The Steam Engine and the Computer: What Makes Technology Revolutionary

#654

by Herbert A. Simon

Herbert A. Simon, the Richard King Mellon University professor of computer sciences and psychology at Carnegie Mellon University, received the Nobel Prize in economic sciences in 1978. He has authored and co-authored numerous books including Organizations, The Sciences of the Artificial, *and* Reason in Human Affairs. *Recipient of the A.M. Turing Award in 1975 from the Association for Computing Machinery, Herbert Simon is one of the world's leading authorities on human decision making and artificial intelligence.*

This article is adapted from remarks delivered to the EDUCOM '86 Conference, November 12, 1986.

It may seem absurdly anachronistic to be reflecting about the steam engine in a conference devoted to the role of computers in higher education. But it's often been said that computers have triggered a second industrial revolution, so perhaps there are some lessons to be learned from the First Industrial Revolution, the one that was triggered by the steam engine—lessons that might have some bearing on what we can and should do with computers and on what computers might do to and for us.

The First Industrial Revolution

We think of revolutions as being sudden events, producing far-reaching changes in a very short period of time. But the revolution launched by the steam engine took, by any reasonable account, 150 years. The invention that started it was Thomas Newcomen's "atmospheric" steam engine, which appeared in about 1711. Newcomen developed his engine primarily because his coal mines were being flooded and he needed more powerful pumps than were currently available to get the water out of the mines. James Watt made some important improvements on the engine in 1769, in the course of trying to repair one of Newcomen's engines. Two generations had already gone by, and one could hardly speak of a revolution yet. The aims and aspirations of these "revolutionaries" were distinctly limited.

In order for the steam engine to produce genuinely revolutionary change, there had to be a whole series of subsequent inventions, none of which were—or could have been—contemplated by its originators, and these took still another generation, reaching well into the 19th century. The steam engine was adapted for use in transportation, giving us the steamboat and the steam locomotive, and in industry, the power loom. Perhaps even more important were a number of complementary inventions that were initially quite independent of the steam engine but that were harnessed to it to produce further changes. The most notable of these was the dynamo, which used the steam engine to generate electricity. (To convince you of how long this revolution took, let me remind you that there were many rural areas in the United States that didn't have electricity until about 50 years ago.) Then there was another series of what it is fair to call "derivative" inventions—the internal combustion engine, the automobile, the electric light bulb, the airplane, the telephone. The point is that when technology reshapes society, it is not the result of a single invention but of a host of additional, completely unanticipated inventions, many of them of the same order of magnitude as the first one in the chain.

Conversely, what we sometimes call technological change actually permeates society, affecting it in far more than merely "technological" ways. Before the automobile, one of the most important skills of a physician, certainly the one he used the most, was the skill of driving a horse. For a while, that was replaced by the less-time consuming activity of driving an automobile. Now, physicians don't even have to do that; their patients come to them. We don't ordinarily think of the steam engine as changing medical practice, yet it did; it certainly altered the physician's time budget. The creation of the suburbs is another example. And—though again it seem surprising at first—one of the largest migrations in human history was brought about by air conditioning, which transformed what many people thought of as uninhabitable parts of our country into the very attractive sunbelt. The invention of air conditioning is usually dated at about 1911, and no one at that time could have anticipated that, as a result of it, a large part of the American population was going to move from one part of the country to another. As one more example, the burning of coal to produce steam for the generation of electricity had a number of adverse consequences, leading us to look for other fuels that could be used for the purpose. As a result, we opened that Pandora's box known as nuclear power.

I've already alluded to one lesson to be drawn from all of this, and that is the lesson of unpredictability. There are no crystal balls that can tell us what the consequences of a fundamental technological change are going to be. A genealogical chart of the First Industrial Revolution would encompass about six generations. Parents all come to understand the impossibility of foretelling how their children are going to turn out; how much more futile it would be for them to try to imagine what their great-great-great-great-grandchildren will be like.

A second lesson, also alluded to, is the extent to which the ramifications of any one technological change depend upon the stimulation it provides to other inventions, and the links that are made from it to inventions that may be independent of it, as steam was linked to electricity.

A third lesson is the importance of what we might call "education by immersion." Most Americans, after all, did not learn to drive automobiles in driver-education classes. Instead, they learned to drive because there was a Model T on the farm, or maybe a tractor, and there was something or someone that had to be moved from here to there—so they got in their cars and figured out what all those levers and pedals did, and they also learned, out of necessity, how to take the car apart and put it together again. None of this was planned ahead of time; nobody sat down and figured out the kinds of courses that would be needed in order to teach people how to use these new contraptions. We educated ourselves about them because we had to, and it was easy to do because they were all around us.

A final lesson to be drawn from this history is the lesson of generality. In the last analysis, the reason that the steam engine and the associated inventions proved to be revolutionary is that they didn't do anything specifically. Rather, they allowed us to move in innumerable directions. They replaced and augmented human and other forms of animal muscle by the muscle of engines, thereby completely transforming the nature of that major input into everything we produce—energy. No single-purpose device is going to bring about a revolution, however convenient or useful it may be. Revolutionary significance lies in generality.

Potential of the Computer

How are these lessons applicable today? If we were to make a genealogical chart for the second industrial revolution, it would of course be far less elaborate than the one for the first, because computers have been around for only about 40 years. Though there have already been a number of derivative inventions, most of them are still fairly closely related to the original conception of the computer whether it is solid-state hardware, time sharing (now almost outmoded), higher-level languages or methods of nonnumerical computation of various sorts, primarily means of making the computer faster and more powerful. At most, there have been two generations so far. It is true that people in the hardware business like to say that they are now in the fifth generation, but that's a little like asking us to accept child marriage. I think it's more accurate to say we're now in the third generation, and even that one is at most in its adolescence.

This third generation is identified by several elements. One is the appearance of minicomputers and microcomputers. Their significance is not in the opportunity they give us to play games at home or to keep the family accounts, but in the fact that they open up the possibility of computer education by immersion. Today, for the first time, we can say that a very large proportion of the American population has had hands-on exposure to a computer (and the event is recent enough to be still very vivid in the minds of most of us). Another element is the development of computer graphics and of new kinds of workstations. Still another is computer-aided instruction. Of course, computer-aided instruction has been around in some form for almost as long as computers have been. We have had an instructional management game in operation at the business school of Carnegie Mellon University since at least 1960. Nevertheless, computer-aided instruction is still rather primitive and has had only a very modest effect on the way education takes place in this country or anywhere else. Robotics, expert systems, and cognitive science are also part of the third generation of the second industrial revolution. And finally, computers are beginning to form links with other parts of technology, in particular with our systems of communications and information transfer. Networking is one example of this—the creation of a system in which computers no longer stand alone but can talk not only to us as individuals, but also to each other, in a great variety of ways.

Surely, the second industrial revolution is just as unpredictable as the first one was—and the second has barely begun. We are closer in time to the first computer than James Watt was to Thomas Newcomen. There is a lot of solemn talk about what computers can't do—there's even a book by that name[1]—but that's not a very interesting subject. Computers today are doing a lot of things they were "known" to be unable to do a while ago, and what they can't do today they may very well be doing tomorrow. (The author of that book has already had to get out several revised editions.) Besides, our task is not to decide what computers can't do but to look ahead for the very short distance that we are capable of and to think about what we can get computers to do, what we would like them to do that they can't do right now. Each year we see more impressive computer systems for handling natural language. Still, most of them are limited to particular domains of discourse. I know of no computer program today that can enter into a general conversation with you about anything you want to bring up. There is plenty of room for developments of that kind without worrying about whether there are ultimately some things that computers will never be able to do.

It is sometimes said, by way of demonstrating the superiority of human beings, that computers cannot be imprecise, even when the situation demands imprecision. I'm not sure that this is a shortcoming—how much have humans really gained from their ability to be imprecise?—but anyone who has had much experience with computers is not likely to believe that they cannot be imprecise. More and more, we are using computers in tasks where the exact course of action is not determinate at each moment. A problem solving program, an expert system, is an organized but highly flexible way of making sure of information that is incomplete and imperfect and that comes to it in a variety of forms and sequences. A capability for imprecision no longer marks the boundary between what computers can and cannot do.

Human beings are also said to possess "intuition." That's a term we use when someone looks at a problem and doesn't at first know the answer, but a minute or two or even just a few seconds later, does know the answer, and knows it without any awareness of the process by which the answer was found. When we see someone coming down the street, we may not know who it is at first,

> *"No single-purpose device is going to bring about a revolution, however convenient or useful it may be."*

but if it's a friend we are likely to know it long before we know how we knew it. But if that's what intuition is, then it can be said that computers have it, too. An important part of the anatomy of an expert system is a database, indexed by a set of cues that the system can recognize. When one of those cues appears, the system retrieves relevant information from its database. A medical diagnosis system, for example, can be presented with a few symptoms, and it will "intuit" what the ailment is. Of course, like any prudent diagnostician, it doesn't entirely trust its intuition, so it goes on to request that certain tests be made, and with the additional information, it confirms or refines or alters the diagnosis. Unless we deliberately want to make a mystique out of human thinking and sometimes it seems that we do because it makes us feel better about ourselves, we have to conclude that what's going on in such a system is exactly what we call intuition when a human being does it. Once again, the supposed contrast between what people can do and what computers can do doesn't fit the facts of computation in the modern world.

When we're especially impressed by an intuition, we are apt to call it "insight," and beyond that we begin to talk about "creativity." Here too, however, there are computer programs that can do things that would be regarded as "insightful" and "creative" if they were done by humans. Theorems in mathematics and logic have been discovered and proved by computers unaided by human hands or minds. Some of my associates and I have been working for six or eight years on a program that we call BACON. We have given it, for example, the data that Kepler had available to him at the time he made his discovery of the third law of planetary motion or that Joseph Black had available when he proposed his theory of specific heat, and we found that the program is able to rediscover the law or the theory.[2] We can conclude that we don't have to postulate any kind of mysterious processes, any kind of fundamentally unknown—much less unknowable—processes to account for what in humans we like to call creativity. Hence, we can't establish the creative process as an Iron Curtain that limits what computers can do.

Computing by Immersion

The first computer we acquired at Carnegie Mellon was an IBM 650. When we got it, in about 1958, we didn't have any idea what we were going to do with it. The electrical engineers didn't want to be associated with it because they were afraid they would have to maintain it; the mathematicians didn't want to be associated with it because it seemed beneath their dignity. So we put it in the basement of the business school, but we were very careful not to put a lock on the door. What happened was that the students swarmed all over it, just as previous generations had swarmed over the Model T. It was then that the faculty began to learn about it, to save themselves from embarrassment. What was happening was education by immersion.

The same phenomenon could be seen in industry. In those early days, a business executive would decide that his company ought to have a computer, often merely because one of his colleagues at another company had just gotten one and he wanted to be progressive and up-to-date. On occasion, I was called in to advise in such situations, and my advice was usually, "Before you buy a computer, decide what you intend to do with it, and then plan your installation around those intended uses." That was terrible advice, and I probably ought to return the fees I got for giving it. Fortunately, my advice often wasn't taken, because the motivation for getting the computer was not to use it but just to have it. But then it arrived, and when I observed what happened afterward, I realized that the best advice was, in fact, just to have one around, for that's the only way a company or a university, or anybody else, is going to learn what to do with it. You have to make friends with it, talk to it, let it talk to you. Hit the keys and see what happens. The computer will tell you about itself, and it tells you probably better than the instruction book. Immerse yourself in the technology. I don't mean we shouldn't have classes about computers—after all, I work for a university—but they are not going to be the major educational force. One of the most impressive things about a computer is its capacity, by virtue of its power and the flexibility of its responses, to be its own instructional device.

Besides helping us learn about themselves, computers can play a role in helping us learn about other things as well. On that subject, I bring to your attention a very informative and thoughtful article by President Bok of Harvard in a recent issue of the EDUCOM Bulletin.[3] Among its other virtues, it is mercifully free—or almost so—of smug remarks about what computers can't do. We are indeed a long way from knowing what they can do for education. In our university, we certainly don't have a detailed blueprint for what our campus is going to look like with a network, or with all the things a network will bring about. We are engaged in an exploration, an adventure.

At Carnegie Mellon today, we have a computerized campus mail system. It's a great convenience, and I think students find it so, too. I'm much more accessible to them now than I was ever able to be before. They can write me a note, without having to go through a secretary, and what's more, I often answer it, and promptly, because it's so easy to do. A fair amount of my correspondence is conducted that way now. It has obvious advantages even over the telephone, because you don't have to be there when the call is made, and the sender doesn't have to be there when you return the call, so information is actually transferred more quickly than is often possible by phone. That's the kind of modest change computers have brought so far to our campus. Change of this kind is hardly revolutionary.

> *"One of the most impressive things about a computer is its capacity, by virtue of its power and the flexibility of its responses, to be its own instructional device."*

A Revolution in Education?

Before the computer and all the associated devices can have any great impact on the educational system, there have to be major developments in our understanding of what the educational process is. Up to now, particularly at the university level, we have operated on what I call the "infection theory" of learning. This theory holds that if you assemble a large number of people in a room and spray a large number of words at them, some of those words will be infectious and will stick with some of those people and perhaps affect their future behavior. (Another form of the theory is that people are infected if they spray themselves with words from a large number of pages of print.)

A different theory might be called the "Mr. Chips" theory, according to which students learn by being treated with tender loving care. But, while tender loving care may be as important for students as it is for patients in a hospital, it is no more adequate as a theory of learning than it is as a theory of curing disease. The "Mark Hopkins and a log" theory is a variant of the Mr. Chips theory, a peculiarly useless one in view of the fact that we wouldn't have enough logs to accommodate 6,000 students—and Carnegie Mellon is far from being a large university.

Technology has helped to implement the infection theory in a modest sort of way. It has provided the means for broadcasting the words, using microphones and loudspeakers or headphones, and for putting the professors on film. (Sometimes I think that it's only the economic self-interest of professors that demands that they be there live at all.) Though some people believe that technology actually interferes with Mr. Chip's ministrations, I think the contrary is true, as I've already suggested in describing our campus mail system. The idea that having a lot of screens and boxes around makes human beings less interested in talking to each other, or doing all the other kinds of things that human beings do, just isn't borne out by the facts. At Carnegie Mellon, the Computer Science Department has been saturated with networked computers for a dozen years, yet it is the most social and sociable department on campus, both at work and at play.

On the other hand, an improved technology of infection still does not amount to a revolution in education. If computers are to have real educational significance, there will have to be a major advance in what's now called cognitive science. We must gain a much deeper understanding of what it is that a student learns, what it is that a student should learn in order to become capable of exercising particular skills, and how that learning comes about. The theory we need does not so much concern the electronics we have available as it does the human component in the system that does our thinking and our learning. A good deal of progress has been made toward that theory, or at least its foundations, in the past 30 years. Now we are just getting to the point where researchers are beginning seriously to apply it to actual educational procedures.

It seems equally obvious to me that computers will not revolutionize education until there are massive changes in the organizational and administrative structure of the educational system as well. There must first of all be a redefinition of the teacher's role. Perhaps we'll never reach the point of having a completely professor-free university, but at least the professors will have to abandon the theory of infection. Secondly, we have to develop new conceptions of the production and marketing of software. There is no more sense in having each university prepare all its own instructional programs than there would have been in having each one publish its own textbooks. In general, for every megabuck we spend in hardware and systems software, we will need to spend another megabuck for research on effective learning and development of modern learning environments in the schools.

"...computers will not revolutionize education until there are massive changes in the organizational and administrative structure of the educational system as well."

By way of conclusion, let me say that, as I hope the examples of the steam engine and the computer make clear, new technology is simply new knowledge; and as such, it resides not in machines but in the human brains that invent them, develop them, and use them. Even though the machines can help us learn about their characteristics by our use of them, still, in the last analysis, we have to think about technology in terms of human knowledge.

Knowledge confers capabilities, but capabilities, like knowledge itself, can be used for good or ill. Prometheus brought us indispensable knowledge, while Pandora brought us mischievous knowledge. Yet, without denying all the problems that we face in contemporary society, some of which are admittedly an outgrowth of our knowledge, I think that most of us would rather be living in the 20th century than in the 13th. Technological revolutions are not something that "happen" to us. We make them, and we make them for better or for worse. Our task is not to peer into the future to see what computers will bring us, but to shape the future that we want to have—a future that will create new possibilities for human learning, including, perhaps most important of all, new possibilities for learning to understand ourselves.

References

1. *Dreyfus, H.L., What Computers Can't Do.* New York, Harper & Row, 1967 (Rev. edition, 1979).
2. Langley, P., H. Simon, G. Bradshaw, and J. Zytkow. *Scientific Discovery.* Cambridge, Mass., MIT Press, 1986.
3. Derek Bok, "Looking into Education's High-Tech Future," *EDUCOM Bulletin,* 20 (1985), pp. 2-10, 17.