Technically Ethical:

A Discourse Analysis of the ACM/IEEE-CS's 2020 Computing Curricula Recommendations

Anna Ma, Elizabeth Patitsas, and Hana Darling-Wolf

Abstract

How to engage scientists and technologists with the sociopolitical and environmental consequences of their work remains an open concern for the Science and Technology Studies (STS) community. In recent years, computer science has made a greater push for teaching ethical and social issues in light of societal "techlash". Past work in STS has described how computing education as teaching undergraduates to "render technical" (Breslin 2018): to take complex, real-world problems, strip away the sociohistorical context, and solve only the technical problem. And given that past work in the history of computing has identified curriculum reports of the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) as significant for shaping the field, we performed a discourse analysis of the most recent computing curricula standards from the ACM and IEEE-CS. We found that although the curriculum report states ethics education is important and should be spread throughout the curriculum, the curriculum standards themselves discursively compartmentalize ethics and then render it into the technical problems of reducing "error" from professionals and enacting a (hegemonic) benevolent designer-user relationship. We illustrate how a scientific field can give an impression of commitment to social and professional education without engaging with the underlying issues, and compare conceptual frameworks for understanding hegemonic ideology in computing.

1 Introduction

Within just recent memory, the modern tech industry has witnessed a well-documented explosion of ethical crises (Dao et al. 2022), resulting in heightened state and corporate surveillance (Ahmed 2018; Chun 2018; Hill 2014, 2020; Jiang 2017; Liptak 2018; O'Neill 2019; Ziv 2019), discriminatory criminal justice sentencing systems (Angwin et al. 2016), increased autonomous weapons development (BBC News 2020; The New York Times 2019; Vincent 2021), and the rampant propagation of online disinformation (Greenfield 2018; Metz and Blumenthal 2019; B. Quinn and Ball 2014). As such, STS scholars have taken a greater interest in how CS is taught, and how to teach STS to CS audiences (e.g. Joyce et al. 2018; Malazita and Resetar 2019; Parvin and Pollock 2020; Shilton 2013; Ramachandran et al. 2022).

We have chosen to focus on investigating and revising the quality of socioethical education within Computer Science (CS) education as a practical, high-leverage approach to improving the ethics of the tech industry writ large. Currently, CS education pays insufficient attention to teaching computing in a way that integrates ethical (Fiesler 2018) and material concerns (Mayhew and Patitsas 2021) with the technical content. This feeds into the "I'm just an engineer" mentality that abdicates professional responsibility for considering the ethical and social impacts of the technologies they build (Fiesler, Garrett, and Beard 2020, 6) and the perpetuation of a professional culture characterized by toxic masculinity (Lehr 2006; Nafus 2012), casual racism (Amrute 2020), and ableism (Wong 2015; Ymous et al. 2020).

According to Fiesler (2018), the ongoing ethics crisis facing the tech industry can be attributed to ineffective ethics education in CS. She argues that the current ethics crisis in tech is a reflection of how ethics is taught independently from technical classes in CS education, which results in engineers perceiving ethics to be an unimportant afterthought when developing new technologies. Fiesler argues that ethics education needs to be embedded in technical course content, raising crucial questions about how the design process for CS curricula needs to be completely re-imagined to include ethics as a fundamental pillar rather than an add-on module. Similarly, Mayhew and Patitsas (2021) recount their experiences as digital logic educators and conclude that CS education is mainly taught in an abstracted way that dismisses computing's material and environmental impacts.

But computer science does not have to be this way. Design methodologists and computer scientists have developed computing methodologies which integrate social, political, ethical and/or environmental considerations. Examples include feminist HCI (Bardzell 2010; Light 2011; Rosner 2018; Suchman 1993), crip technoscience (Hamraie and Fritsch 2019), decolonial community partnerships (Nathan et al. 2017), algorithmic realism (Green and Viljoen 2020), Value-Sensitive Design (Friedman, Kahn, and Borning 2002), and Soft Systems Methodology (Checkland and Poulter 2020). Many of these methods come either directly from STS (e.g. crip technoscience) or are translations from STS (e.g. feminist HCI builds on feminist technoscience). The computer-supported cooperative work (CSCW) community in particular has served as an important bridge for translating STS concepts into CS practice ("Computer-Supported Cooperative Work" 2023)[1].

However, the dominant norms of CS education continue a separation of social and technical. The result is a curriculum that largely ignores the vast material and environmental costs of computing, and results in a *decrease* in CS students' commitment to social justice over the course of their degree, as illustrated by a survey study of more than 5,500 students from over 120 American colleges and universities (Núñez et al. 2021).

[1] CSCW is a design-oriented interdisciplinary field that studies the use of social, collaborative technologies.

If CS majors are taught about ethical and social issues at all, it is typically in an isolated course (Fiesler, Garrett, and Beard 2020; Williams et al. 2022); and in a way that is abstracted, disconnected, and formalized (Mayhew and Patitsas 2021; Suchman 1993; Williams et al. 2022). Ethics is presented as a series of trolley problems without ever asking why the metaphorical people are on the track (Williams et al. 2022); students are taught a handful of ethical theories such as utilitarianism and Kantianism and then effectively asked to apply them like algorithms to the trolley problems (Mayhew and Patitsas 2021).

In this paper we analyze how socioethical education is framed and represented in current CS curriculum standards. Do computing curricula demonstrate any commitment to social, environmental and/or ethical education? And if so, how is it framed?

We contribute to the STS community insight on how a scientific field can give the impression of committing to socioethical and professional issues without engaging with the underlying causes of these socioethical and professional issues. We further contribute a discussion of two useful conceptual frameworks for analyzing ideology in computing: rendering technical (Breslin 2018) and algorithmic formalism (Green and Viljoen 2020). We also add evidence to documented sociological phenomena in computing: the designer/user power dichotomy (Suchman 1993) and the gendered occupational closure (Witz 1992) of the field through the marginalization and ejection of subfields that have become more feminized (Patitsas 2019; Tijdens 1997).

2 Background

2.1 Historical Context

Developing socially responsible CS education at scale will require uptake by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS)—the major professional bodies that have developed the curricula recommendations used to internationally standardize university-level CS education for over the last 50 years (Sahami et al. 2013). Indeed, Skirpan et al. (2018) reference the effectiveness of the top-down guidelines provided by bodies like the ACM in determining the state of ethics education in CS, demonstrating that professional bodies like the ACM and IEEE-CS can play a crucial role in advancing socially responsible CS education.

Dziallas and Fincher (2016) emphasize the formative role that the ACM has played in defining CS as a discipline, recounting that “the ACM has been a major contributor to curricular standards in computing beginning in 1968 when the ACM was the first professional organization to release a model curriculum for computer science” (91-92). Meanwhile, the IEEE-CS is an offshoot of its parent organization, the Institute of Electrical and Electronics Engineers (IEEE), otherwise known as “the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity” (IEEE Computer Society 2023). Since its inception in 1946 as a subcommittee on large-scale computing devices, the IEEE-CS has focused on “engaging computer engineers, scientists, academia, and industry professionals from all areas of computing” in order to set educational standards that align with the evolving demands and advancements of the tech industry (IEEE Computer Society 2023).

From 1991 onward, the ACM published curricula reports jointly with the IEEE-CS, which provided guidelines encompassing “course descriptions [and] learning outcomes”

(Dziallas and Fincher 2015, 81), "specific course sequences" (Dziallas and Fincher 2016, 92), and examples of course implementations. These curricula guidelines have not only "influenced educators at institutions across the U.S." by specifying the depth and breadth of a CS degree's content knowledge, but have also been referenced internationally by the United Kingdom Quality Assurance Agency and the China Computer Federation for the development of their own local CS curricula (Dziallas and Fincher 2016, 92), and were used by the United Arab Emirates University when re-designing their computing curriculum (Samaka 2002, 35). As an example of how the guidelines affect university curricula, Burtscher et al. (2015) describe in detail their method of adopting and implementing the 2013 curricula guidelines at Texas State University.

Consequently, we interrogate the latest updated ACM/IEEE-CS curriculum report titled Computing Curricula 2020: Paradigms for Global Computing Education (CC2020) by using Gee's (2011) discourse analysis. Released on December 31, 2020, this report encompasses seven computing disciplines: Computer Engineering, Computer Science, Cybersecurity, Information Systems, Information Technology, Software Engineering, and Data Science. For the purposes of this paper, we focus specifically on the sections related to CS competencies and social issues, ethics, and professionalism.

2.2 Theory: Hegemony in Computing

We draw on Samantha Breslin's ethnography of undergraduate CS education in Singapore for our theoretical framework.[2] Building on Tania Murray Li (2007), Breslin investigates the hegemonic process of "rendering technical," a set of practices through which CS

[2] Specifically, we use her Gramscian discussion of hegemony as well as her Foucauldian-derived concept of "rendering technical" to explain the preponderance of technical content in current CS curricula that forfeits the integration of socially responsible content focused on ethics and the material impacts of computing.

students learn to filter, translate, and reconstruct reality through code, data structures, algorithms, models, and programs that render nuanced, multi-dimensional policy issues into technical components and computational solutions that are stripped of their critical sociocultural, political, and historical contexts.

Breslin also depicts how the practice of rendering technical leads to a false dichotomy between the social and the technical. This is reflected in the popular design of CS curricula that frequently siloes ethics education and technical content knowledge as separate domains (Fiesler 2018). By rendering technical, computer scientists contribute to reifying the mistakenly drawn boundaries between the social and the technical rather than viewing the two as interconnected.

This separation has long been critiqued by scholars in CSCW and Human-Computer Interaction (HCI) — subfields of CS which are highly multidisciplinary, and have been central sources of social critique within the field (Dourish 2019; Ehn 1988; Suchman 1993; Williams et al. 2022). As these scholars note, the boundary between social and technical knowledge in computing grew out of computer science's historical attempts to gain legitimacy and prestige (Dourish 2019; Ehn 1988). In order to access the resources afforded to "hard" sciences, CS closed itself off from feminized subjects such as skills associated with the humanities and social sciences (Patitsas 2019; Tijdens 1997).

This falls in line with Anne Witz's (1992) Neo-Weberian closure theory, which explains how different social groups attempt to increase their status by entering more privileged groups, or excluding subordinated groups (or both simultaneously). Witz extended Neo-Weberian closure theory to explain how professions and occupations are closed along gender and class lines. Patitsas (2019) used this theory to explain how computing went from female-dominated to male-dominated in Western countries: the field has had multiple boom-bust cycles of growth, and during boom times when educational resources were spread thin, CS departments and employers focused the field towards a stronger connection to formalized math and engineering perspectives and certifications. As a result, CS excluded the women and multidisciplinary perspectives that were common earlier in its history (Patitsas 2019). By the end of the PC boom and the dot-com boom, women's participation in CS programs in North America[3] dropped dramatically from 35 percent in the 1980s, to only 17 percent in the early 2000s (Patitsas 2019). Along with hegemonic norms that police and marginalize minoritized students within the field, CS developed a culture of toxic masculinity that persists (Archer et al. 2017; Breslin 2018; Harvey and Shepherd 2017).

Upholding the boundaries between social and technical knowledge therefore has important social consequences inside and outside the field. In response, feminist HCI scholars such as Lucy Suchman argued for technical practice that acknowledges the political and situated nature of design (Suchman 1993). Drawing on Donna Haraway, Suchman makes the case that developing useful and responsible technical systems is inherently a boundary-crossing activity, requiring collective knowledge from multiple locations across technology design and use.

Influenced by STS, researchers in HCI and CSCW have already started to do this boundary-crossing work via multidisciplinary and socially-integrated frameworks such as Value-Sensitive Design, Critical Design, and Reflective Design (Cockton 2005; Friedman, Kahn, and Borning 2002; Menendez-Blanco, Bjorn, and De Angeli 2017; Sengers et al. 2005). However, these approaches are not widely adopted in computing, instead favouring knowledge that is seen as (purely) technical.

[3] This is a Western phenomenon. As Mellström (2009) notes, gendered participation in computing is culturally situated, giving Malaysia as an example of where computer science continues to be female dominated.

2.3 Discourse Analysis

Given that we use discourse analysis for our analytic method, we would like to offer the reader some background on discourse analysis, which at its core, investigates the processes of social construction through the analysis of language and language use. There are many different approaches to discourse analysis, and we settled on using James Gee's (2011) model due to his clear, accessible language. According to Gee, discourse analysis studies how language, both spoken and written, enacts social and cultural perspectives and identities, given that "language has meaning only in and through social practices" (12). Discourse analysis typically can take either a "descriptive" or "critical" form with descriptive discourse analysis aiming to describe and deeply understand *how* language works and *why* it works that way (Gee 2011). Critical discourse analysis, on the other hand, is particularly interested in the relationship between language, ideology, and power, such as how ideology and power are discursively exercised through language (Oswick 2012).

Gee does not distinguish between descriptive and critical approaches to discourse analysis in his model because he claims that all discourse analysis is, in fact, critical discourse analysis since language itself is inherently political. By this, he means that language plays a pivotal role in the distribution of social goods as it is the key way "we build and sustain our world, cultures, and institutions" (Gee 2011, 10).

Gee asserts that the purpose of his discourse analysis is to explain, beyond description, how and why language works the way it does when it is put into action in a certain domain, in hopes of contributing to a greater understanding of important social, cultural, and political issues and facilitating stronger opportunities for intervention in those issues. His method entails asking questions about how language, at a given time and place, is used to construct and enact 7 “building tasks” using 6 “tools of inquiry”. The 7 “building tasks” of language are “seven things or seven areas of reality” that are constructed in the world whenever language is spoken or written: significance, practices, identities, relationships, politics, connections, and sign systems and knowledge (Gee 2011, 17). The 6 “tools of inquiry” are used to analyze the workings of each building task in specific instances where language is used: situated meanings, social languages, figured worlds, intertextuality, Discourses, and Conversations.

Gee’s discourse analysis uses each tool of inquiry to ask questions about each building task, which translates into asking 6 questions about 7 building tasks, amounting to a total of 42 guiding questions. He notes, however, that in practice, most discourse analyses following his method use only a subset of the 42 questions. This presents itself as an advantage of the method as it allows analysts to adopt a modular framework of questions. We provide now a reference on these modules:

Gee's Building Tasks Explained

	Building Task of Language
Significance	Significance relates to how language is used to render significance, lessen significance, and signal to others how we view their significance.
Identities	Identities refer to language that is used to build and enact identities.
Relationships	Relationships question what sort of relationship or relationships the language in the text is trying to enact with others.
Connections	Connections indicate how language is used to build connections or relevance, such as how a piece of language connects or disconnects things, or how a piece of language makes one thing relevant or irrelevant to another.
Sign systems and knowledge	Sign systems and knowledge refer to how language is used to build privilege or prestige for one sign system or way of knowing over another.

Gee's Tools of Inquiry Explained

	Tool of Inquiry
Social languages	Social languages refer to different varieties of language that allow us to express different socially significant identities and enact different socially meaningful practices and activities. Investigating how different social languages are used and mixed is one tool of inquiry for engaging in discourse analysis.
Figured worlds	Figured worlds are the (often unconscious) theories and stories used to understand and deal with the world, such as assumptions about the organization of society and culture. Questioning what typical stories or figured worlds the words and phrases of the text are assuming and inviting the audience to assume is another tool for engaging in discourse analysis.
Discourses	Discourses (with a capital D) signify ways of combining and integrating language, actions, interactions, ways of thinking, believing, valuing, and using various symbols, tools, and objects to enact a particular sort of socially recognizable identity. Thinking about the different Discourses a piece of language is part of is yet another tool for engaging in discourse analysis.

3 Methods

We chose to discursively analyze CC2020 due to it being the latest curriculum report released by the ACM and IEEE-CS. The ACM and IEEE-CS did release CS-specific curriculum guidelines in 2013, but we felt like CC2020 would provide a more updated snapshot of the ethics education landscape in computing, including CS, than a report released a decade ago.

Given the length of CC2020, our discourse analysis was an iterative process done over multiple rounds. We first familiarized ourselves with the table of contents and noted document sections that self-evidently stood out as relevant to the topic of socioethical education in CS. We did a surface read of these passages to confirm relevance and then scanned the rest of CC2020 using ethics-related keywords as search terms to get a sense of the document's discussion of ethics and social issues. Search terms included ethical, social, societal, political, environment, cultural, philosophy, sustainable, and responsible.

We then felt confident that we were able to pick out the most relevant and insightful passages from CC2020, of which there were three: a table visualizing the landscape of computing knowledge, a section on professionalism and ethics, and the social issues and professional practice knowledge area of the CS draft competencies.

Next, we did a deeper read of the three selected document sections and began to steep our analysis in Gee's 42 questions. We reflected on all the questions but decided to focus on the ones we felt were the most productive. Subsequently, we asked ourselves how the language of CC2020 rendered significance, enacted identities, signaled relationships, built connections, or built privilege for one sign system or way of knowing over another by using certain social languages, figured worlds, or Discourses. We took these initial findings and presented them to the rest of our lab, and then used our lab's feedback to do an additional, even deeper read of CC2020 in accordance with Gee's discourse analysis.

It is important to note that our discourse analysis is *gestalt,* meaning that we viewed and analyzed the text holistically rather than just an additive summation of its separate parts.

4 Results & Analysis

4.1 CC2020 Report Structure

CC2020 is an extensive 205-page document that addresses baccalaureate degree programs in 7 computing disciplines: Computer Engineering, Computer Science, Cybersecurity, Information Systems, Information Technology, Software Engineering, and Data Science. According to the CC2020 Task Force (2020), the objective of the report is to serve as "a guiding light for computing education worldwide" and to help "[computing] educators create effective curricula or improve the curricula they already have" (22).

The curriculum report is divided into 8 main chapters comprised of 42 sub-chapters in total, along with 10 appendices and a references section. Chapter 1 introduces CC2020's project expectations, background, and stakeholders. Chapter 2 addresses the computing landscape in general. Chapter 3 discusses the knowledge-based curricula guidelines from the last comprehensive computing curriculum report published in 2005 (CC2005). Chapter 4 then pushes for the adoption of competency-based computing education. Chapter 5 offers visual representations of computing curricula, which is where we found a revealing table that visualizes the landscape of computing knowledge areas in terms of relative importance (see table 1). Chapter 6 focuses on "global and professional considerations" by discussing worldwide computing nomenclature and degree structures as well as professionalism and ethics in its last sub-chapter.

Out of the entire document, only one sub-chapter out of the 42 sub-chapters in CC2020 is explicitly dedicated to discussing ethics education in computing[4]. This apparent structural marginalization of ethics education in CC2020 reinforces Fiesler's (2018) argument that ethics in CS education is often treated as a separate afterthought rather than an integral pillar of the discipline. It also fits with analyses of ideology in computing that "social" and "technical" are seen as distinct and disconnected (Agre 1998; Breslin 2018; Dourish 2019; Ehn 1988; McPherson 2011).

Chapter 7 of CC2020 subsequently examines the challenges and opportunities faced by institutions in revising and adapting their computing curricula design, and Chapter 8 concludes by looking beyond the CC2020 report.

Out of the 10 appendices, the third appendix (Appendix C) gives examples of preliminary draft competencies for 5 computing disciplines: Computer Engineering, Computer Science, Information Systems, Information Technology, and Software Engineering. Among the CS draft competencies, we looked at the 18 knowledge areas, which are "typically…silo[es] of related material" (Raj and Kumar 2022). We then identified the social issues and professional practice knowledge area as another key passage of interest.

[4] The results of our analysis of this sub-chapter are in section 4.3.

4.2 Visualization of the Landscape of Computing Knowledge Table

The first relevant section we found in CC2020 was the aforementioned table in Chapter 5 which visually represents the computing knowledge landscape by displaying values that signify the relative importance of each computing knowledge area in each of the 6 included computing disciplines for undergraduate degree programs (see table 1).

Table 1. ***Landscape of Computing Knowledge***

		CE		CS		CSEC		IS		IT		SE	
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
1. Users and Organizations	1.1. Social Issues and Professional Practice	2	5	2	4	2	4	3	5	2	4	3	5
	1.2. Security Policy and Management	1	3	2	3	4	5	2	3	2	4	2	4
	1.3. IS Management and Leadership	0	2	0	2	1	2	4	5	1	2	1	2
	1.4. Enterprise Architecture	0	1	0	1	1	2	3	5	1	3	1	3
	1.5. Project Management	1	3	2	3	1	2	4	5	2	3	2	4
	1.6. User Experience Design	1	3	2	4	1	3	2	4	3	4	3	5
2. Systems Modeling	2.1. Security Issues and Principles	2	3	2	3	4	5	2	4	3	4	2	4
	2.2. Systems Analysis & Design	1	2	1	2	1	2	4	5	1	3	2	4
	2.3. Requirements Analysis and Specification	1	2	1	2	0	2	2	4	1	3	3	5
	2.4. Data and Information Management	1	2	2	4	2	3	3	5	2	3	2	4
3. Systems Architecture and Infrastructure	3.1. Virtual Systems and Services	1	3	1	3	1	2	1	2	3	4	1	3
	3.2. Intelligent Systems (AI)	1	3	3	5	1	2	1	2	1	2	0	1
	3.3. Internet of Things	2	4	0	2	1	3	1	3	2	4	1	3
	3.4. Parallel and Distributed Computing	2	4	2	4	1	2	1	3	1	3	2	3
	3.5. Computer Networks	2	4	2	4	2	4	1	3	3	4	2	2
	3.6. Embedded Systems	3	5	0	2	1	3	0	1	0	1	0	3
	3.7. Integrated Systems Technology	1	2	0	2	0	2	1	3	3	4	1	3
	3.8. Platform Technologies	0	1	1	2	1	2	1	3	2	4	0	2
	3.9. Security Technology and Implementation	2	3	2	4	4	5	1	3	2	4	2	4
4. Software Development	4.1. Software Quality, Verification and Validation	1	3	1	3	1	2	1	3	1	2	3	5
	4.2. Software Process	1	2	1	3	0	2	1	3	1	3	3	5
	4.3. Software Modeling and Analysis	1	3	1	3	1	2	2	4	1	3	4	5
	4.4. Software Design	2	4	2	4	1	3	1	3	1	2	4	5
	4.5. Platform-Based Development	0	2	2	4	0	1	1	3	2	4	1	3
5. Software Fundamentals	5.1. Graphics and Visualization	1	2	2	4	0	1	1	1	0	1	0	2
	5.2. Operating Systems	2	4	3	5	2	3	1	2	1	3	1	3
	5.3. Data Structures, Algorithms and Complexity	2	4	4	5	1	3	1	3	1	2	2	4
	5.4. Programming Languages	2	3	3	5	1	2	1	2	1	2	2	3
	5.5. Programming Fundamentals	2	4	4	5	2	3	1	3	2	4	3	5
	5.6. Computing Systems Fundamentals	2	3	2	3	1	2	2	3	1	3	2	3
6. Hardware	6.1. Architecture and Organization	4	5	3	4	1	3	1	2	1	2	1	3
	6.2. Digital Design	4	5	1	2	0	2	0	1	0	1	0	2
	6.3. Circuits and Electronics	4	5	1	2	0	1	0	1	1	2	0	1
	6.4. Signal Processing	3	4	0	1	0	2	0	1	0	1	0	1

(CC2020 Task Force 2020, 64)

These values are based on the “expert opinions” of the steering committee for CC2020 rather than a representative survey of the disciplines (CC2020 Task Force 2020, 64). Steering committee members were asked to rate the importance of each knowledge area included in the curriculum recommendation for the following six disciplines: Computer Engineering, Computer Science, Cybersecurity, Information Systems, Information Technology, and Software Engineering. The values displayed in the table are the rounded average of the responses.

4.2.1 Sign Systems and Knowledge in the Computing Knowledge Table

In our discourse analysis of the table, we found this table to be a clear visual representation of the building task of sign systems and knowledge according to Gee. By having steering committee members rate the relative importance of each knowledge area in CS curricula, we have clear indicators of how relatively privileged or disprivileged certain ways of knowing in CS are.

We noticed that the social issues and professional practice knowledge area in CS was rated reasonably well in importance by the CC2020 steering committee with an average minimum rating of 2 and an average maximum rating of 4 on an importance scale of 0-5. However, adjacent knowledge areas critically related to the social issues and professional practice knowledge area such as “Project Management” (average minimum rating of 2 and average maximum rating of 3), or “Requirements Analysis and Specification” (average minimum rating of 1 and average maximum rating of 2) had their importance rated quite low in comparison (CC2020 Task Force 2020, 64).

Given that both project management and requirements analysis and specification happen in professional contexts deeply affected by social forces and dynamics, this gave us the impression of a rather superficial emphasis placed on the importance of social and professional issues. This reflects the tendency to render social issues technical, by presenting them as separate problems to be solved by (and not within) computing practice.

Moreover, areas of computing that are seen as less technical (and therefore more feminine) — such as project management, and requirements analysis and specification — are explicitly devalued in CS education. This falls in line with a historical practice of occupational closure[5] (Witz 1992), where subfields of computing that become more feminized are cast off from the discipline in order to maintain the prestige of "hard science" and the masculinity of the field (Patitsas 2019).

Consistent with CS's history of gendered occupational closure, it is noticeable that the only computing major in the CC2020 report that is given high weights on project management and requirements analysis is Information Systems (IS). Work from Cukier et al. has documented how computer science has supported the creation of IS as a separate field into which they shunted work that was seen as too "soft" or feminine to be "real computer science" (Cukier, Shortt, and Devine 2002; Cukier 2003; Cukier et al. 2009; Patitsas 2019; Sturman 2009). This is an old tactic in computing, which from its very early days has created and re-created gendered divisions of computing labour (Abbate 2012; Hicks 2017; Patitsas 2019).

Thus, even though the social issues and professional practice knowledge area in CS is nominally rated high in importance by the CC2020 steering committee, education regarding social issues and professional practice, such as socioethical education, is not actually epistemically valued or privileged *in practice*, as seen by the disparity between the social issues and professional practice knowledge area rating and the ratings of knowledge areas that are sites where social issues and professional practice are most obvious.

[5] Occupational closure is explained in section 2.2.

4.3 The Sub-Chapter on Professionalism and Ethics

The second relevant section of CC2020 was the sub-chapter on professionalism and ethics. As mentioned earlier, it was notable that out of CC2020's forty-two sub-chapters and ten appendices, there was only one sub-chapter explicitly dedicated to socioethical content and that it was recycled "from the IT2017 report" (CC2020 Task Force 2020, 76). We performed a discourse analysis on the following section of text from the sub-chapter:

Table 2. ***Professionalism and Ethics Sub-Chapter***

Additional material such as computer history, digital libraries, techniques for tackling ill-defined problems, teamwork with individual accountability, real-life ethical issues, professional standards and guidelines, legal constraints and requirements, and the philosophical basis for ethical arguments may also appear either in a dedicated course or distributed throughout the curriculum. The distributed approach has the advantage of presenting this material in the context of a real application area. On the other hand, the distributed approach can be problematic in that faculty often minimize professionalism and ethics in the scramble to find adequate time for the technical material.

(CC2020 Task Force 2020, 76)

We found two salient building tasks: connections and sign systems and knowledge.

4.3.1 Connections in the P&E Sub-Chapter

While this sub-chapter of CC2020 (see table 2) does demonstrate an awareness that ethics needs to be explicitly incorporated into computing curricula, there is no suggestion of integrating or embedding socioethical content into technical material. CC2020 offers either a "dedicated course" or a "distributed approach" for ethics education. Although the "distributed approach" is still an improvement compared to a "dedicated course", there is no language that suggests the distributed approach connects ethics to technical material in the curricula. The language only suggests that ethics is "distributed throughout the curriculum" which is not the same as explicitly connecting ethics to technical content and reframing CS as a complex sociotechnical domain.

Per Gee, connections are not only about how a piece of language connects things, but also how it disconnects things. In this case, socioethical content is still disconnected from "technical material" in the distributed approach, as seen by the concern that "faculty [will] often minimize professionalism and ethics in the scramble to find time for the technical material". This apparently common sidelining of professionalism and ethics in "the scramble to find adequate time for the technical material" speaks to a hegemonic structural division that continues to disconnect the social from the technical in CS curricula design, which comes as a result of CS curricula being rendered technical and regarding the social and the technical material as separate rather than inextricably linked.

Another (dis)connection at play here is that the ACM and IEEE felt a new curriculum report was needed for a great deal of *technical* material, but not for the socioethical content. They are open that they recycled this content "from the IT2017 report" (CC2020 Task Force 2020, 76). Not only is socioethical content disconnected from the technical content, it is treated as static, as though professional and ethical concerns in the profession have not changed during a time when we have seen intensification in the use of machine learning in positions of power, the gigification of the workforce, the power of social media websites, and the devastation of climate change (Ahmed 2018; Angwin et al. 2016; BBC News 2020; Dao et al. 2022; Greenfield 2018;

Hill 2014, 2020; Metz and Blumenthal 2019; Quinn and Ball 2014; The New York Times 2019; Vaidhyanathan 2018).

4.3.2 Sign Systems and Knowledge in the P&E Sub-Chapter

This section of the professionalism and ethics sub-chapter also constructs another building task: sign systems and knowledge. As mentioned above, CC2020 explicitly warns against the possibility of "faculty often minimiz[ing] professionalism and ethics in the scramble to find adequate time for the technical material" in the distributed approach to teaching ethics. First, we would like to point out that professionalism and ethics would not be minimized if they were taught inextricably from the technical material. Also, this "scramble" by "faculty" not only indicates that CS curricula are still rendered technical in the distributed approach, but also that "technical material" as a way of knowing is privileged compared to "professionalism and ethics". The language used by the ACM/IEEE-CS conveys an implicit prioritization of technical material over professionalism and ethics, and also seems to suggest by the faculty's "scramble to find adequate time" that socioethical material is chaotically positioned in an oppositional zero-sum game with technical material. This is in spite of evidence that teaching technical material can be *enriched* by situating it within broader social, cultural, political, and historical contexts (Almstrum et al. 2008; Goldweber et al. 2012; Impagliazzo et al. 1999; Mayhew 2023; M. J. Quinn 2006; Rupf 2004; Young et al. 2009).

4.4 Computer Science Draft Competencies – SP (Social Issues and Professional Practice)

Finally, we looked at the draft competencies for CS, specifically those in the social issues and professional practice knowledge area (see table 3). We found this knowledge area of CS curricula evoked 4 of Gee's 7 building tasks of language: significance, identities, relationships, and connections.

Table 3. ***SP - Social Issues and Professional Practice***

a. Perform a system analysis for a local organization and present the results to them in a non-technical way. b. Integrate interdisciplinary knowledge to develop a program for a local organization. c. Document industry trends, innovations, and new technologies and produce a report to influence a targeted workspace. d. Present to a group of professionals an innovative computer system by using audience-specific language and examples to illustrate the group's needs. e. Produce a document that is helpful to others that addresses the effect of societal change due to technology. f. Adopt processes to track customer requests, needs, and satisfaction. g. Compare different error detection and correction methods for their data overhead, implementation complexity, and relative execution time for encoding, detecting, and correcting errors and ensure that any error does not affect humans adversely.

(CC2020 Task Force 2020, 114)

4.4.1 Significance in the S&P Competency

We found that the social issues and professional practice knowledge area downplays the real-world significance of technological harms by using *clinical* and *impersonal social language*.

Instead of addressing or acknowledging potential technological *harms*, the word "error" is instead used repeatedly such as "error detection" and "correcting errors", framing the possibility of enacting harmful technologies as if they were mere miscalculations on a lab report. Harm implies that there is a recipient to the harm who is *harmed*. Error, on the other hand, connotes abstract inaccuracy.

The clinical and impersonal social language continues by stating “Correct[] errors and ensure that any error does not affect humans adversely” (CC2020 Task Force 2020, 114). Specifying *humans*, rather than substituting in a similar word like *society*, also contributes to broadening the distance between the technology and its real-world social impacts. Forgoing the word choice of “society” speaks to the absence of a sense of collective responsibility and co-habitation that the word “society” would connote. “Adversely” also reminded us of the language generally used to describe weather conditions that are impersonal, uncontrollable forces of nature rather than the very calculated, human-made societal consequences of technology.

We observe two ideologies driving this language. Firstly, the emphasis on humans and individuals rather than societal and structural issues reflects a liberal ideology. Past work, such as by Coleman and Golub (2008) documents the role of liberalism in the tech world and how multiple variants of liberalism are infused in the hacker ethos. Training computer scientists to think of individuals rather than society has been implicated as contributing to the negative social effects of computing — such as its role in colonialism and eugenics (Halmaghi 2020; Williams et al. 2022).

Secondly, the clinical, impersonal aesthetic reflects scientism. At first glance this may be run-of-the-mill scientism: playing to a larger societal belief that science is the only legitimate source of knowledge, which STS scholars have linked to depoliticization of socio-scientific issues and stifling political mobilization (Kimura and Kinchy 2016).

But there is more than mere scientism at play given that CS is a young discipline that is still trying to prove itself as “science” (Abbate 2016). The field has a teenager-like self-awareness that it is “not about computers nor is it science” (globular-toast 2021), and has routine debates about it having stronger epistemic rooting in applied mathematics and engineering (CACM Staff 2013; Krebsbach 2015; Tedre and Sutinen 2008).

Perhaps this is unsurprising: the very name “computer science” caught on in the 1960s less due to epistemic/methodological alignment with science, and more for political purposes. In particular, the name “computer science” was to convince the US National Science Foundation to treat the area as a “scientific field on par with other disciplines” (Abbate 2016). The framing of computing as “science” further allowed CS to escape the regulations of engineering (Moore 2020; Wagner 2019), and allowed the greater occupational prestige that came from masculinizing. The women who dominated computing in the mid-20th century could thus be dismissed as mere "programmers" and "computer operators" while the men could distinguish themselves as computer *scientists* (Abbate 2012; Patitsas 2019). And as we discussed in section 4.2.1, it allowed another gendered division in the late-20th century between CS and Information Systems.

4.4.2 Identities in the S&P Competency

We noticed that the social issues and professional practice knowledge area of CS in CC2020 enacts a technical, authoritative CS identity vs. a non-technical non-CS identity in need of greater CS expertise through the figured world of technical non-intelligibility vs. non-technical intelligibility. We observed that there is an assumption that a “local organization” would need technical results “present[ed]...to them in a non-technical way” and that, in general, local organizations needed to be guided by CS undergraduate students and have technology translated for them (CC2020 Task Force 2020, 114).

This positions CS undergraduate students as the higher authority on technology vs. the “non-technical” local organization. By doing so, this firstly conflates technical expertise in CS with being an authority on technologies in general. Secondly, it presumes only technical content is non-intelligible and needs to be simplified for non-technical outsiders when in reality, many non-technical domains have just as much complexity and jargon as technical content. While building and enacting a technical, authoritative CS identity is likely meant to position CS students as a helpful source of expertise, it also further confines CS identities as mainly technical rather than sociotechnical.

It is also of note that the competencies deal only with the *production* of technical content— no competencies touch on the questions of refusal or abstention from unethical technologies. The computer scientist is positioned as somebody who merely analyzes and creates technologies. Although the computer scientist is identified as an authority, the power to refuse (Barabas 2022; Baumer et al. 2014) is not in their sanctioned arsenal.

4.4.3 Relationships in the S&P Competency

The relationship the social issues and professional practice knowledge area of CS is trying to enact is a magnanimous relationship of charitable service to the local community. Not only are CS students expected to "perform a system analysis for a local organization and present the results to them in a non-technical way", but they also should "integrate interdisciplinary knowledge to develop a program for a local organization" and "produce a document that is helpful to others that addresses the effect of societal change due to technology" (CC2020 Task Force 2020, 114). In addition, CS students are asked to "produce a report to influence a targeted workspace" on "industry trends, innovations, and new technologies" and to "illustrate the…needs" of "a group of professionals" by presenting to them "an innovative computer system" (CC2020 Task Force 2020, 114).

In the report's language, we see evidence of the CS students' relationship to the local organizations resembling the hierarchical technology designer-user relationship that has been heavily critiqued by Suchman (1993), who eviscerates this idea that the designers and producers of technical systems are impartial actors doing good for the user. Suchman, in fact, argues that the designer-user dichotomy is a false dichotomy that "masks…an increasingly dense and differentiated layering of people and activities" (30). Similar to the previous building task of identities, the relationships discursively enacted by the social issues and professional practice knowledge area position the local non-CS community as in obvious need of the technological benevolence and greater know-how of CS students. Suchman, however, contends that "a problem that underlies the persistence of boundaries between design and use is the premise that technical expertise is the necessary, if not sufficient, form of knowledge" (24).

While presumably, the local community organizations would consent first to the CS students invoking their expertise, we find it notable that this is not overtly stated, as if there is no question that CS students would be welcomed into all professional environments to enact this philanthropic relationship undergirded by a troubling sense of superiority that entitles CS students to "influence" and "illustrate the group's needs" for them (CC2020 Task Force 2020, 144). This magnanimous relationship between CS students and local organizations is reminiscent of another hierarchical relationship found commonly between Western academic technology researchers and underserved communities that has been critiqued by Nathan et al. (2017). They specifically critique the research practice of Western-trained Human-Computer Interaction (HCI) academics who positions themselves in local community partnerships as an "all knowing entity" that solely has "the expertise and experience necessary to identify what is of benefit to all" (Nathan et al. 2017, 1).

The handouts enacted by this relationship between CS students and local organizations may seem superficially appealing for some and mostly innocuous, but should mandating individual student altruism really form the bedrock of the social issues and professional practice knowledge area in CS curricula? Is that what really should constitute a knowledge area explicitly named social issues and professional practice? Aside from "produc[ing] a document that is helpful to others that addresses the effect of societal change due to technology", the social issues and professional practice knowledge area generally neglects to acknowledge the greater social, political, and environmental issues related to technology and instead focuses on encouraging students' aptitude for techno-solutionism by expecting all their interventions in the assumed problems of local organizations to be technically focused like a "systems analysis" or "program" or "an innovative computer system" (CC2020 Task Force 2020, 114). As such, this emphasis on technical interventions is yet another example of CS curricula being rendered technical.

4.4.4 Connections in the S&P Competency

We noted that the social issues and professional practice knowledge area connects social issues and professional practice as related through the Discourse of technology acting as assistive and helpful, which effaces the competing Discourse of harmful technologies. For instance, we observed that the social issues and professional practice knowledge area only nominally addresses social issues by lumping together social issues with professional practice. Most of the competencies for the social issues and professional practice knowledge area address matters related to professional practice rather than explicit social issues, so we found it odd that social issues and professionalism were connected until we realized that the Discourse of technologies being good, assistive, and helpful was essential for that connection. The Discourse of technology being used for only good neutralizes any legitimate concerns one may have about technologies' harmful effects on social issues, permitting the focus to largely be on professional issues instead. It allows the "social issues" related to CS to be framed mainly as issues regarding professionalism and interaction with external non-CS actors.

By painting technology as ubiquitously "helpful to others" and assistive rather than possibly harmful, the social issues and professional practice knowledge area lets social issues be subsumed under professional practice rather than its own distinctive knowledge area that would address the potential social harms of technology.

Another connection the social issues and professional practice knowledge area makes is how it makes the commercial applications of technology relevant to social issues and professional practice. CS students are expected to "adopt processes to track customer requests, needs, and satisfaction" as one of their social issues and professional practice competencies (CC2020 Task Force 2020, 144). This language directly connects the knowledge area of social issues and professional practice to optimizing customer relations for businesses. In our view as scholars who are interested in encouraging socially responsible CS education and equitable uses of technologies, we question whether this type of connection is appropriate or desirable to make as it conceives of social responsibility and professionalism in business-oriented commercial terms rather than the development of a critical sociological awareness.

5 Discussion

5.1 Reflexivity

Given the nature of our methods, we feel it important to be transparent to ourselves and others about our positionality and how it affected our analysis. Upon reflection, we realized that Anna's background as an American impacted her conception of CS vs. software engineering

(SE), as CS and SE are regulated differently in the States than in Canada. Elizabeth, who was trained in CS at Canadian universities, knew well that SE is regulated as a licensed profession in Canada and separate from CS, which is not regulated as such. Whereas Anna had the two disciplines more conflated as the disciplinary boundaries are more blurred in the States, as only a handful of states regulate SE as a profession. Anna was surprised by the relative lack of discussion of social and environmental impacts in CS, as she had expected due to her perception of CS and SE overlapping and her partially complete knowledge that SE in Canada does require some degree of socioethical education.

While having taken some introductory programming classes and workshops, Anna's academic background is largely outside of CS and that influenced her analysis. For instance, as a disciplinary outsider who has not completed a formal CS degree, Anna was bewildered at what sort of alleged bearing "customer requests, needs, and satisfaction" could possibly have on socioethical education in CS (CC2020 Task Force 2020, 144). Elizabeth, however, as a trained computer scientist, was not surprised to observe the explicit connection made between the social concerns of technology and professionalism in CS and the commercialism that emerges from our discourse analysis. From her experience, Elizabeth was already sensitive to how computing dismisses "soft" parts of the field as "not real CS" and that it "belongs in the iSchool" instead.

Anna's outsider status also made her more sensitive to the techno-paternalism found in the draft social issues and professional practice competencies suggested for CS. As an insider to the discipline, Elizabeth did not immediately see how the language of the section enacted a magnanimous relationship of charity between CS students and the local community, yet Anna as a non-technical outsider, was more attuned to how technical expertise is often implicitly viewed as a stand-in for competence and sound judgment even in non-technical contexts, resulting in a hierarchical relationship between those with technical know-how and those without.

Anna’s training in STS played a role in our decision to employ rendering technical as our theoretical framework rather than similar frameworks like algorithmic formalism that have emerged from CS.

Hana joined the paper after the initial analysis was complete and Anna presented the preliminary findings to the lab. Although Hana did not contribute greatly to the analysis, she found the results of this study consistent with her own experience as a recent graduate of an undergraduate CS program in Canada. In addition to the formal CS curriculum (which did not include any ethics courses or content), Hana found the tendency to render technical and distinguish computer scientists from users and local communities reflected in informal spaces such as hackathons.

5.2 Comparison to Other Conceptual Frameworks

As we mentioned, there are other complementary conceptual frameworks that we could have used to analyze the representation of socioethical education in CC2020. We found that rendering technical was more useful for our analysis than the other frameworks available to us; here we briefly discuss the alternatives.

5.2.1 Technochauvinism

As coined by Broussard (2018), technochauvinism “is the belief that tech is always the solution” (7-8). In other words, technochauvinism perceives technology as the primary solution to societal problems. Given that the CC2020 report displays a tendency towards technochauvinism in the expectation that CS students perform system analyses, develop programs, and produce innovative computer systems for local organizations, we would be remiss if we did not consider the utility of technochauvinism as an analytic lens. Similar to rendering technical, technochauvinism leads us to critique the curricular emphasis on technosolutionist competencies for CS students. These technochauvinist/solutionist competencies overshadow the broader sociopolitical implications of technology use and further relegate the importance of social issues to the background.

Although we find that technochauvinism was easy to identify and critique in CC2020, technochauvinism by itself does not *explain* the tendency to elevate technical solutions, whereas rendering technical provides an underlying epistemic explanation for this behavior in CS students. We found an advantage of rendering technical is that it provides a framework for understanding *why* the social is often partitioned from the technical in CS. To us, technochauvinism is the natural consequence of CS curricula being rendered technical and limited to only technical interventions.

5.2.2 Algorithmic Formalism

Green and Viljoen (2020) explore the idea of algorithmic formalism, which is a theoretical framework that is similar to Breslin's (2018) rendering technical. They use the term algorithmic formalism to characterize the dominant way of reasoning within CS that involves the three key orientations of objectivity/neutrality, internalism, and universalism. These three paradigms lead to algorithmic interventions that "entrench existing social conditions, narrow the range of possible reforms, and impose algorithmic logics at the expense of others" (Green and Viljoen 2020, 20). Like Breslin's rendering technical, algorithmic formalism epistemically and methodologically underpins the CS discipline, as computer scientists interpret the world and conceive of problems and solutions via algorithmic thinking and algorithmic interventions that uncritically employ “formal methods in complex social situations where their assumptions may be invalid” (21). Given the similarities between the two frameworks, our first impression of algorithmic formalism was admittedly that of rendering technical in a trenchcoat.

Our second impression, however, was that algorithmic formalism gives us a more granular framework than rendering technical because it enumerates three pathways of how CS is rendered technical or formalized algorithmically: objectivity/neutrality, internalism, and universalism (Green and Viljoen 2020). Having three specific pathways to focus on can be valuable— and we can see how it would have been useful in some of our own prior work.

However, these three sources of algorithmic formalism were created through analogous reasoning to legal formalism (Green and Viljoen 2020), and as a result may miss aspects of computing ideology that are absent in law. For example, our analysis of CC2020 yielded a recurrent theme of viewing social and technical as separate and non-overlapping, which is a phenomenon observed by other scholars (Agre 1998; Breslin 2018; Dourish 2019; Ehn 1988; McPherson 2011). This is not explicitly part of Green and Viljoen’s framework. It is implied to be a problem in their discussions of objectivity/neutrality and internalism but does not rise to a level of being a conceptual tool in their framework.

Also absent from Green and Viljoen’s framework is the push for CS to prove itself as a profession. Unlike law, computer science is a young field, and its status as a (semi-)profession continues to be in flux (Misa 2016; Patitsas 2019; Tijdens 1997). Green and Viljoen’s framework usefully calls attention to the performance of objectivity/neutrality in computing. However, they miss the motivations beyond basic scientism which we observed in our analysis, such as the role gender plays in the professionalization of computing (Abbate 2012; Hicks 2017; Patitsas 2019; Tijdens 1997). In addition to objectivity/neutrality, internalism, and universalism, we found that the CC2020 report reflected the gendered nature of CS's claims to legitimacy, and its attempts to increase its status by demonstrating aspects of a true profession (e.g. a code of ethics, a licensing process, etc) without being regulated like a profession (Moore 2020; Wagner 2019).

So although rendering technical and algorithmic formalism share many commonalities, rendering technical still proved more useful for our curricular analysis, likely due it being developed with ethnographic data in an educational setting.

While the two frameworks converge on some insights, they highlight different aspects of computing: while internalism did not emerge strongly in our analysis we can see how it would be analytically useful in studying how technologists create their technologies.

5.3 Future Work

We invite future work to further examine how these three orientations of algorithmic formalism manifest in CS curricula and pedagogy. We encourage future work on the connection between techno-optimism (Avle et al. 2020) and the conceptual frameworks we discussed.

In both Breslin's (2018) and Li's (2007) work on rendering technical, there is a contrasting concept of rendering natural. This has been used, for example, to talk about how patterns and heuristics in computing are presented as though they are laws of nature (Mayhew and Patitsas 2021). We can see how it would be generally useful, such as in analysing how ubiquitous computing aims to make computing "invisible" (Dourish and Bell 2011; Kerasidou 2019) and machine learning technologies are treated as "enchanted" (Campolo and Crawford 2020), and future work should consider how rendering natural appears in CS curricula.

6 Conclusion

We examined the latest ACM/IEEE-CS curricula recommendation (CC2020) using Gee's model of discourse analysis that considers how language enacts certain sociocultural perspectives and identities. Through the concept of rendering technical as developed by Breslin (2018), we critiqued how current flawed models of CS education fail to integrate socially responsible computing content into technical content, thereby socializing students to view social, ethical, and environmental consequences as disconnected from their work as computing professionals.

Our analysis reveals how scientism is still active in CS through the use of clinical and impersonal social language in CC2020. We also demonstrate that despite there being an argument for ethics to be technically integrated into CS education (e.g. Fiesler 2018), CC2020 structures the socioethical content as non-integrated and as an afterthought. Not only is ethics positioned as an afterthought, but it is also chaotically positioned against technical content in a zero-sum competition for lecture time.

Other contributions include illustrating how CS curricula design makes superficial commitments to social issues and professional practice competencies by highlighting that the related competencies of project management and requirements engineering are not ranked highly in importance. We also outline how the social issues and professional practice competencies in CS enact a magnanimous relationship of handouts between CS students and local community organizations that impart a sense of epistemic superiority to CS students. This relationship also showcases how the false user/designer dichotomy critiqued by Suchman (1993) still goes unchallenged today in CS curricula.

Finally, we compare and contrast three theoretical frameworks for analyzing ideology in computing: technochauvinism (Broussard 2018), rendering technical (Breslin 2018), and algorithmic formalism (Green and Viljoen 2020). We found that technochauvinism is easy to spot, but does not give us as much explanation for *why* technologists behave this way as rendering technical does. Algorithmic formalism in contrast gives three sources of this sort of behavior: neutrality/objectivity, internalism, and universalism. Although we find all three of these at work in the CC2020 report, it misses the gendered aspects in a way that rendering technical allowed us to identify and explore.

References

Abbate, Janet. 2012. *Recoding Gender: Women's Changing Participation in Computing*. History of Computing. Cambridge, Mass.: MIT Press.

———. 2016. "From Handmaiden to 'Proper Intellectual Discipline': Creating a Scientific Identity for Computer Science in the 1960s America." In *Communities of Computing: Computer Science and Society in the ACM*, by Thomas J. Misa. New York, NY, USA: Association for Computing Machinery and Morgan & Claypool. https://doi.org/10.1145/2973856.

Agre, Philip E. 1998. "Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI." In *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide*, edited by Geoffrey Bowker, Susan Leigh Star, Les Gasser, and William Turner. New York: Psychology Press.

Ahmed, Nafeez. 2018. "Pentagon Wants to Predict Anti-Trump Protests Using Social Media Surveillance." *Vice* (blog). 2018. https://www.vice.com/en/article/7x3g4x/pentagon-wants-to-predict-anti-trump-protests-using-social-media-surveillance.

Almstrum, Vicki L., Bill Aspray, Mary Anne Egan, and John Impagliazzo. 2008. "Perspectives on History in Computing and Education." *ACM SIGCSE Bulletin* 40 (1): 2–3. https://doi.org/10.1145/1352322.1352139.

Amrute, Sareeta. 2020. "Bored Techies Being Casually Racist: Race as Algorithm." *Science, Technology, & Human Values* 45 (5): 903–33. https://doi.org/10.1177/0162243920912824.

Angwin, Julia, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. "Machine Bias." ProPublica. 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Archer, Louise, Julie Moote, Becky Francis, Jennifer DeWitt, and Lucy Yeomans. 2017. "The 'Exceptional' Physics Girl: A Sociological Analysis of Multimethod Data From Young Women Aged 10–16 to Explore Gendered Patterns of Post-16 Participation." *American Educational Research Journal* 54 (1): 88–126. https://doi.org/10.3102/0002831216678379.

Avle, Seyram, Cindy Lin, Jean Hardy, and Silvia Lindtner. 2020. "Scaling Techno-Optimistic Visions." *Engaging Science, Technology, and Society* 6 (May): 237–54. https://doi.org/10.17351/ests2020.283.

Barabas, Chelsea. 2022. "Refusal in Data Ethics: Re-Imagining the Code Beneath the Code of Computation in the Carceral State." *Engaging Science, Technology, and Society* 8 (2).

https://doi.org/10.17351/ests2022.1233.

Bardzell, Shaowen. 2010. "Feminist HCI: Taking Stock and Outlining an Agenda for Design." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1301–10. Atlanta Georgia USA: ACM. https://doi.org/10.1145/1753326.1753521.

Baumer, Eric P.S., Morgan G. Ames, Jed R. Brubaker, Jenna Burrell, and Paul Dourish. 2014. "Refusing, Limiting, Departing: Why We Should Study Technology Non-Use." In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, 65–68. Toronto Ontario Canada: ACM. https://doi.org/10.1145/2559206.2559224.

BBC News. 2020. "Mohsen Fakhrizadeh: 'Machine-Gun with AI' Used to Kill Iran Scientist." *BBC News*, 2020, sec. Middle East. https://www.bbc.com/news/world-middle-east-55214359.

Breslin, Samantha. 2018. "The Making of Computer Scientists: Rendering Technical Knowledge, Gender, and Entrepreneurialism in Singapore." Doctoral dissertation, Memorial University of Newfoundland. 13499. Memorial University Research Repository.

Broussard, Meredith. 2018. *Artificial Unintelligence: How Computers Misunderstand the World*. Cambridge, Massachusetts London, England: The MIT Press.

Burtscher, Martin, Wuxu Peng, Apan Qasem, Hongchi Shi, Dan Tamir, and Heather Thiry. 2015. "A Module-Based Approach to Adopting the 2013 ACM Curricular Recommendations on Parallel Computing." In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education - SIGCSE '15*, 36–41. Kansas City, Missouri, USA: ACM Press. https://doi.org/10.1145/2676723.2677270.

CACM Staff. 2013. "Computer Science Is Not a Science." *Communications of the ACM* 56 (1): 8–9. https://doi.org/10.1145/2398356.2398359.

Campolo, Alexander, and Kate Crawford. 2020. "Enchanted Determinism: Power without Responsibility in Artificial Intelligence." *Engaging Science, Technology, and Society* 6 (January): 1–19. https://doi.org/10.17351/ests2020.277.

CC2020 Task Force. 2020. *Computing Curricula 2020: Paradigms for Global Computing Education*. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3467967.

Checkland, Peter, and John Poulter. 2020. "Soft Systems Methodology." In *Systems Approaches to Making Change: A Practical Guide*, edited by Martin Reynolds and Sue Holwell (Retired), 201–53. London: Springer London. https://doi.org/10.1007/978-1-4471-7472-1_5.

Chun, Rene. 2018. "China's New Frontiers in Dystopian Tech." The Atlantic. 2018.

https://www.theatlantic.com/magazine/archive/2018/04/big-in-china-machines-that-scan-your-face/554075/.

Cockton, Gilbert. 2005. "A Development Framework for Value-Centred Design." In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, 1292–95. Portland OR USA: ACM. https://doi.org/10.1145/1056808.1056899.

Coleman, E. Gabriella, and Alex Golub. 2008. "Hacker Practice: Moral Genres and the Cultural Articulation of Liberalism." *Anthropological Theory* 8 (3): 255–77. https://doi.org/10.1177/1463499608093814.

"Computer-Supported Cooperative Work." 2023. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Computer-supported_cooperative_work&oldid=1169295152.

Cukier, Wendy. 2003. "Constructing the IT Skills Shortage in Canada: The Implications of Institutional Discourse and Practices for the Participation of Women." In *Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia--Leveraging Differences and Diversity in the IT Workforce*, 24–33. Philadelphia Pennsylvania: ACM. https://doi.org/10.1145/761849.761853.

Cukier, Wendy, Denise Shortt, and Irene Devine. 2002. "Gender and Information Technology: Implications of Definitions." *ACM SIGCSE Bulletin* 34 (4): 142–48. https://doi.org/10.1145/820127.820188.

Cukier, Wendy, Margaret Yap, Mark Holmes, and Sara Rodrigues. 2009. "Diversity and the Skills Shortage in the Canadian Information and Communications Technology Sector: A Critical Interrogation of Discourse." *I-PROF 2009*, 1–11.

Dao, David, W. Hugo, Alf-André Walla, VocalFan, Eric Rubiel, Fabian Schreiber, Jan Jaworski, et al. 2022. "Daviddao/Awful-Ai: Awful AI - 2021 Edition." Zenodo. https://doi.org/10.5281/ZENODO.5855972.

Dourish, Paul. 2019. "User Experience as Legitimacy Trap." *Interactions* 26 (6): 46–49. https://doi.org/10.1145/3358908.

Dourish, Paul, and Genevieve Bell. 2011. *Divining a Digital Future: Mess and Mythology in Ubiquitous Computing*. Cambridge, Mass. London: MIT Press.

Dziallas, Sebastian, and Sally Fincher. 2015. "ACM Curriculum Reports: A Pedagogic Perspective." In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 81–89. Omaha Nebraska USA: ACM. https://doi.org/10.1145/2787622.2787714.

———. 2016. "The History and Purpose of Computing Curricula (1960s–2000s)." In *Communities of Computing: Computer Science and Society in the ACM*, by Thomas J.

Misa. Association for Computing Machinery and Morgan & Claypool.

Ehn, Pelle. 1988. "Work-Oriented Design of Computer Artifacts." Arbetslivscentrum.

Fiesler, Casey. 2018. "What Our Tech Ethics Crisis Says About the State of Computer Science Education." How We Get To Next. 2018. https://www.howwegettonext.com/what-our-tech-ethics-crisis-says-about-the-state-of-computer-science-education/.

Fiesler, Casey, Natalie Garrett, and Nathan Beard. 2020. "What Do We Teach When We Teach Tech Ethics?: A Syllabi Analysis." In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 289–95. Portland OR USA: ACM. https://doi.org/10.1145/3328778.3366825.

Friedman, Batya, Peter Kahn, and Alan Borning. 2002. "Value Sensitive Design: Theory and Methods." *University of Washington Technical Report* 2 (12).

Gee, James Paul. 2011. *An Introduction to Discourse Analysis: Theory and Method*. 3rd ed. New York, NY: Routledge.

globular-toast. 2021. "We Don't Need Data Scientists, We Need Data Engineers." *Y Combinator Hacker News*. https://news.ycombinator.com/item?id=25778791.

Goldweber, Michael, John Barr, Tony Clear, Renzo Davoli, Samuel Mann, Elizabeth Patitsas, and Scott Portnoff. 2012. "A Framework for Enhancing the Social Good in Computing Education: A Values Approach." In *Proceedings of the Final Reports on Innovation and Technology in Computer Science Education 2012 Working Groups*, 16–38. Haifa Israel: ACM. https://doi.org/10.1145/2426636.2426639.

Green, Ben, and Salomé Viljoen. 2020. "Algorithmic Realism: Expanding the Boundaries of Algorithmic Thought." In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 19–31. Barcelona Spain: ACM. https://doi.org/10.1145/3351095.3372840.

Greenfield, Patrick. 2018. "The Cambridge Analytica Files: The Story so Far." *The Guardian*, 2018, sec. News. https://www.theguardian.com/news/2018/mar/26/the-cambridge-analytica-files-the-story-so-far.

Halmaghi, Horatiu. 2020. "Learning Computer Science Was Hard. Unlearning Computer Science Is Harder." McGill University.

Hamraie, Aimi, and Kelly Fritsch. 2019. "Crip Technoscience Manifesto." *Catalyst: Feminism, Theory, Technoscience* 5 (1): 1–33. https://doi.org/10.28968/cftt.v5i1.29607.

Harvey, Alison, and Tamara Shepherd. 2017. "When Passion Isn't Enough: Gender, Affect and

Credibility in Digital Games Design." *International Journal of Cultural Studies* 20 (5): 492–508. https://doi.org/10.1177/1367877916636140.

Hicks, Mar. 2017. *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing*. 1 online resource (x, 342 pages) : illustrations vols. History of Computing. Cambridge, Massachusetts: The MIT Press. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2517850.

Hill, Kashmir. 2014. "'God View': Uber Allegedly Stalked Users for Party-Goers' Viewing Pleasure (Updated)." Forbes. 2014. https://www.forbes.com/sites/kashmirhill/2014/10/03/god-view-uber-allegedly-stalked-users-for-party-goers-viewing-pleasure/.

———. 2020. "The Secretive Company That Might End Privacy as We Know It." *The New York Times*, 2020, sec. Technology. https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html.

IEEE Computer Society. 2023. "About the IEEE Computer Society." IEEE Computer Society. 2023. https://www.computer.org/about/.

Impagliazzo, John, Martin Campbell-Kelly, Gordon Davies, and John A. N. Lee. 1999. "History in the Computing Curriculum." *IEEE Annals of the History of Computing* 21 (1): 4–16.

Jiang, Sijia. 2017. "Backing Big Brother: Chinese Facial Recognition Firms Appeal to Funds." *Reuters*, 2017, sec. Banks. https://www.reuters.com/article/us-china-facialrecognition-analysis-idUSKBN1DD00A.

Joyce, Kelly Ann, Kendall Darfler, Dalton George, Jason Ludwig, and Kristene Unsworth. 2018. "Engaging STEM Ethics Education." *Engaging Science, Technology, and Society* 4 (March): 1–7. https://doi.org/10.17351/ests2018.221.

Kerasidou, Xaroula (Charalampia). 2019. "Feminist STS and Ubiquitous Computing: Investigating the Nature of the 'Nature' of Ubicomp." In *DigitalSTS: A Field Guide for Science & Technology Studies*, 99–116. Princeton: Princeton University Press. https://doi.org/10.1515/9780691190600-010.

Kimura, Aya H., and Abby Kinchy. 2016. "Citizen Science: Probing the Virtues and Contexts of Participatory Research." *Engaging Science, Technology, and Society* 2 (December): 331–61. https://doi.org/10.17351/ests2016.99.

Krebsbach, Kurt D. 2015. "Computer Science: Not about Computers, Not Science." In *Proceedings of the 2015 International Conference on Frontiers in Education: Computer Science and Computer Engineering, CSREA Press*. Las Vegas, Nevada.

Lehr, Jane L. 2006. "Social Justice Pedagogies and Scientific Knowledge: Remaking Citizenship in the Non-Science Classroom." Doctoral dissertation, Virginia Tech.

Li, Tania Murray. 2007. *The Will to Improve: Governmentality, Development, and the Practice of Politics*. Duke University Press. https://doi.org/10.2307/j.ctv11smt9s.

Light, Ann. 2011. "HCI as Heterodoxy: Technologies of Identity and the Queering of Interaction with Computers." *Interacting with Computers* 23 (5): 430–38. https://doi.org/10.1016/j.intcom.2011.02.002.

Liptak, Andrew. 2018. "Peter Thiel's Data Company Palantir Will Develop a New Intelligence Platform for the US Army." The Verge. 2018. https://www.theverge.com/2018/3/10/17104878/palantir-us-army-contract-peter-thiel-data-intelligence-distributed-common-ground-system-tech.

Malazita, James W., and Korryn Resetar. 2019. "Infrastructures of Abstraction: How Computer Science Education Produces Anti-Political Subjects." *Digital Creativity* 30 (4): 300–312. https://doi.org/10.1080/14626268.2019.1682616.

Mayhew, Eric J. 2023. "'There's No One Trick': Clarifying Critical Pedagogy in the Computer Science Classroom." McGill University.

Mayhew, Eric J., and Elizabeth Patitsas. 2021. "Materiality Matters in Computing Education: A Duoethnography of Two Digital Logic Educators." In *Workshop on Computing within Limits*. https://doi.org/10.21428/bf6fb269.19c98f37.

McPherson, Tara. 2011. "U.S. Operating Systems at Mid-Century: The Intertwining of Race and UNIX." In *Race after the Internet*, edited by Lisa Nakamura and Peter Chow-White. New York: Routledge.

Mellström, Ulf. 2009. "The Intersection of Gender, Race and Cultural Boundaries, or Why Is Computer Science in Malaysia Dominated by Women?" *Social Studies of Science* 39 (6): 885–907.

Menendez-Blanco, Maria, Pernille Bjorn, and Antonella De Angeli. 2017. "Fostering Cooperative Activism through Critical Design." In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 618–29. Portland Oregon USA: ACM. https://doi.org/10.1145/2998181.2998198.

Metz, Cade, and Scott Blumenthal. 2019. "How A.I. Could Be Weaponized to Spread Disinformation." *The New York Times*, 2019, sec. Technology. https://www.nytimes.com/interactive/2019/06/07/technology/ai-text-disinformation.html.

Misa, Thomas J. 2016. *Communities of Computing: Computer Science and Society in the ACM*. New York, NY, USA: Association for Computing Machinery and Morgan & Claypool. https://doi.org/10.1145/2973856.

Moore, Jared. 2020. "Towards a More Representative Politics in the Ethics of Computer Science." In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 414–24. Barcelona Spain: ACM. https://doi.org/10.1145/3351095.3372854.

Nafus, Dawn. 2012. "'Patches Don't Have Gender': What Is Not Open in Open Source Software." *New Media & Society* 14 (4): 669–83. https://doi.org/10.1177/1461444811422887.

Nathan, Lisa P., Michelle Kaczmarek, Maggie Castor, Shannon Cheng, and Raquel Mann. 2017. "Good for Whom?: Unsettling Research Practice." In *Proceedings of the 8th International Conference on Communities and Technologies*, 290–97. Troyes France: ACM. https://doi.org/10.1145/3083671.3083685.

Núñez, Anne-Marie, Matthew J. Mayhew, Musbah Shaheen, and Laura S. Dahl. 2021. "Let's Teach Computer Science Majors to Be Good Citizens. The Whole World Depends on It." EdSurge. 2021. https://www.edsurge.com/news/2021-03-15-let-s-teach-computer-science-majors-to-be-good-citizens-the-whole-world-depends-on-it.

O'Neill, Patrick Howell. 2019. "How WeChat Censors Private Conversations, Automatically in Real Time." MIT Technology Review. 2019. https://www.technologyreview.com/2019/07/15/134178/how-wechat-censors-private-conversations-automatically-in-real-time/.

Oswick, Cliff. 2012. "Discourse Analysis and Discursive Research." In *Qualitative Organizational Research: Core Methods and Current Challenges*, by Gillian Symon and Catherine Cassell, 473–91. 55 City Road: SAGE Publications, Inc. https://doi.org/10.4135/9781526435620.

Parvin, Nassim, and Anne Pollock. 2020. "Unintended by Design: On the Political Uses of 'Unintended Consequences.'" *Engaging Science, Technology, and Society* 6 (August): 320–27. https://doi.org/10.17351/ests2020.497.

Patitsas, Elizabeth. 2019. *Explaining Gendered Participation in Computer Science Education*. University of Toronto (Canada).

Quinn, Ben, and James Ball. 2014. "US Military Studied How to Influence Twitter Users in Darpa-Funded Research." *The Guardian*, 2014, sec. World news. https://www.theguardian.com/world/2014/jul/08/darpa-social-networks-research-twitter-influence-studies.

Quinn, Michael J. 2006. "On Teaching Computer Ethics within a Computer Science Department." *Science and Engineering Ethics* 12 (2): 335–43. https://doi.org/10.1007/s11948-006-0032-9.

Raj, Rajendra K., and Amruth N. Kumar. 2022. "Toward Computer Science Curricular Guidelines 2023 (CS2023)." *ACM Inroads* 13 (4): 22–25. https://doi.org/10.1145/3571092.

Ramachandran, Aishwarya, Jerry Achar, Georgia Green, Brynley Hanson-Wright, Sophie Leiter, and Gunilla Öberg. 2022. "Changing Debates and Shifting Landscapes in Science Studies: Exploring How Graduate Students with Varied Backgrounds Think About the Role of Value-Judgments in Science." *Engaging Science, Technology, and Society* 8 (2). https://doi.org/10.17351/ests2022.1179.

Rosner, Daniela. 2018. *Critical Fabulations: Reworking the Methods and Margins of Design*. Design Thinking, Design Theory. Cambridge, Massachusetts: The MIT Press.

Rupf, John A. 2004. "Teaching the History of Computing (Painlessly)." *Journal of Computing Sciences in Colleges* 20 (2): 212–18.

Sahami, Mehran, Steve Roach, Ernesto Cuadros-Vargas, and Richard LeBlanc. 2013. "ACM/IEEE-CS Computer Science Curriculum 2013: Reviewing the Ironman Report." In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education - SIGCSE '13*, 13. Denver, Colorado, USA: ACM Press. https://doi.org/10.1145/2445196.2445206.

Samaka, Mohammed. 2002. "Changing a Computer Science Curriculum in Light of Computing Curricula 2001." *ACM SIGCSE Bulletin* 34 (4): 32–35. https://doi.org/10.1145/820127.820160.

Sengers, Phoebe, Kirsten Boehner, Shay David, and Joseph "Jofish" Kaye. 2005. "Reflective Design." In *Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility*, 49–58. Aarhus Denmark: ACM. https://doi.org/10.1145/1094562.1094569.

Shilton, Katie. 2013. "Values Levers: Building Ethics into Design." *Science, Technology, & Human Values* 38 (3): 374–97. https://doi.org/10.1177/0162243912436985.

Skirpan, Michael, Nathan Beard, Srinjita Bhaduri, Casey Fiesler, and Tom Yeh. 2018. "Ethics Education in Context: A Case Study of Novel Ethics Activities for the CS Classroom." In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 940–45. SIGCSE '18. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3159450.3159573.

Sturman, Susan Michele. 2009. "'Women in Computing' as Problematic: Gender, Ethics, and Identity in University Computer Science Education." Doctoral dissertation, University of Toronto.

Suchman, Lucy. 1993. "Working Relations of Technology Production and Use." *Computer Supported Cooperative Work* 2 (1–2): 21–39. https://doi.org/10.1007/BF00749282.

Tedre, Matti, and Erkki Sutinen. 2008. “Three Traditions of Computing: What Educators Should Know.” *Computer Science Education* 18 (3): 153–70. https://doi.org/10.1080/08993400802332332.

The New York Times, dir. 2019. *A.I. Is Making It Easier to Kill (You). Here’s How. | NYT*. https://www.youtube.com/watch?v=GFD_Cgr2zho.

Tijdens, Kea Gartje. 1997. *Gender Segregation in IT Occupations*. Citeseer.

Vincent, James. 2021. “They’re Putting Guns on Robot Dogs Now.” The Verge. 2021. https://www.theverge.com/2021/10/14/22726111/robot-dogs-with-guns-sword-international-ghost-robotics.

Wagner, Ben. 2019. “Ethics As An Escape From Regulation. From ‘Ethics-Washing’ To Ethics-Shopping?” In *BEING PROFILED*, edited by Emre Bayamlioglu, Irina Baraliuc, Liisa Albertha Wilhelmina Janssens, and Mireille Hildebrandt, 84–89. Amsterdam University Press. https://doi.org/10.1515/9789048550180-016.

Williams, Rua M., Simone Smarr, Diandra Prioleau, and Juan E. Gilbert. 2022. “Oh No, Not Another Trolley! On the Need for a Co-Liberative Consciousness in CS Pedagogy.” *IEEE Transactions on Technology and Society* 3 (1): 67–74. https://doi.org/10.1109/TTS.2021.3084913.

Witz, Anne. 1992. *Professions and Patriarchy*. International Library of Sociology. London ; New York: Routledge.

Wong, Alice. 2015. “Assistive Technology By People with Disabilities, Part II: Designing Better Makeathons.” *Model View Culture*, 2015. https://modelviewculture.com/pieces/assistive-technology-by-people-with-disabilities-part-ii-designing-better-makeathons.

Ymous, Anon, Katta Spiel, Os Keyes, Rua M. Williams, Judith Good, Eva Hornecker, and Cynthia L. Bennett. 2020. “‘I Am Just Terrified of My Future’ Epistemic Violence in Disability Related Technology Research.” In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–16. Honolulu HI USA: ACM. https://doi.org/10.1145/3334480.3381828.

Young, Alison, Logan Muller, Samuel Mann, and Lesley Smith. 2009. “Computing Education for Sustainability: Madrid and Beyond.” In , edited by Samuel Mann and Michael Verhaart. Napier, New Zealand.

Ziv, Amitai. 2019. “This Israeli Face-Recognition Startup Is Secretly Tracking Palestinians.” *Haaretz*, 2019. https://www.haaretz.com/israel-news/business/2019-07-15/ty-article/.premium/this-israeli-face-recognition-startup-is-secretly-tracking-palestinians/0000017f-f47b-ddde-abff-fc7fe

25c0000.