2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34347

Computing Ethics for the Ethics of Computing

Dr. Robin K. Hill, University of Wyoming

Dr. Hill is an adjunct professor in both the Wyoming Institute for Humanities Research and the Philosophy Department of the University of Wyoming, and a Lecturer in Computer Science. She currently writes a blog on the philosophy of computer science for the online Communications of the ACM. Her teaching experience includes logic, computer science, and information systems courses for the University of Wyoming, University of Maryland University College (European Division), State University of New York at Binghamton, Metropolitan State College, and others. She holds a Ph.D. in Computer Science, an M.S. in Management Information Systems, an M.A. in Mathematical Logic, and a B.A. in Philosophy.

Computing Ethics for the Ethics of Computing

May 15, 2021

Abstract

In an undergraduate computing ethics course, computing analogues can assist in illustrating and grounding some of the content of professional ethics for computer science itself. To introduce students to the standard normative theories, the instructor gives function headings; to show the different ways that these normative theories can be play out in reality, she describes their inheritance mechanism; to motivate gathering of pertinent facts, she invokes the notion of metadata. Care must be taken to emphasize that morals cannot be mechanized, but that such analogies can serve among the many factors that help in the understanding and solution of professional ethical dilemmas.

Introduction

It won't come as a surprise to the many instructors newly pressed into covering such courses that the ethics of computing poses pedagogical challenges. The student viewing a dazzling spread of career opportunities may view a course on dry moral subjects, rooted in the humanities, with little enthisiasm and some disdain. Administrators may support the course grudgingly. Colleagues may grant only superficial attention and respect to the subject matter. I myself have expressed skepticism about the whole endeavor. Plenty of calls for computing ethics pop up in hand-wringing commentaries on modern technology without striking the observer as conscientious commitment with constructive suggestions.

Yet this author is now teaching "Ethics for the Computing Professional," and the experience has been fulfilling and fruitful—on both sides, she hopes. An undergraduate background in philosophy helps; she sympathizes whole-heartedly with any teacher thrown into this without familiarity with humanistic and philosophical methods. The goal here is to provide some useful suggestions in the form of rough analogues between programming phenomena and aspects of normative theories.

The Scene

The Course

Our class in "Ethics for the Computing Professional," a one-credit required course for computer science majors, is offered at the junior level and most often taken by seniors, and most often fully enrolled. It aims to bring moral philosophy to the budding computer scientist. The course

syllabus covers the usual subjects of privacy, security, intellectual property, cyberabuse and other social detriments, and professional codes, but also reaches farther into philosophy, to equip students with the vocabularly and methods of the humanities. We use Herman Tavani [1] as the nominal textbook. (I designate an older edition so that copies may be purchased cheaply, and also provide extracts, compliant with Fair Use, in Course Reserves at our library.)

Most of the material draws from contemporary journalism, case studies, and thought exercises. We strive to find and analyze ethical quandaries in computing; to factor out related phenomena that manifest *non-ethical* quandaries (commerce, etiquette, psychology, and so forth); to identify moral agents and patients in cases; and to select and apply principles (normative theories or other guidelines) to ethical quandaries. Note that although this paper discusses the standard families of normative theories that a student would hear about in a philosophy course, those theories consume only a small proportion of class time.

Normative Theories

The start of the course presents the following common approaches, prominent in contemporary discourse. Standard ethical theories prove indispensable in teaching the idea of principles in general. Among many reasonable ways to explain ethical theories, the brief accounts for this class are given below. For another good list (along with other materials), see the Markkula Center website [2]. The explication emphasizes that these are simplistic accounts of *families* of theories, which must be further specified for application.

Consequentialism: Theories that determine right and wrong based on the outcomes of decisions and conduct. These include Utilitarianism, where the outcomes are measures of good and bad, and "right" is the greatest good for the greatest number.

Deontology: Theories that determine right and wrong by adherence to duty. This includes the Categorical Imperative, under which we choose the conduct that we wish to become a maxim.

Virtue Ethics: Theories that determine right and wrong by reference the dispositions inherent in good character.

Social Contract: Theories that determine right and wrong via reference to cultural practices.

We also take a look at **Supernaturalism** (religion) and **Intuitionism** (trust your instinct), for comparison and contrast.

All six of these standard theories are described informally along with their deficits, that is, their failures to generate reasonable guidance in particular situations. To give a students a better sense of how these normative theories work, I invoke some concepts from computing.

Computing Analogies

These analogies might be better described as conceptual assistance for understanding theories, as their role is instrumental. Such descriptions must be carefully qualified with the warning that

ethics is not actually computable; no mechanical procedure exists for making ethical decisions.

Normative Theories as Subroutine Headings

Ethical theories sketched as function headings outlne high-level decision procedures suitable for carrying out by a human agent, with input and output parameters (given as types). An example of a function heading is the description of Consequentialism ("right and wrong depend on outcomes") as a subroutine that produces `action-scores` as output, given as input both an ethical quandary Q and some knowledge of how the world works (to enable the computation of outcomes):

```
action-scores Consequentialism (Q,causal-knowledge);
```

But, as noted, the students can't take that as implying an algorithm. Leaving the implementation as a glossed heading in a very general pseudocode is a signal of that lack of precision.

There are many ways to write these function headings, of course. Here are more examples for other normative theory families.

```
ranking-under-duty Deontology (Q,rules-of-duty);
assessment-by-good-person Virtue-Ethics (Q,model-character);
accepted-decisions Social-Contract (Q,expectations-and-traditions);
action-to-take Intuitionism (Q);
```

Note that this paradigm efficiently describes Intuitionism, for which the description given to the class is something along the lines of "the theory for those who think the theories are bogus, because you already know what to do"; there is no input other than the current quandary Q, so `action-to-take` is determined by instinctive reaction only. And it should be clear that although this level of specification differentiates the functions, the degree of abstraction remains high. The functions are underdetermined, leaving the analogies informal, more suggestive than directive.

The pedagogical goals of this technique are (1) to expose what must be *given* as input in a normative procedure, and what's *produced* as output, and (2) to show that the major families of theories have different types, as defined by their function headings; they are not substitutable for one another.

Inheritance and Implementation

A theory family is a base class that must be further instantiated to provide actionable norms. The descriptions of theory families above require further particularization, as noted, which is, of course, specification of derived classes. For the base class Supernaturalism, we must designate a particular religion before we have an normative theory that can be applied. For the base class Consequentialism, if we instantiate the measurement of outcomes as the greatest good for the greatest number, we have the normative theory Utilitarianism. The Categorical Imperative inherits from Deontology. And so forth.

This comparison also invites discussion of which variables should be treated as code libraries and which as parameters. In Virtue-Ethics, we ask whether there is one `model-character` (which could be implemented as a library); or do we refer to different model characters (passed as parameters) on different occasions? This offers a path (optional!) to consideration of questions about exemplars and reification of virtues. While it may seem desirable to express this class inheritance analogy in more specific and rigorous pseudocode, this instructor finds that the object-oriented language details necessary (for any programming language) obscure rather than illuminate the concept.

Normative Theories as Subroutines

And what about the function *bodies*? We address this, depending on how the course develops, to verify student understanding. The ethical quandary Q must be analyzed into a compound variable of several factors. Utilitarianism, for example, can be "coded" as nested loops:

```
action-scores Utilitarianism (Q,causal-knowledge);
  for each factor in Q:
    find possible solutions;
    for each solution,
      Compute the harms and benefits for each moral patient (using causal-
          knowledge),
      Ranking each solution by amount of good for number of moral patients,
      Adding the results to a matrix of the return type decision-scores.
```

The pedagogical goals are twofold, with some tension to discuss: (1) To verify understanding of the standard theories; (2) To show that the computations necessary are not plausibly undertaken by an algorithm. In other words, it seems that we humans refer to such a procedure, but only for rough guidance as we take jumps and shortcuts.

Processes and Data

Here follow other comparisons between ethical and computational phenomena, also left vague, that might be useful as comments in passing.

The Algorithmic Platform

Families of ethical theories (grossly simplified) can be mapped to certain "intelligent" programming methods. We can interpret Consequentialist theories as simple calculations, compared to Deontologies as expert systems, which work work by searching for factors from Q among antecedents, and discharging consequents on matches. Compare to Virtue Ethics as a deep-learning approach that can deliver assessments but cannot produce a justifying trace or chain of reasoning. The pedagogical goal of these references to paradigms of artificial intelligence is deepening the student's appreciation of the diversity of problem-solving approaches in ethics.

Data and Metadata

Metadata on information helps to analyze cases of authority and reliability. What would we consider to be decisive in terms of last wishes (or plausible threats), for example, on a website or a tweet? In a dataset, it might be the date, the source, the format, presence of a signature or notarization, and so forth. Analogously, in the real world, in might be the date (also), the source (also), and the presence of authoritative verification (also). The pedagogical goal is exposure to the factors on which we rely to support ethical reasoning.

Caveats and Questions

All of these analogies are based on gross simplifications of concepts from both philosophical ethics and the foundations of computing. Great care must to taken to draw students away from a literal interpretation. That care itself should be overtly modeled, noted, articulated, to cultivate an appreciation of humanistic methods. The pedagogical goal is demonstration that *you*, the computer science student, can respect and pursue ethics along with technology, although the methods and milieus differ.

Presentation of the standard theories, as outlined herein, takes only one or two of 15 meetings. Again, these techniques are the tools deployed to ground the teaching of ethics in computing, but not the subject of that teaching. We move on to study the usual concerns and questions, examining privacy versus security through facial recognition, the digital divide through technology's benefits and harms to children, business models through patent trolls, virual-world behavior through the Gamer's Dilemma, the hacking arms race through government stockpiling of zero-day exploits, and other timely issues.

This instructor regrets that she has no research to indicate whether this pedagogical technique is successful to any degree. But she doubts that such an experiment can be conducted according to scientific protocols, except by teaching courses that are equivalent except for the computational-analogy content, a scenario not available to this department.

These analogies could serve as a platform for larger philosophical inquiries, offering provocative questions about the relationship between (1) general cognition as we apply it to knotty moral problems, and (2) the computational paradigms that continue to emerge as research and development progresses in technology. (Is Virtue Ethics really like learning with a deep neural network that settles on some ideal character? What would that "like" even mean?)

Conclusion

The standard ethical theories, along with other basic moral concepts such as agents and patients, give the class the vocabulary with which to pursue constructive thought and discussion on modern issues of professional ethics in computer science. Used judiciously, with emphasis on the limitations of the comparison, computing analogies can assist computer science students in their understanding, analysis, and resolution of ethical problems they face now and in the future.

References

[1] H. J. Tavani, *Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing*, 4th ed. Wiley, 2013.

[2] Markkula Center Contributors. A framework for ethical decision making. https://www.scu.edu/ethics/ethics-resources/ethical-decision-making/a-framework-for-ethical-decision-making/. Markkula Center for Applied Ethics at Santa Clara University.