International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 11 Issue 4 –NOVEMBER 2014.

UNIQUE ETHICAL PROBLEMS IN INFORMATION TECHNOLOGY

PSR KRISHNA

Lecturer in computer science, KBN College

Psrkrishna04@gmail.com

Abstract— A distinction is made between moral indoctrination and instruction in ethics. It is argued that the legitimate and important field of computer ethics should not be permitted to become mere moral indoctrination. Computer ethics is an academic field in its own right with unique ethical issues that would not have existed if computer technology had not been invented. Several example issues are presented to illustrate this point.

The failure to find satisfactory non computer analogies testifies to the uniqueness of computer ethics. Lack of an effective analogy forces us to discover new moral values, formulate new moral principles, develop new policies, and find new ways to think about the issues presented to us. For all of these reasons, the kind of issues presented deserves to be addressed separately from others that might at first appear similar. At the very least, they have been so transformed by computing technology that their altered form demands special attention.

I. INTRODUCTION

One factor behind the rise of computer ethics is the lingering suspicion that computer professionals may be unprepared to deal effectively with the ethical issues that arise in their workplace. Over the years, this suspicion has been reinforced by mostly anecdotal research that seems to show that computer professionals simply do not recognize when ethical issues are present. Perhaps the earliest work of this kind was done by Donn Parker in the late 1970s at SRI International.1

In 1977, Parker invited highly trained professionals from various fields to evaluate the ethical content of 47 simple hypothetical cases that he had created based in part on his expert knowledge of computer abuse. Workshop participants focused on each action or non-action of each person who played a role in these one-page scenarios. For each act that was performed or not performed, their set task was to determine whether the behaviour was unethical or not, or simply raised no ethics issue at all. Parker found a surprising amount of residual disagreement among these professionals even after an exhaustive analysis and discussion of all the issues each case presented. More surprisingly, a significant minority of professionals held to their belief that no ethics issue was present even in cases of apparent computer abuse. For example, in Scenario 3.1, a company representative routinely receives copies of the computerized arrest records for new company employees. These records are provided as a favour by a police file clerk who happens to have access to various local and federal databases containing criminal justice information nine of the 33 individuals who analysed this case thought disclosure of arrest histories raised no ethics issues at all. Parker's research does not identify the professions represented by those who failed to detect ethics issues, but most of the participants in this early. This left casual readers of Parker's Ethical Conflicts in Computer Science and Technology free to identify computer professionals as the ones who lacked ethical sensitivity. If some of them could not even recognize when ethical issues were present, it is hard to imagine how they could ever hope to deal responsibly with them. According to Parker, the problem may have been fostered by computer education and training programs that encouraged, or at least failed to criminalize, certain types of unethical professional conduct. This perception of professional inadequacy is part of a largely hidden political agenda that has contributed to the development of various curricula in computer ethics. In recent years, the tacit perception that those preparing for careers in computing may need remedial moral education seems to have influenced some accreditation boards. As a result, they have been willing to mandate more and more ethical content in computer science and computer engineering programs. They may also be responding to the increased media attention given to instances of computer abuse, fraud and crime. Others demand more ethical content because they believe that catastrophic failures of computer programs are directly attributable to immoral behaviour.

The growth of interest is gratifying, especially considering that, in 1976, I found it hard to convince anyone that "computer ethics" was anything other than an oxymoron.5 No doubt Norbert Weiner would be pleased to see his work bearing late fruit.6 At the same time, I am greatly disturbed when courses in social impact and computer ethics become a tool for indoctrination in appropriate standards of professional conduct. Donald Gotterbarn, for example, argues that one of the six goals of computer ethics is the "socialization" of students into "professional norms." The fact that these norms are often eminently reasonable, even recommended thoughtfully to us by our professional organizations, does not make indoctrination any less repugnant. The goal cannot be simply to criminalize or stigmatize departures from professional norms. Consider an analogy.

1. That certain ethical issues are so transformed by the use of computers that they deserve to be studied on their own, in their radically altered form,

2.That the involvement of computers in human conduct can create entirely new ethical issues, unique to computing, that do not surface in other areas.

I shall refer to the first as the "weaker view" and the second as the "stronger view."

Although the weaker view provides sufficient rationale, most of my attention will be focused on establishing the stronger view. This is similar to the position I took in, except that I no longer believe that problems merely aggravated by computer technology deserve special status.

II. LEVELS OF JUSTIFICATION FOR THE STUDY OF COMPUTER ETHICS

From weaker to stronger, there are at least six levels of justification for the study of computer ethics. Level One We should study computer ethics because doing so will make us behave like responsible professionals. At worst, this type of rationale is a disguised call for moral indoctrination. At best, it is weakened by the need to rely on an elusive connection between right knowledge and right conduct. This is similar to the claim that we should study religion because that will cause us to become more spiritual. For some people, perhaps it may, but the mechanism is not reliable. Level TwoWe should study computer ethics because doing so will teach us how to avoid computer abuse and catastrophes.

Reports by Parker, Neumann, Forester and Morrison leave little doubt that computer use has led to significant abuse, hijinks, crime, near catastrophes, and actual catastrophes. The question is: Do we get a balanced view of social responsibility merely by examining the profession's dirty laundry? Granted, a litany of computer "horror stories" does provide a vehicle for infusing some ethical content into the study of computer science and computer engineering.

Granted, we should all work to prevent computer catastrophes. Even so, there are major problems with the use of conceptual shock therapy:

The cases commonly used raise issues of bad conduct rather than good conduct. They tell us what behaviours to avoid but do not tell us what behaviours are worth modelling.As Leon Tabak has argued, this approach may harm students by preventing them from developing a healthy, positive and constructive view of their profession.

Persons who use computers for abusive purposes are likely to be morally bankrupt. There is little we can learn from them.

Many computer catastrophes are the result of unintended actions and, as such, offer little guidance in organizing purposive behaviour.

Level Three

We should study computer ethics because the advance of computing technology will continue to create temporary policy vacuums. Long-term use of poorly designed computer keyboards, for example, exposes clerical workers to painful, chronic, and eventually debilitating repetitive stress injury. Clearly employers should not require workers to use equipment that will likely cause them serious injury. The question is: What policies should we formulate to address problems of long-term keyboard use.

New telephone technology for automatic caller identification creates a similar policy vacuum. It is not immediately obvious what the telephone company should be required to do, if anything, to protect the privacy of callers who wish to remain anonymous.

Unlike the first- and second-level justifications I have considered and rejected, this third-level justification does appear to be sufficient to establish computer ethics as an important and independent discipline. Still, there are problems: Since policy vacuums are temporary and computer technologies evolve rapidly, anyone who studies computer ethics would have the perpetual task of tracking a fast moving and ever-changing target. It is also possible that practical ethical issues arise mainly when policy frameworks clash. We could not resolve such issues merely by formulating more policy.

Level Four

We should study computer ethics because the use of computing permanently transforms certain ethical issues to the degree that their alterations require independent study .I would argue, for example, that many of the issues surrounding intellectual property have been radically and permanently altered by the simple question, "What do I own?" has been transformed into the question, "What exactly is it that I own when I own something?" Likewise, the availability of cheap, fast, painless, transparent encryption technology has completely transformed the privacy debate. In the past, we worried about the erosion of privacy now we worry about the impenetrable wall of computer- generated privacy afforded to every criminal with a computer and half a brain.

Level Five

We should study computer ethics because the use of computing technology creates, and will continue to create, novel ethical issues that require special study.

I will return to this topic in a moment.

Level Six

We should study computer ethics because the set of novel and transformed issues is large enough and coherent enough to define a new field. I mention this hopefully as a theoretical possibility. Frankly, after fifteen years, we have not been able to assemble a critical mass of self-defining core issues Joseph Behar, a sociologist, finds computer ethics diffuse and unfocused. Gary Chapman, when he spoke to the Computers and Quality of Life Conference in 1990, complained that no advances had been made in computer ethics. There are various explanations for this apparent (or real) lack of progress:

Computer ethics is barely fifteen years old. Much of its intellectual geography remains uncharted.

So far, no one has provided a complete and coherent concept of the proper subject matter for computer ethics.

We have wrongly included in the domain of computer ethics any unethical act that happened to involve a computer. In the future, we must be more careful to restrict ourselves to those few acts where computers have an essential as opposed to incidental involvement.

Because computer ethics is tied to an evolving technology, the field changes whenever the technology changes. For example, the use of networked computers presents moral problems different from those presented by the use of standalone computers. The use of mouse-driven interfaces raises issues different from those raised by keyboard driven interfaces, particularly for people who are blind.

We adopted, from clever philosophers, the dubious practice of using highly contrived, two-sided, dilemmatic cases to expose interesting but irresolvable ethical conflicts. This led to the false perception that there could be no progress and no commonality in computer ethics. New research may cause this perception to fade

We have remained focused for too long on the dirty laundry of our profession.

On a hopeful note, the Impacts Steering Committee chaired by C. Dianne Martin is halfway through a three-year NSFfunded project that will likely generate a highly coherent picture of how the computer science curriculumcan address social and ethical issues.Impact CS intends to publish specific curriculum guidelines along with concrete models for implementing them.

III. THE SPECIAL STATUS OF COMPUTER ETHICS

I now turn to the task of justifying computer ethics at Level 5 by establishing, through several examples, that there are issues and problems unique to the field. By "unique" I mean to refer to those ethical issues and problems that are characterized by the primary and essential involvement of computer technology, exploit some unique property of that technology, and would not have arisen without the essential involvement of computing technology I mean to allow room to make either a strong or a weak claim as appropriate. For some examples, I make the strong claim that the issue or problem would not have arisen at all. For other examples, I claim only that the issue or problem would not have arisen in its present, highly altered form.

To establish the essential involvement of computing technology, I will argue that these issues and problems have no satisfactory non-computer moral analogy. For my purposes, a "satisfactory" analogy is one that (a) is based on the use of a machine other than a computing machine and (b) allows the ready transfer of moral intuitions from the analog case to the case in question. In broad strokes, my line of argument will be that certain issues and problems are unique to computer ethics because they raise ethical questions that depend on some unique property of prevailing computer technology. My remarks are meant to apply to discrete-state stored program inter-networking fixed instruction-set serial machines of von Neumann architecture. It is possible that other designs (such as the Connection Machine) would exhibit a different set of unique properties.

Next I offer a series of examples, starting with a simple case that allows me to illustrate my general approach.

EXAMPLE 1: Uniquely Stored

One of the unique properties of computers is that they must store integers in "words" of a fixed size. Because of this restriction, the largest integer that can be stored in a 16-bit computer word is 32,767. If we insist on an exact representation of a number larger than this, an "overflow" will occur with the result that the value stored in the word becomes corrupted. This can produce interesting and harmful consequences. For example, a hospital computer system in Washington, D.C., broke down on September 19, 1989, because its calendar calculations counted the days elapsed since January 1, 1900. On the 19th of September, exactly 32,768 days had elapsed, overflowing the 16-bit word used to store the counter, resulting in a collapse of the entire system and forcing a lengthy period of manual operation. At the Bank of New York, a similar 16-bit counter overflowed, resulting in a $32 billion overdraft. The bank had to borrow $24 million for one day to cover the overdraft. The interest on this one-day loan cost the bank about $5 million. In addition, while technicians attempted to diagnose the source of the problem, customers experienced costly delays in their financial transactions.

EXAMPLE 2: Uniquely Malleable

Another unique characteristic of computing machines is that they are very general- purpose machines.

As James Moor observed, they are "logically malleable" in the sense that "they can be shaped and moulded to do any activity that can be characterized in terms of inputs, outputs, and connecting logical operations." The Stats knows he needs a personal computer, what he calls "cleats for the mind." He also knows that he needs to be able to operate that computer without being able to move anything below his neck.

EXAMPLE 3: Uniquely Complex

Another unique property of computer technology is its superhuman complexity. It is true that humans program computing machines, so in that sense we are masters of the machine. The problem is that our programming tools allow us to create discrete functions of arbitrary complexity. In many cases, the result is a program whose total behaviour cannot be described by any compact function Buggy programs in particular are notorious for evading compact description!

IV. CONCLUSION

I have tried to show that there are issues and problems that are unique to computer ethics. For all of these issues, there was an essential involvement of computing technology. Except for this technology, these issues would not have arisen or would not have arisen in their highly altered form. The failure to find satisfactory non-computer analogies testifies to the uniqueness of these issues. The lack of an adequate analogy, in turn, has interesting moral consequences. Normally, when we confront unfamiliar ethical problems, we use analogies to build conceptual bridges to similar situations we have encountered in the past. Then we try to transfer moral intuitions across the bridge, from the analog case to our current situation. Lack of an effective analogy forces us to discover new moral values, formulate new moral principles, develop new policies, and find new ways to think about the issues presented to us. For all of these reasons, the kind of issues I have been illustrating deserves to be addressed separately from others that might at first appear similar.

V. REFERENCES

[1] Parker, D. Ethical Conflicts in Computer Science and Technology. SRI International, Menlo Park, California, 1978.

[2] 2 There was a follow-up study some years later that remedied some of the problems discovered in the original methodology.

[3] Parker, D., Swope, S., and Baker, B., Ethical Conflicts in Information and Computer Science, Technology, and Business.QED Information Sciences, Inc., Wellesley, Massachusetts, 1990