Week 8: Computer ethics

Felix Pinkert, F.Pinkert@warwick.ac.uk

PH212: Applied Ethics
University of Warwick

1 Is there a computer ethics?

1.1 Some themes in computer ethics

- Self-driving cars.
- Robots replacing humans.
- Open and closed source software.
- Computer privacy and surveillance.
- Social networking.
- Virtual reality.
- Remote controlled weapons / drones.
- Lethal autonomous weapons systems a.k.a. killer robots.
- Artificial intelligence (and possibly the end of the human species).

1.2 The uniqueness thesis

- The uniqueness thesis Actions involving computers pose novel and unique ethical questions that require new arguments and discussion.
- Denying uniqueness Actions involving computers simply pose old ethical questions in a new context, and do not require any new kinds of reasoning or argument.

1.3 What is unique about computers?

- Computers allow us to do new things, and to do old things in new ways.
- Doing old things in new ways need not pose new ethical issues, though legal regulation (if any) may have to be adapted:
  - e.g. online defamation vs. defamation of someone in a print newspaper.
- Doing old things in new ways can allows us to do things to a different extent, and this can change the character of what we do and its ethical evaluation:
  - e.g. getting information about someone you just met: can be much more comprehensive by means of computers than offline.

1.4 In support of uniqueness: Computers and new things

- Computers introduce new act-types, relations, and objects into our world.
  - Algorithms.
  - Software.
  - Data.
  - "Entering" a computer system.
  - Unleashing malware.
- The above require new kinds of ethical thinking.

1.5 Against total uniqueness: Not all is new

- We cannot identify an ethical issue with a new technology if there is no similarity at all to familiar, ethically significant actions.
- For example:
  - Unleashing a computer virus: Bringing about harm.
  - Hacking into personal data: Accessing information the person intends to keep private.
  - Piracy: Failing to contribute to the cost of what someone else produced, while benefiting from it, without their consent.
- Deborah Johnson's conclusion: Computers bring about new species of ethical situations, but these are part of familiar *genera* (pl. of *genus*, i.e. higher-order categories), of ethical situations. A completely distinct novel ethics is simply not possible.
- Rest of the lecture: Examine varying kinds of uniqueness in practice.

2 Replacing humans with computers

2.1 Some pictures

2.2 A new kind of situation

- Computers allow us to replace more human labour and contribution with machines.
- Common themes to other technology:
  - Allows us to replace some very simple human labour.
  - Increases the efficiency of human labour and reduces demand for human workers.
- New themes / degree:
  - Allows us to replace a much more complicated, and possibly more paradigmatically human, actions with machines.
    - e.g. creating culture, doing science, caring for the sick or elderly, having a conversation.
  - Call this "computerized machine replacement".

Technology and jobs

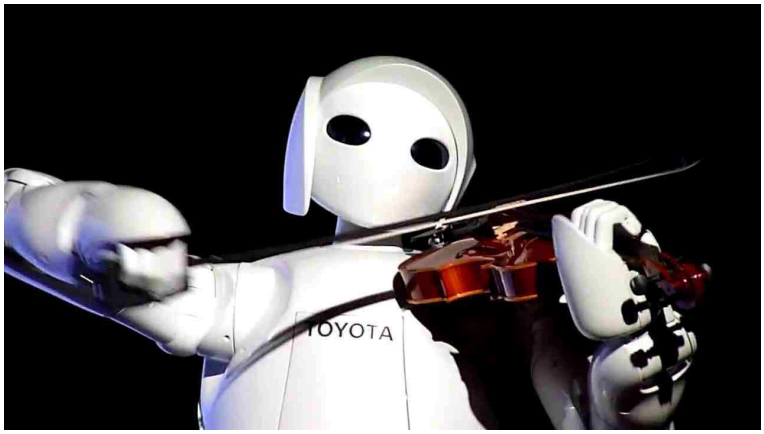
Coming to an office near you

The effect of today's technology on tomorrow's jobs will be immense—and no country is ready for it

Jan 18th 2014 | From the print edition

Timekeeper

Tweet



2.3 Ethical questions

- Using only categories of permissible/impermissible is too coarse and not helpful.
- Instead: Enter value theory:
  - Do we lose anything by letting computers and computer-run robots replace humans in certain activities?
    - Does the human contribution to certain activities constitute part of the value to the activities' human parties/recipients?
    - Is the very act of contributing of value to the human who contributes?
- Ethical significance: The answer to these questions feeds into the consideration of what we lose and gain through technology.
  - These losses and gains matter prudentially: when they affect ourselves as users and beneficiaries, and
  - ethically: when they affect others, e.g. those whose contribution is made redundant, and when these costs are forced upon us.
- Underlying individual, ethical, and social question: Which technologies to use, and how?
  - cf. Neal Stephenson's "Seveneves", part two: "Amistics, the study of the choices made by different cultures as to which technologies they would embrace or spurn."
  - Often, we are not aware that we collectively and individually use technology as a matter of choice. Literature and film can help our imagination here and make us aware of choices and options we have.
  - Sci-Fi examples: lack/limitations of computers in *Battlestar Galactica* or *Dune*; animals, fireweapons, and cooking in *Firefly*.
  - Real-world examples: sitting in this lecture, taking handwritten notes, writing letters, meeting your friends in person,
- To make informed choices, including tradeoffs, we need to know what we gain and lose from technological innovation and mediating or replacing human actions via machines.

2.4 Do we lose by being benefited by machines rather than humans?

- Candidate goods / services / interactions in which human contribution might matter:
  - Material goods: food, clothes, machines, houses.
  - Cultural artifacts: Paintings, photographs, music, literature, film.
  - Services: physical/medical care, teaching, restaurant service, taxi driving, supermarket checkout.
  - Interactions: conversation, companionship, sexual intimacy.

2.5 What do we lose by being benefited by machines rather than humans?

- The good/service/interaction might be *experientially* better if it involves a human:
  - it might be intrinsically different, e.g. the service is done better, or more erratically and therefore interestingly; the product is better.
  - it might be different because supplemented by further human interaction, e.g. talking with the person at the checkout.
  - We then lose the additional human quality of the goods, and the additional human interaction.
- The good/service/interaction might be experientially the same, but still relevantly different if it involves a human:
  - Actually talking to and being understood by another person.
  - Actually being cared for and loved.

2.6 NB: The experience machine rears its head

- cf. Nozick's Experience Machine: The Experience Machine is the ultimate computer-generated reality. Do we lose by going into it?
- Enter the general theory of welfare: what makes a life go better or worse?
  - experience-based (e.g. hedonism) vs. preference satisfac vs. objective list theory.
- Such a theory, developed with the aid of the experience machine example, can help us understand value in less futuristic cases.

2.7 Question: Activities where we don't lose or even gain from machine replacement

- Candidate activities where machine replacement might be an additional good *to the recipient*:
  - waste disposal.
  - home cleaning.
  - personal hygene.
- Explanation: There are activities in which we do not want to have other humans around, but sometimes need them.
- But this does not change the fact that there are many activities where we value human contribution and contributing ourselves.
- Hence computerized machine replacement is probably ethically ambiguous: sometimes problematic, sometimes only beneficial.

2.8 What we lose by not contributing as humans

- James Lenman:
  - "A significant and shared human contribution to the meeting of our needs is itself one of our deepest needs."

  - Core idea: What gives life meaning is not just to get the services, goods, and interactions we need from others or machines, but to provide to oneself and others the fulfilment of these needs.

- Lenman's claim can be understood as providing an element to an objective list of things that make life good.

- But even if you don't like objective list theories, many people will have a strong preference for contributing to the meeting of their and others' needs.

- Example: Being the only one not to bring something to a dinner party.
  - Problem: This example might be more about inequality. What about a robot-cooked (or supermarket ready-made) dinner party?

2.9 What we lose when others lose by not contributing as humans

- Lenman: Not only non-contributors, but also contributors lose out if some members of society only ever consume.

- Example: Being in a conversation where one person is systematically not contributing / being listened to.

- Because contributing is an important social need, not being able to contribute is a severe form of social exclusion.

- Everyone loses as a result of this exclusion: those excluded, and those who have to live in a society that denies some people full social membership.

2.10 Is computerized machine replacement a case of uniqueness?

- Computerized machine replacement is an extension of non-computerized machine replacement.
  - The contribution and exclusion concern is not specific to computerized machine replacement, but to any social change that allows society to forgo the potential contribution of some of its members.

- Computerized machine replacement increasingly allows for replacing more paradigmatically human tasks and makes the replacement concern more pressing.

- It also requires careful thinking of what intelligence, creativity, and caring amount to, and to which extent a give computer program has these features (cf. e.g. discussion threads about computer-generated music on youtube).

- It also requires us to rethink the value of so far distinctly human tasks and activities (cf. the experience machine; aesthetics and computer-generated music).

3 Black boxes and open sources

3.1 Free and Open Source Software: Some examples

Proprietary, closed source software	Free open source software
Microsoft Windows; Mac OS X	Linux, Open BSD, Android
Microsoft Word, Google Docs	Open Office, Abiword, Gnumeric, LaTeX
Internet Explorer, Safari, Opera	Mozilla Firefox, Midori, Chromium, (most of) Chrome
Adobe Photoshop	Gimp, Inkscape

3.2 The Free Software Foundation on free open source software:

"Free software" means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, "free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer". We sometimes call it "libre software" to show we do not mean it is gratis.

We campaign for these freedoms because everyone deserves them. With these freedoms, the users (both individually and collectively) control the program and what it does for them. When users don't control the program, we call it a "nonfree" or "proprietary" program. The nonfree program controls the users, and the developer controls the program; this makes the program an instrument of unjust power.

A program is free software if the program's users have the four essential freedoms:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

A program is free software if it gives users adequately all of these freedoms. Otherwise, it is nonfree. While we can distinguish various nonfree distribution schemes in terms of how far they fall short of being free, we consider them all equally unethical.

3.3 Is closed source software unethical?

- Worry: Is the Free Software Foundation overstating their case? Overreaching the concept of justice and right/wrong, and thereby watering it down?
  - "stealing is unethical" vs. "Microsoft Word is unethical"
  - "slavery is unjust" vs. "Mac OS X is unjust"
- Question: What do you make of the ethical claims of the Free Software Foundation?

3.4 Some concerns about black box machines

- Much computerized technology abstracts the commodity it provides away from the mechanism by which it provides the commodity:
  - i.e. you know what it does, but not how it does it, and you're often not meant to know, either.
- Some concerns about such black box machines:
  - Lack of user input in development and modes of operation.
  - Lack of user understanding of the machine.
  - Requires nearly blind trust of users in the developers.

3.5 NB: What is source code and why does it matter?

A simple programme in action:

```
Hello!
enter number one
10
enter number two
23
sum: 33
```

A brief look into the programme "executable" or "binary", the long string of zero's and one's (often represented as hexadecimal numbers 0-f for 8 zero's and one's, or one byte, per digit):

```
00000870: 5dc3 0f1f 4000 662e 0f1f 8400 0000 0000
00000880: be98 1260 0055 4881 ee98 1260 0048 c1fe
00000890: 0348 89e5 4889 f048 c1e8 3f48 01c6 48d1
000008a0: fe74 15b8 0000 0000 4885 c074 0b5d bf98
000008b0: 1260 00ff e00f 1f00 5dc3 660f 1f44 0000
000008c0: 803d 090c 2000 0075 1155 4889 e5e8 6ef0
```

And now the source code, the programme the programmer actually wrote, and which then got turned into the exectuable by a programme called "compiler":

```
#include <iostream>
using namespace std;
int a;
int b;
main(){
  cout << "Hello!" << endl;
  cout << "enter number one" << endl;
  cin >> a;
  cout << "enter number two" << endl;
  cin >> b;
  cout << "sum: " << a+b << endl;
}
```

3.6 Customization and autonomy

- Much software provides at least some space for customization by the user:
  - Changing background images, sounds, colour schemes.
  - Changing toolbars and shortcuts.
- But customization options are often limited:
  - the developer didn't bother,
  - if the user could decide at every junction what the program does, then this would no longer be user options, but the options would *be* the program.
  - so the developer must make a choice as to what aspects are user customizable and which are not. The developer then decides for you what options you need and do not need.
- The only way to circumvent this control of the developer is to have access to the source code. But often developers and companies want to protect their economic interests by keeping source code secret.
- Users can exercise some choice by switching between different pieces of software for the same kind of task.
- But this is a very small set of options, and is choosing between different bundles of limitations.
- Also: Software companies have a financial interest in trying to capture the market by creating user lock-in: Key example: MS office.
- Question: Restricted user customization and choice may be annoying and/or economically inefficient. But is it a case for ethics?
- Easy answer: It is, when it comes with economic inequality and possibly exaggerated profits of corporations. But this aspect is not unique to computers.

3.7 A non-computer example

- The carpenter
  - You are a carpenter who makes beautiful tables and chairs.
  - You have your own, self-made and customized workshop, and traditional tools which you repair and modify for your specific technique.
  - Some day, customers start demanding furniture that is made with a particular kind of screw.
  - You go and buy the screws.
  - But the screws can only be turned with a special screwdriver.
  - The screwdriver needs a special kind of power supply on the workbench.
  - You have no idea of how the supply works, and can't (and legally may not!) just make your own.
  - Only a few types of workbench provides this power supply.
  - The workbenches only come in a few different dimensions and cannot be modified.

3.8 a

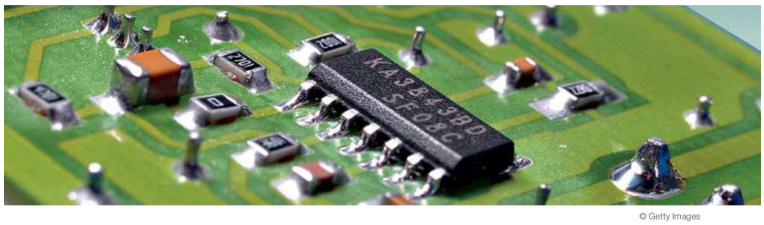
- Question: Did you lose anything of moral significance when you moved to the proprietary screw system?
- You lose money and become captive to the screw-making company and their tool system.
- But you also lose space in which to autonomously exercise your creativity and skill, and follow your individual preferences and ideas of how to do things.
- Instead of shaping your work and life environment, someone else does, and you adapt yourself.
- You are then heteronomous (hetero: other, nomos: law), not autonomous (autos: self, nomos: law).

3.9 Computers, autonomy, and ethics

- End-user computer programs (including for mobile devices) become part of the "space" in which we live and work, and shape our activities in that space.
- Brushing off concerns about lock-in and lack of customization, underestimates how important this space is in our lives.
- Through black box devices and programs, we allow an anonymous group of engineers to metaphorically clutter up our workspaces and homes, to rearrange and change furniture, to change the way we move around, live our lives, work, and to make a lot of decisions for us.
- Objection: This heteronomy is self-chosen and can be escaped.
- Responses:
  - It is often not consciously self-chosen, and very costly to escape e.g. due to lock-in and network effects.
  - Even when it is self-chosen, it is still a loss of which we should be aware.

3.10 But I am not a nerd (unlike you)!

- The Linux Kernel (very core of the system, before you see any programme windows or write your essay) alone has some 19 million lines of code.
- Mozilla Firefox clocks at 10 million LOC.
- Libreoffice/open office: 20 million LOC.
- How is this huge amount of code any more transparent to you and modifiable to you than binary executable files?
- It is not.
- But a large community of programmers with no unified vested economic interest can look at the code and contributes to the code.
  - This is not the case for closed-source software, where only paid developers in a company know the code, and are contractually required to keep it secret.



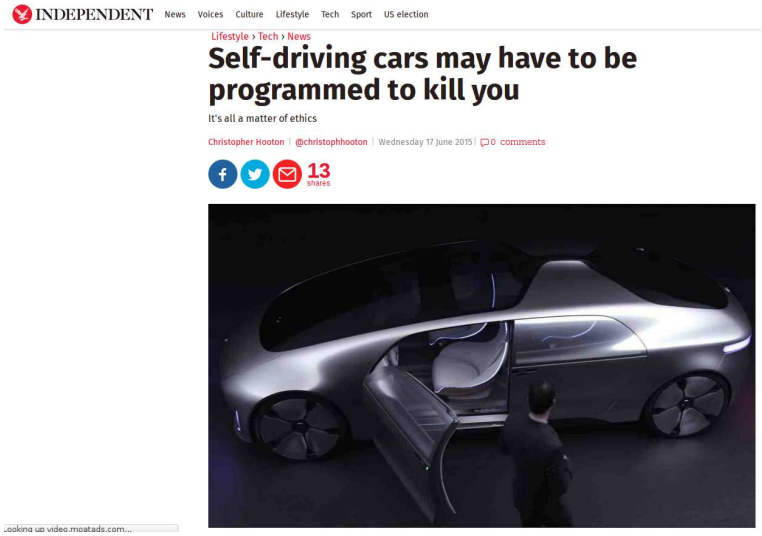
- You can contribute in forums and bug-tracking platforms to suggest or vote on changes, and can sometimes also financial support new features.

3.11 Beyond common software: Microchips and the internet of things

- Programmed microchips are extremely cheap and ubiquitous.
  - An average car has some USD 120 worth of programmable microcontrollers in them, so easily into the hundreds of individual chips.
  - Washing machines, microwaves, doorbells, alarm clocks, radios, ...
  - Example: A homemade medium-sophisticaed alarm clock to control remote-control sockets has about 3000 lines of code running on the main processor, plus code that runs on connected devices and chips.
- More powerful and wirelessly connectible microcontrollers are getting ever cheaper and enable an "internet of things", where e.g. household devices are connected to each other and the internet.
- This technology can lead to enormous efficiency gains e.g. in power distribution.
- But it also raises more privacy and security concerns, and open source software would help alleviate these concerns.
- Likewise, open source software *and hardware* would allow for modifying machines and give us more autonomy in our homes and workspaces.

3.12 Below (but not beyond!) right and wrong

- The dominance of closed-source proprietary software is not technologically necessary, and has not always been the case.
- Closing sources and selling software has been the industry's main way of incentivising innovation.
- This way of incentivising innovation has led to a lot of good new software.
- But it comes at a cost to user autonomy (as well as social accessibility and security).
- This cost does not in itself make closed-source software "unethical", but it is a cost, a *pro tanto* ethical and prudential reason, we need to be aware of.
- Once a market player locks people into a closed-source ecosystem and network effects create pressures on individuals to adapt such a system and bear thia cost, then issues of freedom become more pressing: Such behaviour and market pressures might indeed be a form of oppression.



3.13 Open source software and uniqueness

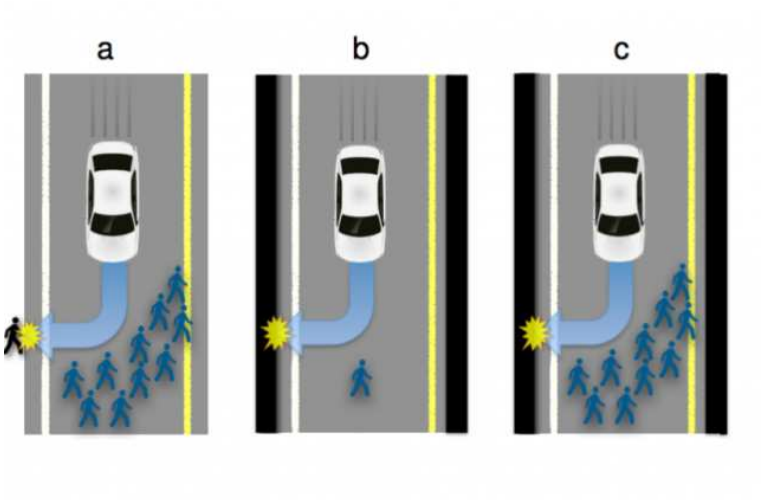
- Are the concerns about black box programs a case for uniqueness of computer ethics?
- Economic concerns about monopolies, lock-in, and profits are not unique to software.
- User autonomy concerns raise their head not only with regard to software, but to other bits of technology that are deliberately designed to not be modified.
- But due to the complexity of computerized machinery, and the easy of obfuscating how it works, user autonomy concerns are particularly salient here.

4 Self-driving cars

4.1 Some pictures

4.2 Ethical question: Self-driving cars and trolley problems

- Question: How should the car be programmed to behave?
  - Go for saving the greatest number?


a
b
c

  - Go for saving the greatest number, but not kill the passenger?
  - Go for saving younger over older people?
  - Prefer more robust collision "targets" over more soft "targets".
  - Leave the choice to the driver?
  - Make a random choice?
  - Make a probabilistic choice that makes live equally dangerous for all road users?
  - Prefer other cars over non-car road users?

4.3 Do self-driving cars need their own ethical code?

- Programming the collision evasion software may seem to make the car think ethically in a particular way.
- But ultimately, it is a mere design decision with tradeoffs in human lives. Other examples:
  - designing airbags and optimizing for lives saved: makes people with some body shape more likely to survive than others.
  - allowing for heavy SUV's which reduce risks in collisions to drivers and passangers, but increase the risk to those in smaller cars around.
  - allowing for cars to have maximum achievable speeds above the national speed limit.
- Explicitly programming a collision evasion software only makes explicit ethical choices and tradeoffs that are already implicit in other engineering decisions.
- So self-driving cars need no new ethics, and only show that engineers practiced ethics and needed ethical reflection all along.
- Given the shared nature of the risks involved, such ethics is a ripe case for public discussion and possibly legislation.

4.4 Ethical question: Self-driving cars and responsibility

- Who is morally responsible if a self-driving car crashes?
- The driver?
  - They chose to use the car in the first place, knowing that cars risk damages and death.
  - But they were merely morally unlucky to have the risk materialized.
- The producer?
  - But crashes are sometimes unavoidable.

4.5 Non-computerized example of responsibility

- A GP prescribes a drug to a patient.
- The drug has a tiny chance of harming the patient, and this chance is unavoidable and clearly advertised.
- The patient gets harmed.
- Who is responsible?
  - The doctor was merely unlucky, and did nothing reckless.
  - The pharmaceutical producer could not produce a side-effect free drug, and had to decide which side effects to avoid and which to accept.