

Repository Entry Template

Embedded EthiCS @ Harvard Teaching Lab

Overview

Course: CS 51: Abstraction and Design in Computation

Course Level: Introductory Undergraduate

Course Description: “CS51 teaches fundamental concepts in the *design* of computer programs, emphasizing the crucial role of *abstraction*. The goal of the course is to give students insight into the difference between programming and programming well. One and the same problem can be solved in different ways, and the different solutions can vary along multiple dimensions including correctness, efficiency, readability, scalability, and elegance.

To emphasize the differing approaches to expressing programming solutions, you will learn to program in a variety of paradigms – including imperative (familiar from CS50 but seen here in a more elemental form), functional, and object-oriented. The elegant multi-paradigm programming language OCaml is the ideal language for manifesting these ideas. Important ideas from software engineering and models of computation will inform these different views of programming. You should come out of the course a better programmer in any language, but also a better computational thinker, with a much broader range of tools at your disposal and ability to analyze the quality of programs.”¹

Module Topic: Moral Responsibility and Social Networks

Module Author: Megan Entwistle

Semesters Taught: Spring 2023

Tags: networks [CS], fake news [CS], moral responsibility [phil], intervening agents [phil], free speech [phil]

Module Overview: The module is an introduction to the topic of moral responsibility in the context of social network design. The first part of the module introduces conceptual tools for thinking about backwards-looking responsibility, i.e. ascribing praise and blame for actions that lead to morally significant outcomes. Students learn to assess what user behavior, if any, Facebook is morally responsible for. The second part of the module focuses on forwards-looking responsibility, or how to design systems with an eye towards potential downstream impacts.

Connection to Course Material: The course discusses issues in computer programming, and how programming can be done well. The module complements the course by looking at the broader impacts of programming. The concept of moral responsibility is used to illustrate how far-reaching these impacts can be. The backwards-looking aspect encourages students to consider how much of the responsibility for bad consequences rests with the initial designers of a program. The forwards-looking aspect encourages students to undertake their programming tasks responsibility, i.e.

The topic of moral responsibility is important for future software engineers to engage with, insofar as it encourages critical examination of the (often default) idea that the designers of a program are morally isolated from any downstream harms perpetuated by users of that program. Focusing on social networks (where platform design

¹ <https://cs51.io/college/syllabus/>

in anticipation of reasonably foreseeable consequences.

choices meet individual user decisions) is a particularly fruitful way to interrogate the ethical relationship between the various participants in a technology.

The topic for this module builds upon a previous iteration by Samuel Dishaw (see <https://embeddedethics.seas.harvard.edu/classes/cs-51-2021-spring>). The present version of the module drops the discussion of Facebook's Oversight Board, focusing instead on the notion of reasonable foreseeability and the forwards-looking responsibility that software engineers have.

Goals

- Module Goals:**
1. Encourage students to appreciate how far-reaching the impacts of programming decisions can be, and in light of that appreciation cultivate forward-looking responsibility for design choices.
 2. Familiarize students with key concepts and distinctions in the philosophy of moral responsibility (causation, intervening agents, omissions, foreseeable consequences).
 3. Give students practice articulating verdicts about moral responsibility in different cases.

- Key Philosophical Questions:**
1. Are we only responsible for those outcomes that are the direct result of our own actions?
 2. Can we be responsible for bad outcomes that are the direct result of someone else's action, but which we played a role in enabling?
 3. Can we be responsible for bad outcomes that we merely allowed to happen?
 4. What constitutes 'reasonable foreseeability' in the context of programming and design?

The discussion of questions (2) and (3) are each paired with a case study.

The first case concerns discriminatory housing ads on Facebook. Facebook's 'Special Ad Audience Tool' gave users the option of excluding particular groups of individuals from seeing the ad, on the basis of race, gender or religion. Following discussion of question (2) in connection with the case study, a new principle of moral responsibility is presented, according to which one is responsible for a (bad) outcome if they either (i) directly caused it, or

(ii) acted in a way that would foreseeably lead to others causing it.

The second case study pairs with question (3) and is concerned with vaccine misinformation on Facebook, focusing on the case of Robert Kennedy Jr. in 2021. In this case, students are asked to consider whether Facebook bore any responsibility for the fact that, as a result of Kennedy Jr.'s posts, fewer people will get a vaccine than otherwise would have. This case study is designed to connect the concept of 'omission' to moral responsibility. Following discussion, the working principle of moral responsibility is amended further to include responsibility for *allowing* others to do something that would foreseeable lead to a bad outcome.

Materials

Key Philosophical Concepts:

- Responsibility (Backwards-Looking vs. Forwards-Looking)
- Causation
- Intervening Agents
- Omission
- Reasonable Foreseeability

The notion of causation is used in the formulation of a first-pass principle of moral responsibility, which says that one is responsible for some bad outcome just in case one caused it.

An intervening agent is someone who acts 'in between' another agent and a bad outcome. The users on Facebook are intervening agents relative to Facebook. When Facebook users act wrongly (e.g. post discriminatory housing ads), they perform an action that they couldn't have performed if it weren't for something that the programmers of Facebook previously did (e.g. give users the option to manually select a target audience along racial lines). For that reason, the question of whether Facebook is responsible for bad outcomes that results from the use of their platform needs to

<p>Assigned Readings:</p> <ul style="list-style-type: none"> ● Zimmerman, MJ (1985). "Intervening Agents and Moral Responsibility", <i>Philosophical Quarterly</i>. ● <i>IEP</i> entry (sections one and two) on "Responsibility": https://iep.utm.edu/responsi/#H3 	<p>address whether one can be responsible for bad outcomes that are mediated by intervening agents.</p> <p>The Zimmerman article is assigned to illuminate the notion of intervening agency. It makes a case for why individuals can be held morally responsible for the outcomes of their actions even where the choices of third parties enter into the causal chain (one of the central points of discussion for the module). Another feature of the article is its wealth of examples and clarity of argumentation. It can demonstrate to computer science students how philosophers argue about moral responsibility more generally. Because it is fairly dense, it is paired with excerpts from an <i>IEP</i> survey article on moral responsibility, which serves as a primer for the topic.</p>
---	---

Implementation	
<p>Class Agenda:</p> <ol style="list-style-type: none"> 1. Responsibility as Causation 2. Intervening Agents: Housing Discrimination 3. Omissions: Vaccine Misinformation 4. Reasonable Foreseeability 	<p>The module begins with a distinction between two kinds of responsibility (forwards-looking and backwards-looking), both of which are important for thinking about the ethical duties of programmers. Agenda items 1 - 3 walk the students through an increasingly sophisticated formulation of a principle of backwards-looking responsibility. Agenda item 4 returns to forwards-looking responsibility, having identified a key notion (viz. reasonable foreseeability) from the discussions of parts 1 - 3.</p>
<p>Sample Class Activity: To problematize the first-pass causation principle of moral responsibility, students are presented with the following scenario and question:</p> <p>Alice runs a small general goods store. On Monday, she sells some rat poison to Zebulon. On Tuesday, Zebulon feeds that rat poison to his</p>	<p>Student responses tended towards (a) over (b) during the poll itself; after group discussion, more students formed the opinion that Alice was potentially complicit. The point of the exercise is for students to appreciate that, whichever way</p>

neighbor's dog, causing its death.

Who is responsible for the dog's death?

- (a) Zebulon
- (b) Zebulon and Alice

The instructor takes a class poll, and then asks students to explain the intuitions behind their answers.

Module Assignment: Write a 300-400 word essay responding to the following question:

In your role as a software engineer, if you can reasonably foresee that a certain design choice or algorithm would lead to harmful outcomes, what should you do?

To develop your answer, provide an example scenario: this can be from your own experience, something you've read about in the news, a case study covered in the lecture, or a hypothetical scenario of your own design. Be sure to specify exactly which features about your example support your ethical conclusion about what to do. Your reasoning should make use of at least one notion discussed in the module.

they voted, the causation principle is unable to extend moral responsibility to Alice, which poses a more general problem for causal chains involving multiple actors. This sets the stage for the instructor to introduce the concept of intervening agents and the central case study of social networks.

The reflection assignment was left intentionally open-ended to encourage students to connect the concepts and lessons from the module to their own experiences as programmers.

The end of the module presents the case study of Amazon's 2015 hiring tool, which used a ML algorithm to rank the resumes of prospective software engineers. Due to its training sets, the algorithm systematically favored white male applicants. Students were asked to consider the extent to which Amazon employees should have anticipated the gender bias in the ML-powered hiring process. This example was designed to set students up for reflecting on forwards-looking responsibility in the assignment.

The essays are peer-evaluated. Each student receives three essays from other students. They are asked to paraphrase the main thesis of each essay and grade them along a provided rubric. Students thus learn not only to express their views using argument, but also to evaluate the arguments of others, and respond to them in a helpful way (feedback on essays is also peer-graded).

Lessons Learned: Student response to the module was overwhelmingly positive. Students engaged enthusiastically with the topic, were ready to offer their own examples, and felt they walked away with

a toolbox of important philosophical concepts applicable to their own experiences.

1. Future TFs might consider scaling back some of the complexity and/or examples in the discussion of backwards-looking responsibility, so as to leave more time to develop and practice applying the notion of reasonable foreseeability in the latter part of the module.
2. Framing the case study discussion prompts in terms of 'who is responsible: Facebook, its users, or both?' is not always the best way to elicit student responses initially. A better strategy might be to ask students to share more open-ended thoughts about the harm done, how that harm came about, and how it might have been prevented, *before* explicitly connecting the case study to the working definition of moral responsibility at hand.