What Is Computer Ethics?

Article *in* Metaphilosophy · August 2007

DOI: 10.1111/j.1467-9973.1985.tb00173.x

1 author:

James Moor
Dartmouth College

METAPHILOSOPHY
Vol. 16, No. 4, October 1985
0026–1068 $2.00

WHAT IS COMPUTER ETHICS?*

JAMES H. MOOR

A Proposed Definition

Computers are special technology and they raise some special ethical issues. In this essay I will discuss what makes computers different from other technology and how this difference makes a difference in ethical considerations. In particular, I want to characterize computer ethics and show why this emerging field is both intellectually interesting and enormously important.

On my view, *computer ethics* is the analysis of the nature and social impact of computer technology and the corresponding formulation and justification of policies for the ethical use of such technology. I use the phrase "computer technology" because I take the subject matter of the field broadly to include computers and associated technology. For instance, I include concerns about software as well as hardware and concerns about networks connecting computers as well as computers themselves.

A typical problem in computer ethics arises because there is a policy vacuum about how computer technology should be used. Computers provide us with new capabilities and these in turn give us new choices for action. Often, either no policies for conduct in these situations exist or existing policies seem inadequate. A central task of computer ethics is to determine what we should do in such cases, i.e., to formulate policies to guide our actions. Of course, some ethical situations confront us as individuals and some as a society. Computer ethics includes consideration of both personal and social policies for the ethical use of computer technology.

Now it may seem that all that needs to be done is the mechanical application of an ethical theory to generate the appropriate policy. But this is usually not possible. A difficulty is that along with a policy vacuum there is often a conceptual vacuum. Although a problem in computer ethics may seem clear initially, a little reflection reveals a conceptual muddle. What is needed in such cases is an analysis which provides a coherent conceptual framework within which to formulate a policy for action. Indeed, much of the important work in computer ethics is devoted to proposing conceptual frameworks for understanding ethical problems involving computer technology.

An example may help to clarify the kind of conceptual work that is required. Let's suppose we are trying to formulate a policy for protecting computer programs. Initially, the idea may seem clear enough. We are looking for a policy for protecting a kind of intellectual property. But then a

* Editor's footnote: This article is the prize-winning essay in *Metaphilosophy*'s essay competition on computer ethics.

number of questions which do not have obvious answers emerge. What is a computer program? Is it really intellectual property which can be owned or is it more like an idea, an algorithm, which is not owned by anybody? If a computer program is intellectual property, is it an *expression* of an idea that is owned (traditionally protectable by copyright) or is it a *process* that is owned (traditionally protectable by patent)? Is a machine-readable program a copy of a human-readable program? Clearly, we need a conceptualization of the nature of a computer program in order to answer these kinds of questions. Moreover, these questions must be answered in order to formulate a useful policy for protecting computer programs. Notice that the conceptualization we pick will not only affect how a policy will be applied but to a certain extent what the facts are. For instance, in this case the conceptualization will determine when programs count as instances of the same program.

Even within a coherent conceptual framework, the formulation of a policy for using computer technology can be difficult. As we consider different policies we discover something about what we value and what we don't. Because computer technology provides us with new possibilities for acting, new values emerge. For example, creating software has value in our culture which it didn't have a few decades ago. And old values have to be reconsidered. For instance, assuming software is intellectual property, why should intellectual property be protected? In general, the consideration of alternative policies forces us to discover and make explicit what our value preferences are.

The mark of a basic problem in computer ethics is one in which computer technology is *essentially* involved and there is an uncertainty about what to do and even about how to understand the situation. Hence, not all ethical situations involving computers are central to computer ethics. If a burglar steals available office equipment including computers, then the burglar has done something legally and ethically wrong. But this is really an issue for general law and ethics. Computers are only *accidently* involved in this situation, and there is no policy or conceptual vacuum to fill. The situation and the applicable policy are clear.

In one sense I am arguing for the special status of computer ethics as a field of study. Applied ethics is not simply ethics applied. But, I also wish to stress the underlying importance of general ethics and science to computer ethics. Ethical theory provides categories and procedures for determining what is ethically relevant. For example, what kinds of things are good? What are our basic rights? What is an impartial point of view? These considerations are essential in comparing and justifying policies for ethical conduct. Similarly, scientific information is crucial in ethical evaluations. It is amazing how many times ethical disputes turn not on disagreements about values but on disagreements about facts.

On my view, computer ethics is a dynamic and complex field of study which considers the relationships among facts, conceptualizations, policies and values with regard to constantly changing computer technology. Computer ethics is not a fixed set of rules which one shellacs and hangs on the wall. Nor is computer ethics the rote application of ethical principles to a value-free technology. Computer ethics requires us to think anew about the nature of computer technology and our values. Although computer ethics is a field between science and ethics and depends on them, it is also a discipline in its own right which provides both conceptualizations for understanding and policies for using computer technology.

Though I have indicated some of the intellectually interesting features of computer ethics, I have not said much about the problems of the field or about its practical importance. The only example I have used so far is the issue of protecting computer programs which may seem to be a very narrow concern. In fact, I believe the domain of computer ethics is quite large and extends to issues which affect all of us. Now I want to turn to a consideration of these issues and argue for the practical importance of computer ethics. I will proceed not by giving a list of problems but rather by analyzing the conditions and forces which generate ethical issues about computer technology. In particular, I want to analyze what is special about computers, what social impact computers will have, and what is operationally suspect about computing technology. I hope to show something of the nature of computer ethics by doing some computer ethics.

The Revolutionary Machine

What is special about computers? It is often said that a Computer Revolution is taking place, but what is it about computers that makes them revolutionary? One difficulty in assessing the revolutionary nature of computers is that the word "revolutionary" has been devalued. Even minor technological improvements are heralded as revolutionary. A manufacturer of a new dripless pouring spout may well promote it as revolutionary. If minor technological improvements are revolutionary, then undoubtedly everchanging computer technology is revolutionary. The interesting issue, of course, is whether there is some nontrivial sense in which computers are revolutionary. What makes computer technology importantly different from other technology? Is there any real basis for comparing the Computer Revolution with the Industrial Revolution?

If we look around for features that make computers revolutionary, several features suggest themselves. For example, in our society computers are affordable and abundant. It is not much of an exaggeration to say that currently in our society every major business, factory, school, bank, and hospital is rushing to utilize computer technology. Millions of personal computers are being sold for home use. Moreover, computers are integral parts of products which don't look much like computers such as watches and automobiles. Computers are abundant and inexpensive, but so are pencils. Mere abundance and affordability don't seem sufficient to justify any claim to technological revolution.

One might claim the newness of computers makes them revolutionary. Such a thesis requires qualification. Electronic digital computers have been around for forty years. In fact, if the abacus counts as a computer, then computer technology is among the oldest technologies. A better way to state this claim is that recent engineering advances in computers make them revolutionary. Obviously, computers have been immensely improved over the last forty years. Along with dramatic increases in computer speed and memory there have been dramatic decreases in computer size. Computer manufacturers are quick to point out that desk top computers today exceed the engineering specifications of computers which filled rooms only a few decades ago. There has been also a determined effort by companies to make computer hardware and computer software easier to use. Computers may not be completely user friendly but at least they are much less unfriendly. However, as important as these features are, they don't seem to get to the heart of the Computer Revolution. Small, fast, powerful and easy-to-use electric can openers are great improvements over earlier can openers, but they aren't in the relevant sense revolutionary.

Of course, it is important that computers are abundant, less expensive, smaller, faster, and more powerful and friendly. But, these features serve as enabling conditions for the spread of the Computer Revolution. The essence of the Computer Revolution is found in the nature of a computer itself. What is revolutionary about computers is *logical malleability*. Computers are logically malleable in that they can be shaped and molded to do any activity that can be characterized in terms of inputs, outputs, and connecting logical operations. Logical operations are the precisely defined steps which take a computer from one state to the next. The logic of computers can be massaged and shaped in endless ways through changes in hardware and software. Just as the power of a steam engine was a raw resource of the Industrial Revolution so the logic of a computer is a raw resource of the Computer Revolution. Because logic applies everywhere, the potential applications of computer technology appear limitless. The computer is the nearest thing we have to a universal tool. Indeed, the limits of computers are largely the limits of our own creativity. The driving question of the Computer Revolution is "How can we mold the logic of computers to better serve our purposes?"

I think logical malleability explains the already widespread application of computers and hints at the enormous impact computers are destined to have. Understanding the logical malleability of computers is essential to understanding the power of the developing technological revolution. Understanding logical malleability is also important in setting policies for the use of computers. Other ways of conceiving computers serve less well as a basis for formulating and justifying policies for action.

Consider an alternative and popular conception of computers in which computers are understood as number crunchers, i.e., essentially as numerical devices. On this conception computers are nothing but big calculators. It might be maintained on this view that mathematical and scientific applications should take precedence over nonnumerical applications such as word processing. My position, on the contrary, is that computers are logically malleable. The arithmetic interpretation is certainly a correct one, but it is only one among many interpretations. Logical malleability has both a syntactic and a semantic dimension. Syntactically, the logic of computers is malleable in terms of the number and variety of possible states and operations. Semantically, the logic of computers is malleable in that the states of the computer can be taken to represent anything. Computers manipulate symbols but they don't care what the symbols represent. Thus, there is no ontological basis for giving preference to numerical applications over non-numerical applications.

The fact that computers can be described in mathematical language, even at a very low level, doesn't make them essentially numerical. For example, machine language is conveniently and traditionally expressed in 0's and 1's. But the 0's and 1's simply designate different physical states. We could label these states as "on" and "off" or "yin" and "yang" and apply binary logic. Obviously, at some levels it is useful to use mathematical notation to describe computer operations, and it is reasonable to use it. The mistake is to reify the mathematical notation as the essence of a computer and then use this conception to make judgments about the appropriate use of computers.

In general, our conceptions of computer technology will affect our policies for using it. I believe the importance of properly conceiving the nature and impact of computer technology will increase as the Computer Revolution unfolds.

Anatomy of the Computer Revolution

Because the Computer Revolution is in progress, it is difficult to get a perspective on its development. By looking at the Industrial Revolution I believe we can get some insight into the nature of a technological revolution. Roughly, the Industrial Revolution in England occurred in two major stages. The first stage was the technological introduction stage which took place during the last half of the Eighteenth Century. During this stage inventions and processes were introduced, tested, and improved. There was an industrialization of limited segments of the economy, particularly in agriculture and textiles. The second stage was the technological permeation stage which took place during the Nineteenth Century. As factory work increased and the populations of cities swelled, not only did well known social evils emerge, but equally significantly corresponding changes in human activities and institutions, ranging from labor unions to health services, occurred. The forces of industrialization dramatically transformed the society.

My conjecture is that the Computer Revolution will follow a similar two stage development. The first stage, the introduction stage, has been occurring during the last forty years. Electronic computers have been created and refined. We are gradually entering the second stage, the permeation stage, in which computer technology will become an integral part of institutions throughout our society. I think that in the coming decades many human activities and social institutions will be transformed by computer technology and that this transforming effect of computerization will raise a wide range of issues for computer ethics.

What I mean by "transformed" is that the basic nature or purpose of an activity or institution is changed. This is marked by the kinds of questions that are asked. During the introduction stage computers are understood as tools for doing standard jobs. A typical question for this stage is "How well does a computer do such and such an activity?" Later, during the permeation stage, computers become an integral part of the activity. A typical question for this stage is "What is the nature and value of such and such an activity?" In our society there is already some evidence of the transforming effect of computerization as marked by the kind of questions being asked.

For example, for years computers have been used to count votes. Now the election process is becoming highly computerized. Computers can be used to count votes and to make projections about the outcome. Television networks use computers both to determine quickly who is winning and to display the results in a technologically impressive manner. During the last presidential election in the United States the television networks projected the results not only before the polls in California were closed but also before the polls in New York were closed. In fact, voting was still going on in over half the states when the winner was announced. The question is no longer "How efficiently do computers count votes in a fair election?" but "What is a fair election?" Is it appropriate that some people know the outcome before they vote? The problem is that computers not only tabulate the votes for each candidate but likely influence the number and distribution of these votes. For better or worse, our electoral process is being transformed.

As computers permeate more and more of our society, I think we will see more and more of the transforming effect of computers on our basic institutions and practices. Nobody can know for sure how our computerized society will look fifty years from now, but it is reasonable to think that various aspects of our daily work will be transformed. Computers have been used for years by businesses to expedite routine work, such as calculating payrolls; but as personal computers become widespread and allow executives to work at home, and as robots do more and more factory work, the emerging question will be not merely "How well do computers help us work?" but "What is the nature of this work?"

Traditional work may no longer be defined as something that normally happens at a specific time or a specific place. Work for us may become less doing a job than instructing a computer to do a job. As the concept of work begins to change, the values associated with the old concept will have to be reexamined. Executives who work at a computer terminal at home will lose some spontaneous interaction with colleagues. Factory workers who direct robots by pressing buttons may take less pride in a finished product. And similar effects can be expected in other types of work. Commercial pilots who watch computers fly their planes may find their jobs to be different from what they expected.

A further example of the transforming effect of computer technology is found in financial institutions. As the transfer and storage of funds becomes increasingly computerized the question will be not merely "How well do computers count money?" but "What is money?" For instance, in a cashless society in which debits are made to one's account electronically at the point of sale, has money disappeared in favor of computer records or have electronic impulses become money? What opportunities and values are lost or gained when money becomes intangible?

Still another likely area for the transforming effect of computers is education. Currently, educational packages for computers are rather limited. Now it is quite proper to ask "How well do computers educate?" But as teachers and students exchange more and more information indirectly via computer networks and as computers take over more routine instructional activities, the question will inevitably switch to "What is education?" The values associated with the traditional way of educating will be challenged. How much human contact is necessary or desirable for learning? What is education when computers do the teaching?

The point of this futuristic discussion is to suggest the likely impact of computer technology. Though I don't know what the details will be, I believe the kind of transformation I am suggesting is likely to occur. This is all I need to support my argument for the practical importance of computer ethics. In brief, the argument is as follows: The revolutionary feature of computers is their logical malleability. Logical malleability assures the enormous application of computer technology. This will bring about the Computer Revolution. During the Computer Revolution many of our human activities and social institutions will be transformed. These transformations will leave us with policy and conceptual vacuums about how to use computer technology. Such policy and conceptual vacuums are the marks of basic problems within computer ethics. Therefore, computer ethics is a field of substantial practical importance.

I find this argument for the practical value of computer ethics convincing. I think it shows that computer ethics is likely to have increasing application in our society. This argument does rest on a vision of the Computer Revolution which not everyone may share. Therefore, I will turn to another argument for the practical importance of computer ethics which doesn't depend upon any particular view of the Computer Revolution. This argument rests on the invisibility factor and suggests a number of ethical issues confronting computer ethics now.

The Invisibility Factor

There is an important fact about computers. Most of the time and under most conditions computer operations are invisible. One may be quite knowledgeable about the inputs and outputs of a computer and only dimly aware of the internal processing. This invisibility factor often generates policy vacuums about how to use computer technology. Here I will mention three kinds of invisibility which can have ethical significance.

The most obvious kind of invisibility which has ethical significance is invisible abuse. *Invisible abuse* is the intentional use of the invisible operations of a computer to engage in unethical conduct. A classic example of this is the case of a programmer who realized he could steal excess interest from a bank. When interest on an bank account is calculated, there is often a fraction of a cent left over after rounding off. This programmer instructed a computer to deposit these fractions of a cent to his own account. Although this is an ordinary case of stealing, it is relevant to computer ethics in that computer technology is essentialy involved and there is a question about what policy to institute in order to best detect and prevent such abuse. Without access to the program used for stealing the interest or to a sophisticated accounting program such an activity may easily go unnoticed.

Another possibility for invisible abuse is the invasion of the property and privacy of others. A computer can be programmed to contact another computer over phone lines and surreptitiously remove or alter confidential information. Sometimes an inexpensive computer and a telephone hookup is all it takes. A group of teenagers, who named themselves the "414s" after the Milwaukee telephone exchange, used their home computers to invade a New York hospital, a California bank, and a government nuclear weapons laboratory. These break-ins were done as pranks, but obviously such invasions can be done with malice and be difficult or impossible to detect.

A particularly insidious example of invisible abuse is the use of computers for surveillance. For instance, a company's central computer can monitor the work done on computer terminals far better and more discreetly than the most dedicated sweatshop manager. Also, computers can be programmed to monitor phone calls and electronic mail without giving any evidence of tampering. A Texas oil company, for example, was baffled why it was always outbid on leasing rights for Alaskan territory until it discovered another bidder was tapping its data transmission lines near its Alaskan computer terminal.

A second variety of the invisibility factor, which is more subtle and conceptually interesting than the first, is the presence of invisible programming values. *Invisible programming values* are those values which are embedded in a computer program.

Writing a computer program is like building a house. No matter how detailed the specifications may be, a builder must make numerous decisions about matters not specified in order to construct the house. Different houses are compatible with a given set of specifications. Similarly, a request for a computer program is made at a level of abstraction usually far removed from the details of the actual programming language. In order to implement a program which satisfies the specifications a programmer makes some value judgments about what is important and what is not. These values become embedded in the final product and may be invisible to someone who runs the program.

Consider, for example, computerized airline reservations. Many different programs could be written to produce a reservation service. American Airlines once promoted such a service called "SABRE". This program had a bias for American Airline flights built in so that sometimes an American Airline flight was suggested by the computer even if it was not the best flight available. Indeed, Braniff Airlines, which went into bankruptcy for awhile, sued American Airlines on the grounds that this kind of bias in the reservation service contributed to its financial difficulties.

Although the general use of a biased reservation service is ethically suspicious, a programmer of such a service may or may not be engaged in invisible abuse. There may be a difference between how a programmer intends a program to be used and how it is actually used. Moreover, even if one sets out to create a program for a completely unbiased reservation service, some value judgments are latent in the program because some choices have to be made about how the program operates. Are airlines listed in alphabetical order? Is more than one listed at a time? Are flights just before the time requested listed? For what period after the time requested are flights listed? Some answers, at least implicitly, have to be given to these questions when the program is written. Whatever answers are chosen will build certain values into the program.

Sometimes invisible programming values are so invisible that even the programmers are unaware of them. Programs may have bugs or may be based on implicit assumptions which don't become obvious until there is a crisis. For example, the operators of the ill-fated Three Mile Island nuclear power plant were trained on a computer which was programmed to simulate possible malfunctions including malfunctions which were dependent on other malfunctions. But, as the Kemeny Commission which investigated the disaster discovered, the simulator was not programmed to generate simultaneous, independent malfunctions. In the actual failure at Three Mile Island the operators were faced with exactly this situation – simultaneous, independent malfunctions. The inadequacy of the computer simulation was the result of a programming decision, as unconscious or implicit as that decision may have been. Shortly after the disaster the computer was reprogrammed to simulate situations like the one that did occur at Three Mile Island.

A third variety of the invisibility factor, which is perhaps the most disturbing, is ***invisible complex calculation***. Computers today are capable of enormous calculations beyond human comprehension. Even if a program is understood, it does not follow that the calculations based on that program are understood. Computers today perform and certainly supercomputers in the future will perform calculations which are too complex for human inspection and understanding.

An interesting example of such complex calculation occurred in 1976 when a computer worked on the four color conjecture. The four color problem, a puzzle mathematicians have worked on for over a century, is to show that a map can be colored with at most four colors so that no adjacent areas have the same color. Mathematicians at the University of Illinois broke the problem down into thousands of cases and programmed computers to consider them. After more than a thousand hours of computer time on various computers, the four color conjecture was proved correct. What is interesting about this mathematical proof, compared to traditional proofs, is that it is largely invisible. The general structure of the proof is known and found in the program and any particular part of the computer's activity can be examined, but practically speaking the calculations are too enormous for humans to examine them all.

The issue is how much we should trust a computer's invisible calculations. This becomes a significant ethical issue as the consequences grow in importance. For instance, computers are used by the military in making decisions about launching nuclear weapons. On the one hand, computers are fallible and there may not be time to confirm their assessment of the situation. On the other hand, making decisions about launching nuclear weapons without using computers may be even more fallible and more dangerous. What should be our policy about trusting invisible calculations?

A partial solution to the invisibility problem may lie with computers themselves. One of the strengths of computers is the ability to locate hidden information and display it. Computers can make the invisible visible. Information which is lost in a sea of data can be clearly revealed with the proper computer analysis. But that's the catch. We don't always know when, where, and how to direct the computer's attention.

The invisibility factor presents us with a dilemma. We are happy in one sense that the operations of a computer are invisible. We don't want to inspect every computerized transaction or program every step for ourselves or watch every computer calculation. In terms of efficiency the invisibility factor is a blessing. But it is just this invisibility that makes us vulnerable. We are open to invisible abuse or invisible programming of inappropriate values or invisible miscalculation. The challenge for computer ethics is to formulate policies which will help us deal with this dilemma. We must decide when to trust computers and when not to trust them. This is another reason why computer ethics is so important.

Dartmouth College
Hanover, NH 03755 USA