COMPUTERS, ETHICS, AND SOCIETY

SECOND EDITION

Edited by

M. David Ermann

Mary B. Williams

Michele S. Shauf

New York Oxford
OXFORD UNIVERSITY PRESS
1997

47. A. Speer, *Trial of the Major War Criminals*, International Military Tribunal, Nuremberg, Germany, 1947. (Quoted from [38].) vol. XVI.

48. L. M. Wooten, Albert Speer: how to manage an atrocity, *J. Hum. Psychol.* 21, 21–38 (1981).

49. D. Hayes, *Behind the Silicon Curtain: The Seductions of Work in a Lonely Era*, South End Press, Boston, 1989.

50. *Story of Women* (film, 1989). Advertisement.

51. R. K. Gilbert, The dynamics of inaction: psychological factors inhibiting arms control activism, *Am. Psychol.* 43, 755–764 (1989).

52. The corporate elite: the chief executives of the 1000 most valuable publicly held U.S. companies, *Business Week*, 55–274 (19 October 1990).

53. H. Rachlin, *Judgement, Decision and Choice: A Cognitive / Behavioural Synthesis*, W. H. Freeman and Co., New York, 1989.

54. B. S. Gorman, A. E. Wessman, G. R. Schmeidler, and S. Thayer, Linear representation of temporal location and Stevens' law, *Mem. Cognit.* 1, 169–171 (1973).

55. Canadian Armed Forces, The reserve: part-time adventure (advertisement), *The Tribune-Post*, Sackville, New Brunswick, Canada, Oct. 17, 1990, p. 17.

56. J. Platt, Social traps, *Am. Psychol.* 28, 641–651 (1973).

57. J. G. Cross and M. J. Guyer, *Social Traps*, University of Michigan Press, Ann Arbor, Michigan, 1980.

58. J. Baron, *Thinking and Deciding*, Cambridge University Press, New York, 1988.

59. G. R. Hardin, The tragedy of the commons, *Science* 162, 1243–1248 (1968).

60. A. Rappaport, Experiments with N-person social traps I, *J. Conflict Res.* 32, 457–472 (1989).

61. W. Mischel, Preference for delayed reinforcement: an experimental study of a cultural observation, *J. Abnormal Soc. Psychol.* 56, 57–61 (1958).

62. M. L. Commons, R. J. Herrnstein, and H. Rachlin, eds. *Quantitative Analyses of Behavior, Vol. II: Matching and Maximizing Accounts*, Ballinger Publishing, Cambridge, Massachusetts, 1982.

63. H. Rachlin, R. Battalio, J. Kagel, and L. Green, Maximization theory in behavioral psychology, *Behav. Brain Sci.* 4, 371–417 (1981).

64. R. J. Herrnstein, Rational choice theory: necessary but not sufficient, *Am. Psychol.* 45, 356–367 (1990).

65. P. Sanger, *Blind Faith: The Nuclear Industry in One Small Town*, McGraw-Hill Ryerson Ltd., Toronto, 1981.

66. C. Giangrande, *The Nuclear North: The People, The Regions and The Arms Race*, Anansi, Toronto, 1983.

67. A. G. Mojtabai, *Blessed Assurance: At Home with The Bomb in Amarillo, Texas*, Houghton-Mifflin, Boston, 1986.

31

People Are Responsible, Computers Are Not

The authors assert that people who use or design computer systems are morally responsible for any resulting harm. They discuss existing computer practices that increase the tendency for users and designers to feel little responsibility for harmful outcomes. To correct this problem the authors suggest alternative approaches to computer system design.

*Batya Friedman and Peter H. Kahn, Jr.**

Societal interest in responsible computing perhaps most often arises in response to harmful consequences that can result from computing. For instance, consider the frustration and economic loss incurred by individuals and businesses whose computer systems have been infected by the Internet worm or other computer viruses. Or consider the physical suffering and death of the cancer patients who were overradiated by Therac-25, or of civilians accidentally bombed in the Persian Gulf war by "smart" missiles gone astray. Largely in reaction to events like these, we have in recent years seen a surge of interest in preventing or at least minimizing such harmful consequences. But if responsible computing is to be understood as something more than a form of damage control, how are we to understand the term? Moreover, how can responsible computing be promoted within the computing community?

Design to Support Human Agency and Responsible Computing

[W]e propose that responsible computing often depends on humans' clear understanding that humans are capable of being moral agents and that computational systems are not. However . . . this understanding can be distorted in one of two ways. In the first type of distortion, the computational system diminishes or undermines the human user's sense of his or her own moral agency. In such systems, human users are placed into largely mechanical roles, either mentally or physically, and frequently have little understanding of the larger purpose or meaning of their individual actions.

*Reprinted by permission of the publisher from "Human Agency and Responsible Computing: Implications for Computer System Design," Batya Friedman and Peter H. Kahn, Jr., *Journal of Systems Software*, pp. 7-14. Copyright © 1992 by Elsevier Science Inc.

To the extent that humans experience a diminished sense of agency, human dignity is eroded and individuals may consider themselves to be largely unaccountable for the consequences of their computer use. Conversely, in the second type of distortion the computational system masquerades as an agent by projecting intentions, desires, and volition. To the extent that humans inapropriately attribute agency to such systems, humans may well consider the computational systems, at least in part, to be morally responsible for the effects of computer-mediated or computer-controlled actions.

Accordingly, to support humans' responsible use of computational systems, system design should strive to minimize both types of distortion. That is, system design should seek to protect the moral agency of humans and to discourage in humans a perception of moral agency in the computational system. How might design practices achieve these goals? Given that little research exists that addresses this question directly, we seek to provide some initial sketches by examining three types of computer practices.

Anthropomorphizing the Computational System

Anthropomorphic metaphors can be found in some of the definitions and goals for interface design. For example, some interfaces are designed to ''use the process of human-human communication as a model for human-computer interaction'' ([1], p. 86), to ''interact with the user similar to the way one human would interact with another'' ([1], p. 87), or to be ''intelligent'' where intelligence is based on a model of human intelligence. When such anthropomorphic metaphors become embedded in the design of a system, the system can fall prey to the second type of distortion by projecting human agency onto the computational system.

Moreover, even in unsophisticated designs of this type, there is some evidence that people do attribute agency to the computational system. For example, Weizenbaum [2] reported that some adults interacted with his computer program DOCTOR with great emotional depth and intimacy, ''conversing with the computer as if it were a person'' (p. 7). In a similar vein, some of the children Turkle [3] interviewed about their experiences with an interactive computer game called Merlin that played Tic-Tac-Toe attributed psychological (mental) characteristics to Merlin. For example, children sometimes accused Merlin of cheating, an accusation that includes a belief that the computer has both the intention and desire to deceive. In another example, Rumelhart and Norman [4] attempted to teach novices to use an editing program by telling the novices that the system was like a secretary. The novices drew on this human analogy to attribute aspects of a secretary's intelligence to the editing system and assumed (incorrectly) that the system would be able to understand whether they intended a particular string of characters to count as text or as commands.

While these examples of human attribution of agency to computational systems have largely benign consequences, this may not always be the case. Consider Jenkins' [5] human factors experiment that simulated a nuclear power plant failure. In the experiment, nuclear power plant operators had access to an expert system to aid them in responding to the plant failure. Although previously instructed on the expert system's limitations, ~~the ''operators expected that the expert system's limitations, the~~ ''operators expected that the expert system implemented in the computer 'knew' about the failures of the cooling system without being told. The system [however] was neither designed nor functioned as an automatic fault recognition system'' (p. 258). Jenkins attributed this overestimation of the system's capabilities to the power plant operators' expectations for the expert system to know certain information, presumably the type of information that any responsible human expert would know or attempt to find out in that situation.

Because nonanthropomorphic design does not encourage people to attribute agency to the computational system, such designs can better support responsible computing. To clarify what such design looks like in practice, consider the possibilities for interface design. Without ever impersonating human agency, interface design can appropriately pursue such goals as learnability, ease and pleasure of use, clarity, and quick recovery from errors. In addition, nonanthropomorphic interface design can employ such techniques as novel pointing devices, nonanthropomorphic analogies, speech input and output, and menu selection. Or consider the characteristics of another plausible technique: direct manipulation. According to Jacob [6], direct manipulation refers to a user interface in which the user ''seems to operate directly *on* the objects in the computer rather than carrying on a dialogue *about* them'' (p. 166). For example, the Xerox Star desktop manager adapted for systems such as the Apple Macintosh uses images of standard office objects (e.g., files, folders, and trash cans) and tasks to represent corresponding objects and functions in the editing system [7]. In this environment, disposing of a computer file is achieved by moving the image of the file onto the image of the trash can, akin to disposing of a paper file by physically placing the file in a trash can. There is no ambiguity in this direct manipulation interface as to who is doing the acting (the human user) and what the user is acting upon (objects in the computational system). The defining characteristics of direct manipulation suggest that this technique would not lead to projecting human agency onto the system. This is because direct manipulation involves physical action on an object as opposed to social interaction with an other as an undenying metaphor. Additionally, direct manipulation seeks to have the human user directly manipulate computational objects, thereby virtually eliminating the possibility for the human user to perceive the computer interface as an intermediary agent.

Nonanthropomorphic design considerations fit within a larger vision for interface design that is already part of the field. For example, Shneiderman [8] draws on Weizenbaum [2] to advocate design that "sharpen[s] the boundaries between people and computers . . . [for] human-human communication is a poor model for human-computer interaction" (p. 434). More recently, Shneiderman [9] writes that "when an interactive system is well designed, it almost disappears, enabling the users to concentrate on their work or pleasure" (p. 169). Winograd and Flores [10] similarly advocate the design of nonanthropomorphic computer tools that provide a transparent interaction between the user and the resulting action. "The transparency of interaction is of utmost importance in the design of tools, including computer systems, but it is not best achieved by attempting to mimic human faculties" (p. 194). When a transparent interaction is achieved, the user is freed from the details of using the tool to focus on the task at hand. The shared vision here is for the interface to "disappear," not to intercede in the guise of another "agent" between human users and the computational system.

Delegating Decision Making to Computational Systems

When delegating decision making to computational systems, both types of distortions can occur. The discussion that follows examines these distortions in the context of the APACHE system [11, 12]. More generally, however, similar analyses could be applied to other computer-based models and knowledge-based systems such as MYCIN [13] or the Authorizer's Assistant used by the American Express Corporation [14].

APACHE is a computer-based model implemented but not yet used clinically that determines when to withdraw life support systems from patients in intensive care units. Consider the nature of the human-computer relationship if APACHE, used as a closed-loop system, determines that life support systems should be withdrawn from a patient, and then turns off the life support systems. In ending the patient's life the APACHE system projects a view of itself to the medical personnel and the patient's family as a purposeful decision maker (the second type of distortion). Simultaneously, the system allows the attending physician and critical care staff to distance or numb themselves from the decision making process about when to end another human's life (the first type of distortion).

Now, in actuality, at least some of the researchers developing APACHE do not recommend its use as a closed-loop system, but as a consultation system, one that recommends a course of action to a human user who may or may not choose to follow the recommendation [11]. These researchers write: "Computer predictions should never dictate clinical decisions, as very often there are many factors other than physiologic data to be considered when a decision to withdraw therapy is made" (p. 1096). Thus, used as a consultation system, APACHE functions as a tool to aid the critical care staff with making difficult decisions about the withdrawal of therapy. Framed in this manner, the consultation system approach seems to avoid the distortions of human agency described above: the consultation system does not mimic purposeful action or inappropriately distance the medical staff from making decisions about human life and death.

In practice, however, the situation can be more complicated. Most human activity, including the decision by medical personnel to withdraw life support systems, occurs in a web of human relationships. In some circumstances, because a computational system is embedded in a complex social structure human users may experience a diminished sense of moral agency. Let us imagine, for instance, that APACHE is used as a consultation system. With increasing use and continued good performance by APACHE, it is likely that the medical personnel using APACHE will develop increased trust in APACHE's recommendations. Over time, these recommendations will carry increasingly greater authority within the medical community. Within this social context, it may become the practice for critical care staff to act on APACHE's recommendations somewhat automatically, and increasingly difficult for even an experienced physician to challenge the "authority" of APACHE's recommendation, since to challenge APACHE would be to challenge the medical community. But at this point the open-loop consultation system through the social context has become, in effect, a closed loop system wherein computer prediction dictates clinical decisions.

Such potential effects point to the need to design computational systems with an eye toward the larger social context, including long-term effects that may not become apparent until the technology is well situated in the social environment. Participatory design methods offer one such means [15, 16]. Future users, who are experienced in their respective fields, are substantively involved in the design process. As noted at a recent conference [17], Thoresen worked with hospital nurses to design a computer-based record-keeping system. In the design process, nurses helped to define on a macro level what institutional problems the technology would seek to solve, and on a micro level how such technological solutions would be implemented. From the perspective of human agency, such participatory design lays the groundwork for users to see themselves as responsible for shaping the system's design and use.

Delegating Instruction to Computational Systems

Instructional technology programs that deliver systematically designed computer-based courseware to students can suffer from the first type of distortion—computer use that erodes the human user's sense of his or her own agency. Often absent from this type of instructional technology is a meaningful notion of the student's responsibility for learning. Johnsen and Taylor [18] have discussed this problem in a paper aptly titled "At cross-purpose: instructional technology and the erosion of personal responsibility." According to Johnsen and Taylor, instructional technology "define[s] responsibility operationally in the context of means/ends rationality. The singular responsibility for a student's education becomes identified with the success of the program" (p. 9). They further point to the logical conclusion of this educational view for students, parents, teachers, and government: failure to educate comes to mean that the instructional technology failed to teach, not that students failed to learn.

As an example of this type of instructional technology, consider how the GREATERP intelligent tutoring system (described in [19]) for novice programmers in LISP handles students' errors. When GREATERP determines the student has entered "incorrect" information, the tutor interrrupts the tr/ student's progress toward the student's porposed solution (viable or not) and forces the student to backtrack to the intelligent tutor's "correct" solution. Thus GREATERP assumes responsibility not only for student learning but also for preventing student errors along the way and for the process of achieving a solution. In so doing, this intelligent tutoring system—and other comparable instructional technology programs—can undermine the student's sense of his or her own agency and responsibility for the educational endeavor.

In contrast, other educational uses of computing promote students' sense of agency and active decision making. For example, just as consultation systems can to some degree place responsibility for decision making on the human user, so educational uses of computer applications software (e.g., word processors, spreadsheets, data bases, microcomputer-based labs) can place responsibility for learning on the student. With computer applications students determine when the applications would be useful and for what purposes, and evaluate the results of their use. Moreover, the social organization of school computer use can contribute to students' understanding of responsible computing. As with participatory design, consider the value of student participation in creating the policies that govern their own school computer use. For example, as discussed in a recent article by Friedman [20], students can determine the privacy policy for their own electronic mail at school. To establish such a privacy policy, "students must draw on their fundamental understandings of privacy rights to develop specific policies for this new situation. In turn, circumstances like these provide opportunities for students not only to develop morally but to make decisions about a socially and computationally powerful technology, and thus to mitigate a belief held by many people that one is controlled by rather than in control of technology." Through such experiences, students can learn that humans determine how computer technology is used and that humans bear responsibility for the results of that use.

Conclusion

We argued initially that humans, but not computers (as they can be conceived today in material and structure), are or could be moral agents. Based on this view, we identified two broad approaches by which computer system design can promote responsible computer use. Each approach seeks to minimize a potential distortion between human agency and computer activity. First, computational systems should be designed in ways that do not denigrate the human user to machine-like status. Second, computational systems should be designed in ways that do not impersonate human agency by attemtping to mimic intentional states. Both approaches seek to promote the human user's autonomous decision making in ways that are responsive to and informed by community and culture.

What we have provided, of course, are only broad approaches and design sketches. But if we are correct that human agency is central to most endeavors that seek to understand and promote responsible computing, then increased attention should be given to how the human user perceives specific types of human-computer interactions, and how human agency is constrained, promoted, or otherwise affected by the larger social environment. In such investigations, it is likely that research methods can draw substantively on existing methods employed in the social-cognitive and moral-developmental psychological fields. Methods might include 1) semistructured hypothetical interviews with participants about centrally relevant problems [21–25]; 2) naturalistic and structured observations [26–28]; and 3) semistructured interviews based on observations of the participant's practice [29–31]. Of note, some anthropologists [32] and psychologists [33] working in the area of human factors have with some success incorporated aspects of these methods into their design practices.

A final word needs to be said about the role of moral psychology in the field of computer system design. As increasingly sophisticated computational systems have become embedded in social lives and societal practices, increasing pressure has been placed on the computing field to go beyond purely technical considerations and to promote responsible computing. In response, there has been, understandably, a desire to know the "right" answer to ethical problems that arise, where "right" is understood to mean something like "philosophically justified or grounded." We agree that there is an important place for philosophical analyses in he field. But philosophy seldom tells us how or why problems relevant to a philosophical position involving computing occur in practice, let alone what can most effectively resolve them. Such issues require empirical data that deal substantively with the psychological reality of humans. Thus, by linking our technical pursuits with both philosophical inquiry and moral-psychological research, responsible computing can be enhanced as a shared vision and practice within the computing community.

References

1. R. E. Eberts and C. G. Eberts, Four approaches to human computer interaction, in *Intelligent Interfaces: Theory, Research and Design* (P. A. Hancock and M. H. Chignell, eds.), Elsevier Science Publishers, New York, 1989.
2. J. Weizenbaum, *Computer Power and Human Reason,* W. H. Freeman & Company, New York, 1976.
3. S. Turkle, *The Second Self: Computers and the Human Spirit,* Simon & Schuster, New York, 1984.
4. D. E. Rumelhart and D. A. Norman, Analogical processes in learning, in *Cognitive Skills and Their Acquisition* (J. R. Anderson, ed.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.
5. J. P. Jenkins, An application of an expert system to problem solving in process control displays, in *Human-Computer Interaction* (G. Salvendy, ed.), Elsevier Science Publishers, New York, 1984.
6. R. J. K. Jacob, Direct manipulation in the intelligent interface, in *Intelligent Interfaces: Theory, Research and Design* (P. A. Hancock and M. H. Chignell, eds.), Elsevier Science Publishers, New York, 1989.
7. D. C. Smith, C. Irby, R. Kimball, W. Verplank, and E. Marslem, Designing the user interface, *Byte* 7, 242–282 (1982).
8. B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.
9. B. Shneiderman, Designing the user interface, in *Computers in the Human Context: Information Technology, Productivity, and People* (T. Forester, ed.), The MIT Press, Cambridge, Massachusetts, 1989.
10. T. Winograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design,* Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
11. R. W. S. Chang, B. Lee, S. Jacobs, and B. Lee, Accuracy of decisions to withdraw therapy in critically ill patients: clinical judgment versus a computer model, *Crit. Care Med.* 17, 1091–1097 (1989).
12. J. E. Zimmerman, ed., APACHE III study design: analytic plan for evaluation of severity and outcome, *Crit. Care Med.* 17 (Part 2 Suppl), S169–S221 (1989).
13. E. H. Shortliffe, Medical consultation systems: designing for doctors, in *Designing for Human-Computer Communication* (M. E. Sime and M. J. Coombs, eds.), Academic Press, New York, 1983.
14. C. L. Harris *et al.*, Office automation: making it pay off, in *Computers in the Human Context: Information Technology, Productivity, and People* (T. Forester, ed.), The MIT Press, Cambridge, Massachusetts, 1989.
15. P. Ehn, *Work-oriented Design of Computer Artifacts,* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1989.
16. J. Greenbaum and M. Kyng, eds., *Design at Work: Cooperative Design of Computer Systems,* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.
17. A. Namioka and D. Schuler, eds., *Proceedings from the Conference on Participatory Design 1990,* Computer Professionals for Social Responsibility, Palo Alto, California, 1990.
18. J. B. Johnsen and W. D. Taylor, At cross-purpose: instructional technology and the erosion of personal responsibility, paper presented at the annual meeting of the American Educational Research Association, New Orleans, April 1988.
19. R. Kass, Student modeling in intelligent tutoring systems—implications for user modeling, in *User Models in Dialog Systems* (A. Kobsa and W. Wahlster, eds.), Springer-Verlag, New York, 1989.
20. B. Friedman, Social and moral development through computer use: a constructivist approach, *J. Res. Comput. Educ.* 23: 560–567 (1991).
21. W. Damon, *The Social World of the Child,* Jossey-Bass, San Francisco, 1977.
22. L. Kohlberg, Stage and sequence: the cognitive-developmental approach to socialization, in *Handbook of Socialization Theory and Research* (D. A. Goslin, ed.), Rand-McNally, Chicago, 1969.
23. J. Piaget, *The Child's Conception of the World,* Routledge & Kegan Paul, London, 1929.
24. J. Piaget, *The Moral Judgement of the Child,* Routledge & Kegan Paul, London, 1932.
25. E. Turiel, *The Development of Social Knowledge: Morality and Convention,* Cambridge University Press, Cambridge, England, 1983.
26. R. DeVries and A. Goncu, Interpersonal relations in four-year dyads from constructivist and Montesorri programs, *J. Appl. Dev. Psychol.* 8, 481–501 (1987).
27. B. Friedman, Societal issues and school practices: An ethnographic investigation of the social context of school computer use, paper presented at the annual meeting of the American Educational Research Association, Boston, April 1990 (ERIC Document Reproduction Service No. ED 321 740).
28. L. P. Nucci and M. Nucci, Children's responses to moral and social conventional transgressions in free-play settings, *Child Dev.* 53, 1337–1342 (1982).
29. R. DeVries, Children's conceptions of shadow phenomena, *Gen. Soc. Gen. Psychol. Monographs* 112, 479–530 (1986).
30. L. P. Nucci and E. Turiel, Social interactions and the development of social concepts in preschool children, *Child Dev.* 49, 400–407 (1978).
31. G. B. Saxe, *Culture and Cognitive Development: Studies in Mathematical Understanding,* Lawrence Erlbaum Press, Hillsdale, New Jersey, 1990.
32. L. A. Suchman, *Plans and Situated Actions: The Problem of Human-Ma-*

chine Communication, Cambridge University Press, Cambridge, England, 1987.

33. C. Allen and R. Pea, Reciprocal evolution of research, work practices and technology, in *Proceedings from the Conference on Participatory Design 1990* (A. Namioka and D. Schuler, eds.), Computer Professionals for Social Responsibility, Palo Alto, 1990.

32

The Ten Commandments of Computer Ethics

These guidelines were created by the Computer Ethics Institute, which sponsors symposiums on computer ethics throughout the year. Contact the Institute at P. O. Box 42672, Washington, D.C. 20015 for more information. You may also join its listserv (*cei-l@american.edu*) by mailing a subscription command to *listserv@american.edu*.

*Computer Ethics Institute**

The Ten Commandments of Computer Ethics

1. Thou shalt not use a computer to harm other people.
2. Thou shalt not interfere with other people's computer work.
3. Thou shalt not snoop around in other people's computer files.
4. Thou shalt not use a computer to steal.
5. Thou shalt not use a computer to bear false witness.
6. Thou shalt not copy or use proprietary software for which you have not paid.
7. Thou shalt not use other people's computer resorces without authorization or proper compensation.

*Copyright © by the Computer Ethics Institute. Reprinted by permission.

8. Thou shalt not appropriate other people's intellectual output.
9. Thou shalt think about the social consequences of the program you are writing or the system you are designing.
10. Thou shalt always use a computer in ways that insure consideration and respect for your fellow humans.

33

ACM Code of Ethics and Professional Conduct

On October 16, 1992, the Executive Council of the Association for Computing Machinery voted to adopt the following revised Code of Ethics. The Code contains twenty-four imperatives that define the personal responsibilities of computing professionals.

*Association for Computing Machinery (ACM)**

Commitment to ethical professional conduct is expected of every voting, associate, and student member of ACM. This Code, consisting of 24 imperatives formulated as statements of personal responsibility, identifies the elements of such a commitment.

It contains many, but not all, issues professionals are likely to face. Section 1 outlines fundamental ethical considerations, while Section 2 addresses additional, more specific considerations of professional conduct. Statements in Section 3 pertain more specifically to individuals who have a leadership role, whether in the workplace or in a volunteer capacity, for example with organizations such as ACM. Principles involving compliance with this Code are given in Section 4.

The Code is supplemented by a set of Guidelines, which provide explanation to assist members in dealing with the various issues contained in the Code. It is expected that the Guidelines will be changed more frequently than the Code.

The Code and its supplemented Guidelines are intended to serve as a basis for ethical decision making in the conduct of professional work. Secondarily, they may serve as a basis for judging the merit of a formal complaint pertaining to violation or professional ethical standards.

*Copyright © Association for Computing Machinery. Reprinted by permission.

It should be noted that although computing is not mentioned in the moral imperatives section, the Code is concerned with how these fundamental imperatives apply to one's conduct as a computing professional. These imperatives are expressed in a general form to emphasize that ethical principles which apply to computer ethics are derived from more general ethical principles.

It is understood that some words and phrases in a code of ethics are subject to varying interpretations, and that any ethical principle may conflict with other ethical principles in specific situations. Questions related to ethical conflicts can best be answered by thoughtful consideration of fundamental principles, rather than reliance on detailed regulations.

1. General Moral Imperatives

As an ACM member I will . . .

1.1 Contribute to society and human well-being

This principle concerning the quality of life of all people affirms an obligation to protect fundamental human rights and to respect the diversity of all cultures. An essential aim of computing professionals is to minimize negative consequences of computing systems, including threats to health and safety. When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare.

In addition to a safe social environment, human well-being includes a safe natural environment. Therefore, computing professionals who design and develop systems must be alert to, and make others aware of, any potential damage to the local or global environment.

1.2 Avoid harm to others

"Harm" means injury or negative consequences, such as undesirable loss of information, loss of property, property damage, or unwanted environmental impacts. This principle prohibits use of computing technology in ways that result in hamr to any of the following: users, the general public, employees, employers. Harmful actions include intentional destruction or modification of files and programs leading to serious loss of resources or unnecessary expenditure of human resources such as the time and effort required to purge systems of computer viruses.

Well-intended actions, including those that accomplish assigned duties, may lead to harm unexpectedly. In such an event the responsible person or