Teaching Professional Morality & Ethics to CS Students through Cognitive Apprenticeship & Case Studies: Experiences in CS-HU 130 'Foundational Values'

Journal:	*Computing in Science and Engineering*
Manuscript ID	Draft
Manuscript Type:	SI: Best of RESPECT 2019
Date Submitted by the Author:	n/a
Complete List of Authors:	Winiecki, Donald; Boise State University, Organizational Performance & Workplace Learning; Salzman, Noah; Boise State University, Electrical & Computer Engineering, IDoTeach
Keywords:	K.7.m.b Ethics < K.7.m Miscellaneous < K.7 The Computing Profession < K Computing Milieux, Professional Morality, Inclusion, Diversity, Social Justice, Computer Science, Higher Education

Teaching Professional Morality & Ethics to Undergraduate Computer Science Students through Cognitive Apprenticeships & Case Studies: Experiences in CS-HU 130 'Foundational Values'

Don Winiecki and Noah Salzman

Abstract—This article describes and details experience in guiding undergraduate computer science students to identify and address issues related to inclusion, diversity and social justice as they occur in computer science education, computer science professions, and in the products of computer science work. Specific details reported here arise from experience teaching a one-credit undergraduate course at Boise State University (CS-HU 130, Foundational Values).

Index Terms— Diversity, Ethics, Inclusion, Professional Morality, Social Justice

1 Introduction

EVEN with only occasional attention in the popular press and academic research, one encounters a steady stream of stories describing how members of underrepresented groups in engineering and computer science have encountered personal and systemic instances of bias. These experiences are sometimes subtle, sometimes overt, and come from both individuals and from the very systems in which people learn, work and interact with technologies [1]–[8].

This set of evidence makes one thing clear — attitudes that underpin actions that result in preference for some people and which discount or exclude others (i.e., bias) is a fact in the social world. Since computer science is part of the social world, its occurrence in CS is itself unsurprising (though at one time there was belief that CS would be free of this phenomenon [9].) It is also apparent that accounts of the effect of such bias are often dismissed and rationalized by the meritocratic fallacy in computer science and other fields. The 'meritocratic fallacy' is a set of beliefs that lead to the claim that raw technical and intellectual skill (i.e., 'merit') is all that is required to achieve success and status in computer science and related fields, and the resulting 'intuitive' conclusion that underrepresentation of particular groups of individuals in computer science and related fields reflects the lack of technical and intellectual skill among those individuals. From this, those who subscribe to the meritocratic fallacy allow themselves to conclude that 'underrepresented groups' are simply not qualified or even capable of meeting the demands of the field.

With that in mind, the Computer Science department at Boise State University and its industry partners have committed to addressing systemic bias in the field, across the undergraduate curriculum and into student internships in local industries. This effort has been substantively supported through an NSF RED grant (Revolutionizing Engineering & Computer Science Departments) that supports what we call the 'Computer Science Professional's Hatchery' (aka CSP-Hatchery).

This commitment to transformation is manifested through an innovative curricular framework of one-credit courses that enables the BSU computer science department to rapidly develop and roll-out special-topics courses that address specific needs identified by local industry, and include curricular and project-level requirements addressing documented sources of bias as occur within the scope of that topic. Every undergraduate student is now required to take at least five of these one-credit courses as part of the student's degree, and in so doing become competent in technical and professional skills needed by their eventual employers, and sensitive and able to respond to socio-technical factors that could lead to biases. Targeted and systemic incentives are being used to facilitate faculty participation in this project.

The fact these one-credit courses are designed around technical topics of critical strategic importance for industry means that they also provide a powerful opportunity for introducing aspects of ethics and professional morality as they relate to those topics. In other words, by coupling technical and ethics-related issues in the framework of a nominally 'technical' course we are able to provide demonstration that ethics are neither separate from nor irrelevant to technical aspects of computer science. This itself is an important part of the environment provided by the CSP-

- *Don Winiecki is with Organizational Performance and Workplace Learning, Boise State University, Boise, ID 83725. E-mail: dwiniecki@boisestate.edu.*
- *Noah Salzman is with the Electrical and Computer Engineering Department, Boise State University, Boise, ID 83725. E-mail: noahsalzman@boisestate.edu.*

Hatchery in support of our interests in promoting inclusion, diversity, and social justice. Mixing technical and ethics topics within courses helps us avoid the usual separation of societal concerns from technical concerns that occurs when a curriculum just includes a dedicated course on ethics and related topics that may be offered by another department on campus (e.g., philosophy, sociology, etc.). In such a curriculum, students are allowed to see ethical concerns as separate from what they believe to be the technical core of their studies *because* the curriculum itself has separated them.

The first of these one-credit courses (and the first Computer Science course required of all students) is CS-HU 130 'Foundational Values'. CS-HU 130 exposes students to case studies that document breaches in inclusion, diversity and social justice in computer science education and professional practice, and in the products of computer science work [2], [5], [10]–[12]. In addition to a nominal exposure to these moral and ethical breaches in the context of real world CS activity, students are provided with a problem-solving rubric and cognitive apprenticeship-type support in using that rubric to systematically analyze the problem(s) and offer systemic solutions to them [13], [14]. More detail on this process is provided below.

In the first case introduced to students, the experiences of Susan Fowler in her role as a 'Site Reliability Engineer' at Uber [5] highlights how one can assign responsibility for harassing behavior to individuals (i.e., those doing the harassing), but also to systemic failures in responsibility and accountability in the surrounding organization. In analysis of the case, students find that acting to punish or remove the individual harasser would not remedy the systemic problems that led the HR department at Uber to attempt covering for and protecting him because he was a 'high contributor' to the organization's cash flow, and upper management's pattern of diminishing the contributions of women in the workplace when awarding incentives. Addressing the issues that affected Fowler and others at Uber would instead require systemic responses that influence personnel, and several layers of management at the same time. While the type of issues at play in this case are not consistent with a narrowly-defined commonsense understanding of an *engineering problem*, engineers acting as responsible agents in the social system of an organization can use the tools provided to *engineer* sustainable solutions to such problems and in turn facilitate higher performance by engineers doing engineering work. In CS-HU 130 students do just that.

Similarly, the failure of due diligence and responsibility over time and across societal institutions and subsets of computer science (e.g., machine learning) allowed for the longstanding existence of racial bias in society (i.e., the way racism figured into assumptions and ideology of validity in 'big data' and social media) to be reified and literally built into computing applications in the use of the COMPAS software program intended to assign risk of recidivism among individuals accused of crimes in the State of Florida [10].

In analyzing this case students identify that bias was literally built into the data that was used, and process and products of machine learning, as reflected in COMPAS. They also identify that this is enabled by an incorrect definition of the problem to be addressed with machine learning, a priori biased assumptions, and unreliable and biased data (all enabled by insensitivity to bias and lack of awareness of basic social science facts and research).

When case data are made available and students are provided with detailed support in identifying and sorting through the historical creation of bias in criminal justice, to bias in the data that was created for use by COMPAS, to bias in the way these data are processed, our entry-level computer science students are able to produce an *engineering* analysis of the problem and to prototype a path to sustainable solutions that go beyond a narrow definition of computer science. Perhaps more importantly, through this process students identify that they *as aspiring computer scientists* can have a central part of addressing associated societal problems. In other words, our students begin to see the power of science taken broadly as a thoroughgoing focus on validity and reliability of data, data processing, and data interpretation in order to advance the creation and maintenance of social goods.

In accomplishing their analysis of these and other cases, students follow a rubric based on Rawls' theory of justice [13], and models of organizational performance improvement [14], to propose systemic and sustainable changes in organizations that will help to reduce and remove influences that result in these breaches of ethical conduct and responsible computer science in educational settings, professional settings, and the effects of computer science work on societal factors. More details are provided on this theory and rubric below.

In addition, other courses in the curriculum are being developed and updated to include modules which connect what has historically been considered to be 'purely technical' content with social science content supporting inclusion, diversity and justice. This aims to gradually build a curriculum that operates against the usual situation in which an explicit effort to separate societal factors from technical factors actually facilitates an environment where individuals who claim to be unbiased are able to introduce their implicit assumptions about social matters but claim not to be doing so. In other words, by allowing a claimed separation of technical and societal, we allow individuals claiming to be 'purely technical' to smuggle their implicit assumptions into the environment as a 'natural part' of the 'purely technical' or objective features of the environment.

In combination, these innovations aim to produce a learning ecosystem in which technical and societal factors are always included in computer science learning and working communities. Through this process we aim to help produce a 'new generation' of computer scientists who understand the value in collaborating with social science experts and in this process becoming knowledgeable about the micro- and macro-societal effects of their actions in computing. In turn, we aim to provide to current and emerging industries entry-level professionals who are fluent in necessary technical skills and equipped with the moral sensitivity to identify and mitigate known and emerging risks and biases, all for the betterment of their organizations, society and its members.

2 Discussion

The core of CS-HU 130 'Foundational Values' modules that focus on inclusion, diversity and social justice into courses throughout the computer science curriculum is a framework for systematically analyzing case study material documenting and detailing violations of inclusion, diversity and social justice in computer science learning and working environments and the effects of computing products when deployed in society, and then using a rubric based on Rawls' theory of justice [13] and principles of organizational performance improvement [14] to propose interventions that focus on systemic (i.e.., individual *and* organizational) forces that are root causes of identified problems.

This rubric is presented as a worksheet used by students to systematically analyze the problems and identify their sub-components, and then synthesize a system of rules, incentives, and disincentives with added information to address the identified problems in a way that may produce a more inclusive, diverse and just organizational whole. Note that we do not expect students to demonstrably solve such problems in CS-HU 130, but rather to develop analytical skills and habits of mind that can be put to use in improving their learning and working environments as they grow.

The rubric contains four components: (a) identify empirical aspects of the situation in which the problem occurs, (b) identify rights and duties that should exist and which apply to each status group involved in the problem situation, (c) identify/propose objective rules, and material incentives and disincentives that can be put in place systemically in order to pull or push stakeholders to fulfill their rights and duties, and (d) provide a brief narrative or bullet-list type description of how incentives and disincentives will help to support and protect the rights and duties of stakeholders across the system, and thus systemically solve the problem.

Items '(a)' and '(b)' come directly from Rawls' 'theory of justice' [13], and are consistent with what Vallor [16] references as 'right thinking' and 'right behavior' in her analysis of what she calls techno-moral virtue, and how it can be accomplished. Rawls' theory of justice [13] is an analytic framework that attempts to synthesize and synchronize principal variants of ethical theory (e.g., consequentialism, deontology/rights, social contract/common good, justice, and virtue) in a way that would make them all mutually reinforcing and practicable in a diverse and socially liberal capitalist society. While there are principled disagreements with Rawls' formulation (some saying he 'goes too far' others that he 'doesn't go far enough' in regulating particular angles of each embedded framework, and some claiming he is simply misinterpreting ethical theory in order to produce his formulation) it remains a very influential framework that converges wide variations in ethical theory in a way that fits into current Western variants of liberal capitalist society without also requiring highly detailed and nuanced knowledge of philosophical, social, and economic theory. This last point is a primary warrant for using it as a basis of our developments in CS-HU 130, where individuals are more than likely to be incoming first-year students in computer science without such detailed knowledge.

Rawls himself uses the social contract perspective as a sort of wrapper for other ethical frameworks with the idea that any group of individuals working toward similar ends will require a more or less agreed-upon set of expectations or tacit rules for what is allowable and what is not allowable. If these individuals can be seen to be somehow cooperating (if not collaborating) then it is also relevant to assume that each member of the group will have more or less definable rights and duties that must be fulfilled in order to realize individual and group goals. Fundamental assumptions of capitalism (which leans into the consequentialist ethical perspective) hold that in order for individuals to do anything with reliability, there must be some material rewards or threats to rewards that pulls those individuals into a regular pattern of action. This focus on what is allowable or not allowable, rights and duties, and promise of rewards for fulfilling expectations (or disincentives for failing to fulfill those expectations) is a foundation for the CS-HU 130 curriculum.

Rawls also acknowledges that modern liberal capitalist societies are built upon an expectation that innovation, and thus ongoing changes in the details of everyday action, will be a constant. With this in mind there is no realistic possibility of providing a highly-detailed and rigid set of rules, rights and duties, or incentives and disincentives. However, there is also a recognition that actors will orient to the effects of their actions and their innovations and seek to refine them so that benefits are available to all members of the system. In other words, while Rawls admits that individuals must have the ability to innovate in processes to realize shared goals as innovations changes the means and details of ends for those goals, those actors will not simply produce work or innovations without also refining them for the benefit of all. This requires what Rawls called the 'wide reflective equilibrium' — a habit of mind that requires an individual to consider (a) the varying values and interests of different members in the system along with (b) innovations in processes even while (c) keeping top-most goals constant, and then to refine innovations so they restore and/or maintain equilibrium even within the changes of a constantly evolving modern society.

However, the kernel of the focus in CS-HU 130 is built upon what Rawls calls the 'veil of ignorance'. For Rawls, when one is under the veil of ignorance, that individual is unaware of one's status in a system (gendered, racial or ethnic, socio-economic, aged, etc.) even while one is aware of prevailing norms, assumptions, and ways — including embedded biases — in the system.

With the veil of ignorance, Rawls believes that an individual is less likely to accept or ignore implicit assumptions that benefit only certain classes or groups of individuals in a society and that (because each individual is unaware of one's own membership in a particular class or group) could put oneself at a disadvantage in a competitive liberal system. Thus, with the veil of ignorance one is more likely to consider adopting a position relative to rules, rights, duties, and incentives and disincentives that would protect oneself regardless of one's own status in the system. Because adoption of such positions may or may not actually benefit one's own situation it is parallel to (but not the same as) a virtue ethics principle in which one does things for their overall 'rightness' rather than for explicit personal benefit, and a utilitarian ethics principle underlying the protection of an individual's stake in the system.

Item '(c)' in the rubric (identify objective rules, and material incentives and disincentives that can be put in place systemically in order to pull or push stakeholders to fulfill their rights and duties) reflects elements of consequentialist ethics content incorporated into Rawls theory, but is primarily derived from proven techniques for effecting systemic behavior and process changes in organizational performance [14]. These techniques are based on established principles of behavioral psychology and rational-choice theory in sociology [17] in which widespread (although not universal) and sustainable systemic change can be accomplished by synchronizing two things:

- Knowledge and understanding of rules and their consequences.
- Knowledge and understanding of the benefits/costs of material incentives or disincentives that come from fulfillment/violation of the rules.

Importantly, these two elements occur mostly in the *environment* of an organization and do not require a high degree of personalization or customization in order to appeal to specific individuals. This is consistent with the principle in behavioral psychology and rational-choice theory that systemic change is prompted by stimuli (i.e., information, rules/duties, policies, etc.) in the performance environment, that are reinforced and made sustainable by consistent provision of incentives for fulfillment of the rules/duties, or disincentives that come when the rules/duties are not fulfilled.

As a result of being part of the performance *environment* and not as highly flexible variable that require constant customization for each individual in an organization, the system is more easily implemented and sustained within typical organizational structures. This is consistent with Rawls' overall interest in a system of ethics that protects common good/social contract, justice, rights, and utility, and gestures to principles of the virtue ethics perspective so that everyone can realize some aspects of their own personal ethical orientation.

Identifying problems associated with inclusion, diversity, and social justice, identifying rights and duties, and proposing solutions is not difficult following a careful reading of the case with issues of ethics and justice in mind. However, it is also not something that appears to be commonplace in the current ecology of technology work [1]-[12] (or, perhaps any kind of work!). As a result, it is critical to provide students with concrete experiences and coaching in how to accomplish it in the course of activities in CS-HU 130 'Foundational Values'. It is perhaps even more important to effect changes across the entire environment of computer science education and professional work — an issue that we will return to in the conclusion below.

In CS-HU 130 'Foundational Values,' this is accomplished through a variant of the 'cognitive apprenticeship' process [18]. In this process the instructor leads a structured analysis and discussion of the case using the framework students are expected to use as modeled by Schön [19].

The following section of this paper contains a description of the lessons that constitute the main elements of CS-HU 130. This description follows a publicly-available lesson plan and demonstration/prototype we have developed that fulfills the theory and ideas included above. The description below is built around an example case and a worked-out example of the rubric described above <http://tinyurl.com/yxqc95uf>). Please refer to that document as you proceed with this article.

3 Details of the Lesson

The demonstration/prototype available at the URL included above contains the following elements:

- Page 1 contains the instructional objectives/outcomes for the lesson. Students do not have access to this information.
- Pages 1-6 includes a narrative description of the lesson and included activities. This provides an orientation to some of the nuances the instructor will include during performance of the lesson. Students do not have access to this information.
- Page 7 (Appendix A) is a blank rubric students will complete during the lesson. Students have access to this as a Google Doc, and should have it open on their computers so they can add details through the lesson. The instructor also has this rubric displayed on the classroom projection system.
- Pages 8-10 contains the case example students should have read before this lesson. This is available online for students and they should have it open on their computers so they can reference it through the lesson. Footnotes embedded in the case provide references to relevant related concepts from social science and ethics.
- Pages 11-13 includes an example worked-out rubric for the instructor's reference.

Referring to the lesson plan at the URL indicated above, part 1 of the lesson (page 2) sets the stage for analysis by helping students identify primary characters and actions reflected in the case. Part 2 of the lesson (page 3) includes a recursive analysis of the case in which 'problems' are identified. The case is written so that 'problems' are identifiable as impediments to learning in a computer science class, which reflect stereotypical bias against a female computer science student (i.e., loss of inclusion and social justice in the learning environment, including failure to use diversity as a positive contributor to learning). Following identification of problems, in part 3 of the lesson (page 3-4) students are guided to identify rights and duties that should be assigned and accepted by individuals in the story in order to fulfill their learning goals. Rights and duties are symmetrical in the sense that if one can claim certain rights for oneself, then there are also duties to protect and ensure others have the same rights. In part 4 of the lesson (page 4-5) the instructor guides students to identify incentives and disincentives that will provide material reinforcement for accomplishing rights and duties. Part 5 constitutes creation of a brief description of how and why all of the preceding parts will work to contribute to solutions of the identified problems.

While the lesson plan reflects a serial process, it is not uncommon for students to start thinking ahead and actually make observations and suggestions that could be included in subsequent parts of the rubric. The instructor actively enters details into each section of the rubric as relevant to accomplishment of lesson objectives.

Turning to the actual case example (page 8-10), we read a case where a new female computer science student (Jamie) is confronted with gender-bias in a course team project. This is the case even though she demonstrates knowledge and capacity to perform equal to or better than teammates. The bias is explicitly directed from one individual (Andy, the nominal 'team leader'), but lack of public response and push-back from other team members provides a tacit ratification for this bias across the team. This makes is relevant to point out that not addressing obvious bias will actually make someone a passive participant in propagating the bias (i.e., even if one claims to be 'not biased,' if one also does not act to defuse bias, one becomes an agent in ensuring the bias persists).

Jamie responds in a manner that may suggest stereotype threat in response to the micro- and not-so-micro aggressions and she shys from seeking more participation in the team project. After one of the other team members (Roy) reaches out to Jamie and asks to work together with her without the knowledge of the others he discovers that Jamie has much more knowledge and skill in computer science that contributes to the overall goal of learning, and which explicitly help Roy. This points up the idea that acting to commit stereotyped bias not only hurts the victim in a personal way, but also hurts everyone in the situation in systemic ways.

When Roy and Jamie return to the group to report their gains, Andy confronts them and amplifies and diversifies his gender-bias, all the while he and the remaining team member (Vince) have not made accountable progress in the project. This reinforces the backsplash effect of biased behavior.

In the case we see that Andy appears not to be interested in the authentic shared goal — learning, and subsequent improvements in computer science skill and knowledge. In his biased reactions he is not only hurting Jamie but actually working against the explicit goal of the course and the team project and in turn producing a situation where a very capable student (Jamie) is no longer able to help he and others accomplish the actual goals of the class.

We also see a situation where there does not appear to be any explicit expectations in the class for conduct conducive of inclusion of members underrepresented groups. In the context of this lack of explicit expectations, prevailing social biases and coping strategies held by any of the actors are allowed to enter the situation, along with other micro-cultural norms that typify the historical lack of inclusiveness in computer science and other technical disciplines [3], [4], [8], [9].

Faced with this case, students are asked to put on a 'veil of ignorance' to identify problems that could very well affect themselves in similar situations (if they turn out not to be a member of one of the 'normal' groups), then identify rights and duties that members of the classroom should have, and rules, incentives and disincentives that could be put in place to help prevent the problems from occurring in the future. This requires the use of a 'wide reflective equilibrium' [13] in order to satisfy the (usually diverse) values and assumptions of actual students in CS-HU 130 without explicitly dis-including anyone, but while still protecting what should be the principal rights and duties of each member toward realizing the overall goals of the system (in this case, learning and skill development).

In practice this often requires creativity and caution on the part of the instructor to ensure that all students can contribute to the discussion and completion of the rubric. Even students who hold assumptions that work against the goals of inclusion, diversity and justice have to be heard. This is the case because their views and assumptions are part of the authentic environment of computer science and everyone has to acquire knowledge, skills, and strategies that account for even contrary ideas.

For example, it is not uncommon for someone in the class to note that Andy is himself disadvantaged by Roy's and Jamie's private work on the project. This observation is usually stated something like this: "If only Jamie and Roy had told Andy that they were going to work on some things independently, and then bring their discoveries back to Andy and Vince, then Andy may have been more amenable to their activity."

Given this, the instructor can help students identify a rule/duty specifying that independent experiments on the project (sometimes called the 'divide and conquer' approach) should be made public so that every stakeholder is aware of more contingencies in the case. We have never heard any student disagree with this proposed rule/duty because it does in fact provide potential benefits to everyone on the team.

In this way, even students with nominally contrary views and who defend actions by apparent perpetrators of problems are included in the discussion. When these individuals learn that their statements are not simply 'shut down' they often become more willing participants in the class.

In identifying of incentives and disincentives it is common for students to focus on disincentives that would act to punish perpetrators of identified 'problems.' This is the perfect place to introduce students to behavioral psychology research showing that provision of incentives for fulfilling rights/duties, and withholding those incentives for non-fulfillment, is actually more effective than overt punishment.

In creation of the brief description (part 4 of the rubric) students construct what in business circles is commonly referred to as 'the pitch' or 'elevator speech.' While for our purposes this provides evidence that a student understands the system of problem solving and its value, we also point out to students that in professional practice the elevator speech is seen as a means for an individual to provide succinct and convincing evidence that demonstrates to successively higher levels of management that a problem and its factors and variables are understood. Not only that, but also that through this understanding one has induced reputable means for addressing the problem.

This also provides an opportunity to tell students that this also demonstrates how individuals in an organization can effectively 'manage up' by proactively providing solutions to one's team leader or manager. 'Managing up' both takes problems off the backs of management, and helps to address problems before they threaten wellbeing of others or the organization (as happened in the case of Susan Fowler at Uber when the problem was apparently not widely identified, understood, or addressed). As such, it provides a means through which a relatively lower-level employee can communicate to management that one is ready and able to take on some of the inevitable issues that face management regardless the industry. Because it is common for some class members to also have internships or full-time jobs in software development, it is also common for those students to reinforce this statement and the value of being a problem-solver who 'manages up.' This provides a direct connection to real-world professional practice for students in what is otherwise an entry-level course in their degree program.

Students accomplish this activity four times in CS-HU 130. In the first instance, the instructor leads students through the entire framework in one 75-minute class session, using Socratic questioning that points students back to guided (re)reading of the case, and introducing new material related to organizational performance improvement [14] in the process of coaching students in the perspectives and process to be used. On the second, third, and fourth instances, the instructor begins this process but requires students to complete successively more of the framework in teams of four to five students. As in any laboratory class, the instructor consults with each student team, asking questions about their analyses, perspectives and processes, and guiding them to a coherent system of interventions involving facts from the case, social science information, ethical frameworks and rules derived from them, and incentives and disincentives. The instructor intentionally questions groups about unstated assumptions behind their analysis and their interventions, in order to bring into explicit focus the basis of their lines of reasoning.

With the description above, and more detail provided in the lesson plan at the URL indicated above, it is possible to see that this reflects the general process of 'cognitive apprenticeship [18]:

- <u>Modeling</u>: Demonstrate the desired activity and process
- <u>Coaching</u>: Provide dynamic feedback/evaluation & feedforward/coaching/prompting to focus students on important elements as they practice
- <u>Scaffolding</u>: Start simple and build complexity as students gain relevant knowledge, comprehension, and fluency
- <u>Articulation</u>: Describe what you're doing when demonstrating, coaching, and scaffolding so students can 'see' your intellectual process and translate it into their own activity
- <u>Reflection</u>: Think back through your process as you go, in order to abstract it into functional units
- <u>Exploration</u>: Consider embedded cases, side cases, and thought experiments to exercise the process

More, and more elaborate, examples of how the cognitive apprenticeship process can be incorporated into instruction in technical problem domains are provided by Schön [19]. Similarly, Vallor provides substantive intellectual grounding and examples for teaching what she calls 'techno-moral' ethics in [16].

As suggested above, while the process of cognitive apprenticeship is listed as a taxonomy, it is actually implemented in a rhizomatic fashion in which each element can lead-into and arise-from other elements [19], [20].

Across the four semesters this course has been offered, instructional process has been dynamically adjusted. One of these adjustments has produced a notable pattern of improvements in knowledge, comprehension, application, analysis and culminating synthesis of course content [20]. This improvement occurs parallel with a straightforward application of the instructional process of 'fading' – the gradual decrease of support offered to students as they demonstrate greater familiarity and fluency with the subject matter and its application in successively more complex problem sets. Through each class, student teams demonstrate development of ability to identify (know), describe (understand), and apply the resulting knowledge and understanding of bias and loss of social justice as *problems associated with how social structures have been informally developed and stabilized* – thus as engineering problems. Once the problems have been so defined and understood, they are then able to develop approaches to solving the problems from the standpoint of human performance engineering [14].

Notably, reconfiguring of issues from the standpoint of societal problems of subjective bias to engineering problems (albeit still from Rawls' perspective) has also afforded a switch that appears to allow students skeptical of "liberal" (scare quotes are intentional) conceptions of 'social justice' to participate and contribute in ways that suggest an important turn in both personal and professional responsibility.

Importantly, the topics addressed in case studies in CS-HU 130 do not all conform to the sort of interpersonal biases described in the case above. We also include case studies of commercial software that reflects algorithmic bias reifying long standing biases in criminal justice and racial and gender bias in facial recognition technologies. Cases highlighting the problems associated with panoptic observation/surveillance and use of 'big data' in ubiquitous machine learning required for smart city technologies are under development.

The goal in this range of case examples is to provide students with tools and experiences that permit them to see that computer science is not immune from the sort of social problems that fill our news feeds daily, and in fact that those problems occur and are in some cases reinforced by the very technologies they aspire to develop.

However, the goal is not to take a determinist perspective asserting that technology is bad. Instead, it is our goal to help students develop perspectives and skills in the use of tools and those perspectives that allow them to design and implement processes and technologies that fight against social problems.

Across all of the above, in CS-HU 130 it is not uncommon for students to rapidly 'get the pattern' and to produce very good outputs for individuals in their first year of undergraduate computer science study. This provides evidence of 'near transfer' that allows us to say that our approach is not only practicable in class situations but also affects students in their day-to-day lived experiences. Anecdotally, some of these affected students are those who initially expressed resistance to a class that explicitly incorporates an orientation to *inclusion, diversity, and social justice* in the catalog course description. In discussions with these individuals, they confirm that the approach to cognitive apprenticeship – and explicitly our focus on the (Rawlsian) notion that justice is itself a social good that both includes them and does not require them to behave like (whatever is their image of) the "social justice warrior" (scare quotes are intentional) – helps to break down their initial resistance to the content.

Focusing on the the outputs of this process in this way has proven to be a useful 'selling point' for the class to some students. In response to the requirement of describing how incentives and disincentives can help to support the rights and duties of stakeholders in the case analyses, one student wrote in the end-of-course feedback survey "I don't always agree that [these cases identify] problems or even problems for computer science, but if they are problems that management wants to solve, I now know how to do that and how to improve my promotion potential in the process." This demonstrates how our focus on issues of inclusion, diversity and justice as organizational problems (in addition to foundational social problems) makes the course, its content, and its takeaways valuable for more individuals than just those who have a reason to consider problems of bias as fundamental social problems. This is consistent with the approach sometimes called 'social justice lite' [15] in which the realization of social justice is a *side effect* of more corporate and consequentialist goals rather than a primary focus on realizing social justice.

Thus, we know our curriculum and curricular methods work within the bounds of the course itself. *However, this is still (just) a one-credit course taken early in their student careers*. In order to assess the extent to which the outcomes of this course transfer to their lives after CS-HU 130, we are conducting an ongoing set of interviews with students who have completed CS-HU 130 through their career at Boise State University. In these interviews we ask about their experiences in class and university life and about their perceptions and actions with respect to topics introduced in CS-HU 130.

In the four semesters that CS-HU 130 has been offered, over 500 students have taken the course. We have completed 96 interviews with 52 students (including up to four interviews per student. The majority of our interviewees identify as white (40), and include Asian (6), Latinx (1) and individuals who identify as multiracial (5).

In our interviews we find that students mostly retain an ability to identify social problems in newspaper type accounts that explicitly focus on loss of social justice in either technology work, or in the design, deployment and use of technology products. However, we find that transfer of this sensitivity to 'the wild' of their everyday lives in class and university life to be less common than we would hope.

Specifically, while students demonstrate the ability to analyze and propose systemic solutions to 'prepared' case examples similar to those used in class, it is not as common for students to be able to identify episodes of bias in their daily lives. We do not see this as a problem with our curriculum in CS-HU 130, but rather as a feature of very durable habits of mind that they have developed and which have been reinforced by a social world and disciplines (e.g., computer science) that has historically been insensitive to bias affecting underrepresented groups [3], [4], [8], [9].

The obvious next step is to embed course assignments similar to those described above into other one-credit courses offered in our 'Hatchery'. This has been partially realized, though evaluation of its effects has not been completed.

In addition to this we are initiating efforts to filter similar and new activities into the more conventional three-credit courses that make up the bulk of our computer science curriculum so that students are exposed to a broader diversity of cases and more practice through the overall undergraduate curriculum.

Parallel with our developments we are encouraged to see that many incoming faculty members have begun to incorporate social-justice-oriented innovations into courses. Regular conversations with faculty members allows us to learn more about their approaches and share them with students in CS-HU 130 with the message that this is an authentic concern of computer science. Some of our CS-HU 130 students have followed up with those faculty members and even been added to their research teams with the stated aim of becoming part of that 'new generation' of computer science.

3 Conclusion

Above we have described curricular innovations that are aimed at systemically addressing both conventional technical content and societal concerns about loss of inclusion, diversity, and social justice in computer science education, professional practice, and in the effects of CS on society. We have described our curriculum and provided a link and detailed description of detailed lesson plan for completing one element of the curriculum in CS-HU 130 'Foundational Values.' CS-HU 130 is the first course undergraduate computer science students are required to take toward their degree at Boise State University. As such it is intended to set the scene for students and in particular to set expectations and provide tools and practice at identifying and proposing solutions to social problems in computer science education and professions.

In the context of CS-HU 130 we have provided evidence to indicate that our students can successfully accomplish these goals. However, additional research underway also shows that transfer outside of this class has not been widespread. We offer suggestions for addressing this and will report on our own experiences with these efforts in the future.

Acknowledgment

The authors wish to thank Erika Abbot and Michelle Fretwell for their contributions to this work. This project was supported by the National Science Foundation, grant IUSE/PFE:RED: 1623189.

References

[1] S. Brayne, “Big Data Surveillance: The Case of Policing,” Am. Sociol. Rev., vol. 82, no. 5, pp. 977–1008, Oct. 2017.

[2] J. A. Buolamwini, “Gender shades : intersectional phenotypic and demographic evaluation of face datasets and gender classifiers,” Thesis, Massachusetts Institute of Technology, 2017.

[3] S. J. Ceci, W. M. Williams, and S. M. Barnett, “Women’s underrepresentation in science: Sociocultural and biological considerations.,” Psychol. Bull., vol. 135, no. 2, pp. 218–261, 2009.

[4] Committee on Women Faculty in the School of Science, “A Study on the Status of Women Faculty in Science at MIT,” MIT, Mar. 1999.

[5] S. Fowler, “Reflecting on one very, very strange year at Uber,” Susan Fowler, 19-Feb-2017. .

[6] E. Huet, “Facebook’s Hiring Process Hinders Its Effort to Create a Diverse Workforce,” Bloomberg, New York, NY, 09-Jan-2017.

[7] Female graduate students and research staff in the Laboratory for Computer Science and the Artificial Intelligence Laboratory at M.I.T., “Barriers to Equality in Academia: Women in Computer Science at M.I.T.,” M.I.T., Cambridge, MA, Feb. 1983.

[8] H. Natanson, “‘A Sort of Everyday Struggle,’” The Harvard Crimson, Cambridge, MA, 20-Oct-2017.

[9] M. S. Mahoney, “Boys’ Toys and Women’s Work: Feminism Engages Software,” in Feminism in twentieth-century science, technology, and medicine, A. N. H. Creager, Elizabeth. Lunbeck, and L. L. Schiebinger, Eds. Chicago: University of Chicago Press, 2001.

[10] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, “Machine Bias: There’s software used across the country to predict future criminals. And it’s biased against blacks.,” ProPublica, 23-May-2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. [Accessed: 24-May-2016].

[11] J. Damore, “Google’s Ideological Echo Chamber: How bias clouds our thinking about diversity and inclusion,” Jul-2017. .

[12] Y. Young, “Holder’s recommended Uber diversity initiatives failed,” San Francisco Chronicle, 06-Jul-2017.

[13] J. Rawls, A Theory of Justice, Rev. ed. Cambridge, Mass: Belknap Press of Harvard University Press, 1999.

[14] D. G. Langdon, Aligning performance: improving people, systems, and organizations. San Francisco, CA: Jossey-Bass, Pfeiffer, 2000.

[15] D. N. Brewis, “Social Justice ‘Lite’? Using Emotion for Moral Reasoning in Diversity Practice,” Gend. Work Organ., vol. 24, no. 5, pp. 519–532, 2017.

[16] S. Vallor, Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting. Cambridge: Oxford University Press, 2016.

[17] I. Sánchez-Cuenca, “A Preference for Selfish Preferences: The Problem of Motivations in Rational Choice Political Science,” Philos. Soc. Sci., vol. 38, no. 3, pp. 361–378, Sep. 2008.

[18] A. Collins, J. S. Brown, and S. E. Newman, “Cognitive Apprenticeship: Teaching the Craft of Reading, Writing and Mathematics,” Think. J. Philos. Child., vol. 8, no. 1, pp. 2–10, 1988.

[19] D. A. Schön, Educating the reflective practitioner: toward a new design for teaching and learning in the professions. San Francisco: Jossey-Bass, 1987.

[20] B. S. Bloom, Taxonomy of educational objectives; the classification of educational goals, 1st ed.]. New York: Longmans, Green, 1956.

Don Winiecki received the Ed.D. degree in Instructional Technology from Texas Tech University in 1996 and the Ph.D. degree in Sociology from Central Queensland University in 2006. His research and teaching addresses the affects of technologies in corporate and societal contexts, and the development of moral and ethical commitments and practices to steer development and/or use of technologies to support inclusion, diversity, and social justice.

Noah Salzman received the Ph.D. degree in Engineering Education from Purdue University. He is an Assistant Professor in Electrical and Computer Engineering and an affiliated faculty with the IDoTeach and Engineering Plus programs at Boise State University. His research interests include community formation in undergraduate programs, secondary school teaching career pathways and preparation for engineers, and students' transtions from pre-college to university engineering programs. Contact him at noahsalzman@boisestate.edu.