. Acker, "Teachers, Gender and Resistance," p. 307.

29. For a review of some of this literature, see Apple, *Education and Power*; and Alice Kessler-Harris (1982), *Out to Work*, New York: Oxford University Press.

30. Apple, *Official Knowledge*; and Apple, *Cultural Politics and Education*. For a more general theoretical discussion of the process of commodification and what conceptual resources might be necessary to understand it in all its complexity, see the dense but important book by Philip Wexler (1988), *Social Analysis and Education*, New York: Routledge.

Chapter 7

1. An earlier version of this paper was presented at the April 1993 annual meeting of the American Educational Research Association, Atlanta, Georgia. Authors' addresses: Peter H. Kahn, Jr., Department of Education and Human Development, Colby College, Waterville, ME 04901. Electronic mail: phkahn@colby.edu. Batya Friedman, Department of Mathematics and Computer Science, Colby College, Waterville, ME 04901. Electronic mail: b_friedm@colby.edu.

Sacks, C. H., Y. Bellisimo, and J. Mergendoller. (1993–94). "Attitudes Toward Computers and Computer Use: The Issue of Gender." *Journal of Research on Computing in Education, 26*(2): 256–69.

Sadker, M., and D. Sadker. (1985). "Sexism in the Schoolroom of the '80s." *Psychology Today, 19*: 54–57.

Sanders, J. S. (1984). "The Computer: Male, Female, or Androgynous." *The Computing Teacher, 11*: 32–34.

Sanders, J. S., and A. Stone. (1986). *The Neuter Computer: Computers for Girls and Boys*. New York: Neal-Schuman Publishers, Inc.

Schofield, J. W. (1995). *Computers and Classroom Culture*. Cambridge: Cambridge University Press.

Schumacher, P., J. Morahan-Martin, and A. Olinsky. (1993). "Computer Experiences, Attitudes, Computer and mathematical Anxiety, and Grades of MBA Students." *Collegiate Microcomputer, 11*(3): 183–93.

Shashaani, L. (1994). "Socioeconomic Status, Parents' Sex-Role Stereotypes, and the Gender Gap in Computing." *Journal of Research on Computing in Education, 26*(4): 433–51.

———. (1995). "Gender Differences in Mathematics Experience and Attitude and Their Relation to Computer Attitude." *Educational Technology, 35*(3): 32–38.

Siann, G., H. Macleod, P. Glissov, and A. Durndell. (1990). "The Effect of Computer Use on Gender Differences in Attitudes to Computers." *Computers and Education, 14*(2): 183–91.

Smith, S. D. (1986). "Relationships of Computer Attitudes to Sex, Grade-level, and Teacher Influence." *Education, 106*(3): 338–44.

———. (1987). "Computer Attitudes of Teachers and Students in Relationship to Gender and Grade Level." *Journal of Educational Computing Research, 3*(4): 479–94.

Sutton, R. E. (1991). "Equity and Computers in the Schools: A Decade of Research." *Review of Educational Research, 61*(4): 475–503.

Temple, L., and H. M. Lips. (1989). "Gender Differences and Similarities in Attitudes Toward Computers." *Computers in Human Behavior, 5*(4): 215–26.

The Tico Times. (1989). $77.9 *Billion Budget Rushed Through Congress*. 8 December.

Trejos, A., ed. (1991). *Geografía ilustrada: Costa Rica*. San José, Costa Rica: Trejos Editores.

Turkle, S. (1984). *The Second Self: Computers and the Human Spirit*. New York: Simon & Schuster.

Turkle, S., and S. Papert. (1990). "Epistemological Pluralism: Styles and Voices Within the Computer Culture." *Signs: Journal of Women in Culture and Society, 16*(1): 128–57.

Wahab, Z. (1983). *Education in Costa Rica: An Overview*. Paper presented at the Annual Conference of the Western Region of the Comparative and International Education Society and the Far Western Philosophy of Education Society, Los Angeles, CA.

Ware, M. C., and M. F. Stuck. (1985). "Sex-Role Messages vis-à-vis Microcomputer Use: A Look at the Pictures." *Sex Roles, 13*(3–4): 205–14.

Wilder, G., D. Mackie, and J. Cooper. (1985). "Gender and Computers: Two Surveys of Computer-Related Attitudes." *Sex Roles, 13*(3–4): 215–28.

Williams, J. E., and D. L. Best. (1982). *Measuring Sex Stereotypes: A Thirty Nation Study*. London: Sage.

Williams, S. W., and S. M. Rosenwasser. (1987–89). "Computer Interest Differences in Preschool Children According to Sex and Psychological Sex-Typing." *Psychology and Human Development, 2*(2): 55–60.

Woodrow, J. E. (1994). "The Development of Computer-Related Attitudes of Secondary Students." *Journal of Educational Computing Research, 11*(4): 307–38.

Chapter 7

"A Summary of the ACM/IEEE-CS Joint Curriculum Task force Report: Computing Curricula 1991." (1991). *Communications of the ACM, 34*(6): 69–84.

Appadurai, A., ed. (1988). *The social life of things*. New York: Cambridge University Press.

Appel, C. (1985). "L-S 'superusers' run the computer room." *The Sudbury Town Crier*, May 16, pp. 3, 43.

Arnstine, D. (1973). "The use of coercion in changing the schools." *Educational Theory, 23*: 277–87.

Arsenio, W., and A. Lover. (1995). "Children's conceptions of sociomoral affect: Happy victimizers, mixed emotions, and other expectancies." In M. Killen and D. Hart, eds., *Morality in everyday life: Developmental perspectives*, pp. 87–128. New York: Cambridge University Press.

Baldwin, J. M. (1973). *Social and ethical interpretations in mental development*. New York: Arno. (Original work published 1897)

Baszucki. (1992). *Interactive physics II.* [Computer program]. San Francisco: Knowledge Revolution.

Berlins, M., and L. Hodges. (1981). "Nationality Bill sets out three new citizenship categories." *The London Times*, January 15, pp. 1, 15.

Bodker, S. (1991). *Through the interface: A human activity approach to user interface design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Bromley, H. (1992). "Culture, power, and educational computing." In C. Bigum and B. Green, eds., *Understanding the new information technologies in education.* Deakin University, Australia.

Bynum, T. W., W. Maner, and J. L. Fodor. (1992). *Teaching computer ethics.* New Haven, CT: Research Center on Computing and Society.

Chandrasekaran, B. (1992). "What's in a logo?" *IEEE Expert*, 7(5): 2.

Cummins, J. (1988). "From the inner city to the global village: The microcomputer as a catalyst for collaborative learning and cultural interchange." *Language, Culture, and Curriculum*, *1*(1): 1–13.

Damon, W. (1977). *The social world of the child.* San Francisco: Jossey-Bass.

Dawkins, R. (1976). *The selfish gene.* New York: Oxford University Press.

Denning, P. J. (1992). "Educating a new engineer." *Communications of the ACM*, *35*(12): 83–97.

DeVries, R. (1988). *Constructivist education.* Paper presented to the Association for Constructivist Teaching, West Point Academy.

DeVries, R., and L. Kohlberg. (1990). *Constructivist early education: Overview and comparison with other programs.* Washington, DC: National Association for the Education of Young Children.

DeVries, R., and B. Zan. (1994). *Moral classrooms, moral children: Creating a constructivist atmosphere in early education.* New York: Teachers College.

Dewey, J. (1966). *Democracy and education.* New York: Macmillan. (Original work published 1916)

Dormer, J. (1981). "Computer center is a novel program." *Concord Journal*, June 18, p. 19.

Ermann, M. D., M. B. Williams, and C. Gutierrez, eds. (1990). *Computers, ethics, and society.* New York: Oxford University Press.

Friedman, B. (1986). "If I only had one more computer. . . . Facing the sticky issues of resource allocation." *Classroom Computer Learning*, October, pp. 44–45.

———. (1988). *Problems and solutions: A case study of an exceptional elementary school and its developing computer education program.* Paper presented at the annual meeting of the International Association of Computing Education, New Orleans.

———. (1991). "Social and moral development through computer use: A constructivist approach." *Journal of Research on Computing in Education, 23*: 560–67.

———. (1997). "Social judgments and technological innovation: Adolescents' understanding of property, privacy, and electronic information." *Computers in Human Behavior, 13*: 327–52.

Friedman, B., and P. H. Kahn, Jr. (1992). "Human agency and responsible computing: Implications for computer system design." *Journal of Systems Software, 17*: 7–14.

Friedman, B., and L. Millet. (1995). "'It's the computer's fault'—Reasoning about computers as moral agents." *Proceedings of the Chi' 95 Conference*, pp. 226–27. New York: Association for Computing Machinery.

Friedman, B., and H. Nissenbaum. (1996). "Bias in computer systems." *ACM Transactions on Information Systems, 14*: 330–47.

Friedman, B., and T. Winograd, eds. (1990). *Computing and social responsibility: A collection of course syllabi.* Palo Alto, CA: Computer Professionals for Social Responsibility.

Gotterbarn, D. (1992). "The use and abuse of computer ethics." *Journal of Systems Software, 17*: 75–80.

Greenbaum, J., and M. Kyng, eds. (1991). *Design at work: Cooperative design of computer systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Harvey, B. (1980). *Using computers for educational freedom.* Paper presented at the Second annual Computer Conference, Lesley College, Boston.

———. (1983). *The Lincoln-Sudbury Regional High School computer department: History, goals, strategies, and problems.* Unpublished manuscript.

Hogan, R. (1975). "Theoretical egocentricism and the problem of compliance." *American Psychologist, 30*: 533–39.

Horowitz, P. (1984). "Telecommunications: An electronic, interstate school newspaper." *Electronic Learning*, p. 48.

Johnson, D. G. (1985). *Computer ethics.* Englewood Cliffs, NJ: Prentice-Hall.

Johnson, D. G., and H. Nissenbaum, eds. (1995). *Computers, ethics, & social values.* Englewood Cliffs, NJ: Prentice Hall.

Kahn, P. H., Jr. (1991). "Bounding the controversies: Foundational issues in the study of moral development." *Human Development, 34*: 325–40.

———. (1992). "Children's obligatory and discretionary moral judgments." *Child Development, 63*: 416–30.

———. (1993). *A culturally sensitive analysis of culture in the context of context: When is enough enough*? Paper presented at the biennial meeting of the Society for Research in Child Development, New Orleans. (ERIC Document Reproduction No. ED 365 616)

———. (1995). Commentary on D. Moshman's "The construction of moral rationality." *Human Development, 38*, 282–88.

Kahn, P. H., Jr., and B. Friedman. (1995). "Environmental views and values of children in an inner-city Black community." *Child Development, 66*: 1403–17.

Kearsley, G., B. Hunter, and M. Furlong. (1992). *We teach with technology: New visions for education.* Wilsonville, OR: Franklin, Beedle & Associates.

Kiesler, S. and L. Sproull. (in press). "Addendum to 'When the Interface is a Face,' or what is 'social' about a computer?" In B. Friedman, ed., *Human values and the design of computer technology.* Stanford: Center for the Study of Language and Information, Stanford University.

Kling, R., ed. (1996). *Computerization and controversy: Value conflicts and social choices* (2nd ed.). San Diego, CA: Academic Press.

Kohlberg, L. (1969). "Stage and sequence: The cognitive-developmental approach to socialization." In D. A. Goslin, ed., *Handbook of socialization theory and research*, pp. 347–480. New York: Rand McNally.

———. (1980). "High school democracy and educating for a just society." In R. L. Mosher, ed., *Moral education: A first generation of research*, pp. 20–57. New York: Praeger.

———. (1984). *Essays on moral development: The psychology of moral development.* Vol. 2. San Francisco: Harper & Row.

———. (1985). "The just community approach to moral education in theory and practice." In M. W. Berkowitz and F. Oser, eds., *Moral education: Theory and application*, pp. 27–87. Hillsdale, NJ: Lawrence Erlbaum.

Laupa, M. (1991). "Children's reasoning about three authority attributes: Adult status, knowledge, and social position." *Developmental Psychology, 27*: 321–29.

Lewis, P. H. (1992). "Nurturing democracy through used computers." *The New York Times*, November 8, p. F12.

Lourenco, O. (1996). "Reflections on narrative approaches to moral development." *Human Development, 39*: 83–99.

Martinez, M. (1993). *Developments in moral education across the curriculum.* Paper presented at the April meeting of the American Educational Research Association, Atlanta.

Miller, K. (1988). "Integrating computer ethics into the computer science curriculum." *Computer Science Education, 1*: 37–52.

Namioka, A., and D. Schuler, eds. (1990). *Proceedings of the 1990 Participatory Design Conference.* Palo Alto, CA: Computer Professionals for Social Responsibility.

Nass, C., Y. Moon, J. Morkes, E. Kim, and B. J. Fogg. (in press). "Computers are social actors: A review of current research." In B. Friedman, ed., *Human values and the design of computer technology.* Stanford: Center for the Study of Language and Information, Stanford University.

Nass, C., J. Steuer, E. Tauber, and H. Reeder. (1993). "Anthropomorphism, Agency, and Ethopoeia: Computers as social actors." In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, and T. White, eds., *InterCHI '93 adjunct proceedings*, pp. 111–12. Amsterdam, The Netherlands.

Noble, D. D. (1991). *The classroom arsenal: Military research, information technology and public education.* London: Falmer Press.

Nucci, L. (1996). "Morality and the personal sphere of actions." In E. S. Reed, E. Turiel, and T. Brown, eds., *Values and knowledge*, pp. 41–60. Mahwah, NJ: Lawrence Erlbaum.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Parker, D. B., S. Swope, and B. N. Baker. (1990). *Ethical conflicts in information and computer science, technology, and business.* Wellesley, MA: QED Informations Sciences.

Perrolle, J. A. (1987). *Computers and social change: Information, property, and power.* Belmont, CA: Wadsworth.

Piaget, J. (1969). *The moral judgment of the child.* Glencoe, IL: Free Press. (Original work published 1932)

Power, C., A. Higgins, and L. Kohlberg. (1989). "The habit of the common life: Building character through democratic community schools." In L. Nucci, ed., *Moral development and character education: A dialogue*, pp. 125–43. Berkeley: McCutchan.

Rohwer, W. D., Jr., C. P. Rohwer, and J. R. B-Howe. (1980). *Educational psychology: Teaching for student diversity.* New York: Holt, Rinehart, & Winston.

Rumelhart, D. E., and D. A. Norman. (1981). "Analogical processes in learning." In J. R. Anderson, ed., *Cognitive skills and their acquisition*. Hillsdale, NJ: Lawrence Erlbaum.

Scheffler, I. (1991). *In praise of cognitive emotions and other essays in the philosophy of education*. New York: Routledge.

Searle, J. R. (1981). "Minds, brains, and programs." In D. R. Hofstadter and D. C. Dennett, eds., *The mind's I*, pp. 353–82. New York: Basic Books.

———. (1984). *Minds, brains and science*. Cambridge: Harvard University Press.

———. (1990). "Is the brain's mind a computer program?" *Scientific American*, *262*: 26–31.

———. (1992). *The rediscovery of the mind*. Cambridge: MIT Press.

Selman, R. L. (1980). *The growth of interpersonal understanding*. New York: Academic Press.

Shneiderman, B. (1987). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.

Shneiderman, B., and A. Rose. (in press). "Social impact statements: Engaging public participation in information technology design." In B. Friedman, ed., *Human values and the design of computer technology*. Stanford: Center for the Study of Language and Information, Stanford University.

Shweder, R. A. (1986). "Divergent rationalities." In D. W. Fiske and R. A. Shweder, eds., *Metatheory in social science: pluralisms and subjectivities*, pp. 163–96. Chicago: University of Chicago Press.

Smetana, J. (1982). *Concepts of self and morality: Women's reasoning about abortion*. New York: Praeger.

Snyder, T. (1985a). *Our town meeting*. [Computer program]. Cambridge, MA: Tom Snyder Productions.

———. (1985b). *The other side*. [Computer program]. Cambridge, MA: Tom Snyder Productions.

Sproull, L., M. Subramani, S. Kiesler, J. H. Walker, and K. Waters. (1996). "When the interface is a face." *Human-Computer Interaction*, *11*: 97–124.

Turiel, E. (1997). "The development of morality." In W. Damon, ed., *Handbook of child psychology*, 5th Ed., Vol 3: N. Eisenberg, ed., *Social, emotional, and personality development*. New York: Wiley.

Turkle, S. (1984). *The second self: Computers and the human spirit*. New York: Simon & Schuster.

US General Accounting Office. (1992). Patriot missile defense: Software problem led to system failure at Dhahran, Saudi Arabia (GAO/IMTEC-92–26). Washington, D.C.

Vygotsky, L. S. (1978). *Mind in society*. Cambridge: Harvard University Press.

Wainryb, C. (1995). "Reasoning about social conflicts in different cultures: Druze and Jewish children in Israel." *Child Development*, *66*: 390–401.

Weizenbaum, J. (1976). *Computer power and human reason*. New York: Freeman and Co.

Wigginton, E. (1986). *Sometimes a shining moment: The Foxfire experience*. Garden City, NY: Anchor Books.

Winograd, T. (1992). "Computers, ethics, and social responsibility." *The CPSR Newsletter*, *10*(3): pp. 17–27.

Winograd, T., and F. Flores. (1986). *Understanding computers and cognition: A new foundation for design*. Reading, MA: Addison-Wesley.