Repository Entry Template
Embedded EthiCS @ Harvard Teaching Lab

Overview

Course:	CS 152: Programming Languages	
Course Level:	Upper Level Undergraduate	
Course Description:	Comprehensive introduction to the principal features and overall design of both traditional and modern programming languages, including syntax, formal semantics, abstraction mechanisms, modularity, type systems, naming, polymorphism, closures, continuations, and concurrency. Provides the intellectual tools needed to design, evaluate, choose, and use programming languages.[1]	
Module Topic:	Programming Ethical Performance	
Module Author:	Zachary Gabor	
Semesters Taught:	Spring 2021	
Tags:	Software verification and validation [CS], Machine Ethics [both], moral rights [phil]	
Module Overview:	The goal of this module is to convey to students a tractable methodology for thinking about designing systems to perform in ethically acceptable ways and testing them.	
Connection to Course Material:	From David Gray Grant's Entry: "In the lead-up to the module, the course covers automated techniques that can be used to verify that a software system will behave in accordance with its design specifications. In this module, we introduce the idea of ethical design specifications, and consider how these might be verified (either using techniques covered in the course, or other methods)."	This module is adapted from previous in-person modules for remote instruction.

Goals

Module Goals:	1. Familiarize students with a tractable general framework for identifying ethical requirements on software performance. 2. Provide students with practice devising such constraints to applications.	
Key Philosophical Questions:	1. What are ethical requirements on machine performance and how do we identify them? 2. How can ethical operation be ensured during the process of verification and validation?	Students may be familiar with the concepts of verification (confirming that a design meets design specifications) and validation (confirming that a design which meets these specifications can perform its intended function) as generally applied in software engineering. In this module, students are asked to use this framework to think about the ethical features - both

[1] my.harvard

		intended effects and the specifications designed to implement them - of their designs.

Materials		
Key Philosophical Concepts:	● Rights ● Interests ● Stakeholders	The core tool that the module presents is the method of stakeholder analysis, whereby designers specify stakeholders in a project and their rights and interests as guideposts to formulating rules which articulate ethical constraints on the operation of the machine they are designing.
Assigned Readings:	● Chloe Rose Stuart-Ulin (2018), "Microsoft's politically correct chatbot is even worse than its racist one" ● Jason Torchinsky (2018) Consumer Group Says Tesla's Autopilot Is 'Deceptive,' Calls For Investigation (jalopnik.com)	These articles summarize two of the cases on which the modules focus: Microsoft's outrageously offensive and more subtly offensive chatbots Tay and Zo, and Tesla's autopilot function which has been engaged during several fatal accidents and has drawn criticism of Tesla related both to its marketing and its safety design.

Implementation		
Class Agenda:	1. Introduction to three case studies: the Therac-25 (radiation therapy machine that killed and injured patients due to software bugs), Microsoft's Tay and Zo, and Tesla's autopilot 2. Presentation: Stakeholder Analysis 3. Stages in ethical software design 4. Activity: apply ethics-focused design	To explain stakeholder analysis, the Embedded EthiCS TA defines and explains stakeholder analysis. Stakeholder analysis is the identification of moral rights and interests of a project's various stakeholders. Its four steps are: 1) identification of ethical requirements by stakeholder analysis; 2) formulation of rules governing the operation of the system to meet these requirements; 3) Verification of software in relation to these requirements; 4) Validation of system performance in relation to these requirements.
Sample Class Activity:	Students are asked to consider how the method of stakeholder analysis could have had an effect in mitigating the harms in the three cases discussed.	This activity provides students a chance to think about the practicalities of applying the

	Divided into groups, students are asked to perform a stakeholder analysis on each of the three cases. They then are prompted to formulate rules constraining the performance of the systems being designed in each case. Finally, they consider ways in which ethical performance in the case they were studying could have been ensured by verification or validation.	approach to ethical design presents.
Module Assignment:	Students are asked to apply stakeholder analysis to the design of an automated weapons system. They are prompted to formulate rules which respect the identified ethical requirements in cases where communication between the weapons system operator and the system dropped or experienced high latency.	This assignment offers students the chance to practice applying the analysis and design approach taught in the module in a new context. Students are provided with the doctrine of just war as a strict ethical constraint to govern the behavior of their systems. This makes the task of identifying design specifications less open-ended.
Lessons Learned:	Student responses to the follow-up assignment were quite detailed and well thought out. We think this is a sign that it is probably doable and worthwhile to challenge students by introducing subtleties about stakeholder analysis in class. For instance: in major enterprises like the projects of multinational corporations or state actors, it is conceivable that *everyone in the world* is a stakeholder in the enterprise, i.e. someone whose rights or interests might well be affected by how the project goes. To accommodate this point it could be useful to discuss ways of thinking about stakeholdership as a matter of degree and ways in which this affects ethical constraints on design.	