

CS 61 Repository Entry Embedded EthiCS @ Harvard Teaching Lab

Overview

Course: CS 61: Systems Programming and Machine Organization

Course Level: Undergraduate

Course Description: “Fundamentals of computer systems programming, machine organization, and performance tuning. This course provides a solid background in systems programming and a deep understanding of low-level machine organization and design. Topics include C and assembly language programming, program optimization, memory hierarchy and caching, virtual memory and dynamic memory management, concurrency, threads, and synchronization.”

Module Topic: Ethics of Language Encoding

Module Author: Eliza Wells

Semesters Taught: Fall 2021-22

Tags: natural language encoding [CS], Unicode [CS], emojis [CS], harms [phil], representational harm [phil], allocative harm [phil], reasonable accommodation [phil]

Module Overview: This module explores the ethics of natural language representation in computer systems. Historically, language encoding systems have not represented all of the world’s languages, or have not represented them equally well. The module helps students articulate these failures in terms of allocative and representational harms. One problem with systems before the introduction of the current encoding standard (UTF-8) is that avoiding those harms would have involved incurring technical costs, since systems would then require more energy and storage. Introducing the principle of reasonable accommodation provides one way of thinking through how to balance technical costs while avoiding harms. Students then apply these tools to case studies.

Connection to Course Material: Students learn about data representation and storage in the course. One application of these concepts is language encoding systems. This module is co-taught with the professor for the course, who lectures on the technical material alongside the Embedded EthiCS fellow.

The advantages of language encoding as a case study include:
a) the link between computer scientists’ decisions and impacts on other people is easy to see, b) it involves clear examples of both allocative and representational harms, c) there are technical as well as social solutions to the problems it raises, and d) it allows for fun examples, such as emojis or encoding Klingon.

Goals

Module Goals: 1. Consider how language encoding systems can cause harms

<p>Key Philosophical Questions:</p>	<p>2. Understand and be able to identify instances of two specific kinds of harms: representational and allocative</p> <p>3. Think through how to balance avoiding those harms while avoiding technical costs using the principle of reasonable accommodation</p> <p>4. Apply these concepts using case studies</p> <p>1. Do computer scientists have moral obligations to incur technical costs in order to accommodate communities?</p> <p>2. How can choices about language encoding harm communities?</p> <p>3. How should harms be weighed against technical costs?</p>	<p>The goal of the module is to equip students to answer the first question by considering the second and third questions. While the case study of language encoding can make these questions particularly visible, the module also aims to provide students with tools to address similar trade-offs they might need to make in other cases. Another version of this module might ask students for examples of reasonable accommodation in other design decisions to bring this latter goal out more clearly.</p>
--	--	--

<p>Materials</p>		
<p>Key Philosophical Concepts:</p>	<ul style="list-style-type: none"> ● Representational harm ● Allocative harm ● Reasonable accommodation ● Undue burden 	<p>The vocabulary of representational and allocative harms helps students see different ways in which choices about language encoding can harm communities. However, identifying different harms is merely a first step in understanding computer scientists' moral obligations, because language encoding has historically involved technical costs. The principle of reasonable accommodation provides a way to determine what one's obligations are – in the language encoding case and in other situations students might encounter – by presenting standards for making tradeoffs between avoiding harms to communities and avoiding technical costs.</p>
<p>Assigned Readings:</p>	<ul style="list-style-type: none"> ● Deborah Anderson, "Global Linguistic Diversity for the Internet," <i>Communications of the ACM</i> 48, no. 1, 2005 	<p>Anderson's paper is written in a post-UTF-8 world, where the technical costs of language encoding are low. She discusses examples of harms that communities whose languages are not encoded experience. In class,</p>

these harms can be understood in allocative and representational terms.

Implementation

Class Agenda:	<ol style="list-style-type: none"> 1. Introduce the concepts of representational and allocative harm using examples of emojis and language encoding systems prior to Unicode 2. Note that avoiding some harms can cause other harms: encoding systems that can represent more languages are more technically expensive 3. Introduce the concepts of reasonable accommodation and undue burden 4. CS professor lectures on UTF-8, a technical solution that decreases the burden 5. Active learning exercises with case studies involving real languages 	<p>This module had students read a history of language encoding up to UTF-16, the most recent encoding standard before UTF-8, before class. In encoding standards before UTF-8, there were significant technical costs to representing more languages. Students learned about the principle of reasonable accommodation as one way to ethically navigate tradeoffs between avoiding harms and avoiding technical costs. Once these trade-offs and principles to navigate them were made clear, the professor for the course lectured on the modern, less costly UTF-8. From this, students learned that there can be technical as well as social solutions to ethical problems. The reasoning that they practiced using the principle of reasonable accommodation, however, can also help them to navigate other conflicts in which there are not yet technical solutions in the future. The goal of this activity is to give students practice using the vocabulary they've learned in the module. It prompts students to consider the contextual details of each case to see that different harms can arise in different ways, and to see that answers about what ought to be done are sensitive to that contextual information. This module used Toto, Klingon, and Maya as examples.</p>
Sample Class Activity:	<p>Students are presented with case studies of different scripts that have not yet been encoded in Unicode. Students are asked to discuss the following questions in small groups: what are the most significant harms at stake in choosing whether or not to encode these scripts in Unicode? What should computer scientists do?</p>	<p>The goal of this activity is to give students practice using the vocabulary they've learned in the module. It prompts students to consider the contextual details of each case to see that different harms can arise in different ways, and to see that answers about what ought to be done are sensitive to that contextual information. This module used Toto, Klingon, and Maya as examples.</p>
Module Assignment:	<p>Because this was a large class, students were given a brief multiple choice quiz testing comprehension of key terms.</p> <p>Example questions:</p> <ul style="list-style-type: none"> ● One representational harm of limited language encoding is... ● The principle of reasonable accommodation asks... 	<p>A more engaging assignment might involve asking students to write a short essay about a new language case study that raises the technical costs of encoding (perhaps by involving billions of characters that would require increasing the</p>

Lessons Learned:

1. Students were engaged and seemed to easily grasp the importance of thinking about different kinds of harms. They were able to apply that distinction in thinking about the case studies.
2. Students wanted to have more conversation about reasonable accommodation than this module had time for – some students wanted to reject it on utilitarian grounds, while others wanted to discuss whether we are all morally obligated to create reasonable accommodations.
3. This module used examples of reasonable accommodation in non-technical contexts (e.g. accommodations for disability or religious needs), but additional examples of creative technical accommodations would also have been helpful in giving students tools to apply this principle in their work.

amount of storage required for the encoding standard).