Adapting to Pervasive Computing, and Making Great Pedagogy Pervasive

Cynthia Lee
Stanford University
cbl@stanford.edu

ABSTRACT

This white paper outlines two major issues for the immediate future of computer science education: First, redefining our educational role in an academy where every field is deeply integrated with computing—adapting to pervasive computing. Second, working to better disseminate the pedagogical best practices developed and piloted in the CS education community—making great pedagogy pervasive. These two issues are related in that the current rapid expansion of teaching of computer science provides a valuable but rapidly closing window of opportunity to export our cutting-edge pedagogic best practices along with the export of our cutting-edge technologies.

1. Introduction

Computer Science is undergoing a period of rapid expansion and integration into every discipline of study. It is difficult to quantify such a shift, but this year’s Nobel Prize in chemistry provides an evocative anecdote. In honoring the recipients for modeling the chemistry of biological processes using computers, some saw it as a win for computer science as much as chemistry. [1] Evidence of the ubiquity of computing in other disciplines can also be found in awards in the popular press. Every one of CNN’s “Top 10 Thinkers” of 2013 [2] arguably falls under the purview of computer science, including a government official whose title is CTO, two educators working in educational technology, an epidemiologist who uses cell phone data to track malaria, and the maker of the software-heavy Tesla cars. What does it mean that so many of the best thinkers of our day are in computing, and what does it mean that so many of them were educated and housed in fields other than computer science?

Computer science education needs to adapt to address what will be a significant, qualitative, permanent change in who does computing, why, and how. At the same time that we face the challenge of adapting to this new terrain, the rapid expansion of who is doing—and teaching—computer science provides a unique opportunity to ensure that our known pedagogical content knowledge [11] and best practices are adopted as widely as our technology.

1.1 Related Work

As early as 1961, Alan Perlis identified computer programming as a skill that would eventually be considered a fundamental literacy and a component of liberal education, a theme reprised by Guzdial et al. in comparing favorably comparing programming to calculus in terms of its broad applicability [3,4]. Even as researchers in every department adopt computer science methods to accelerate their discovery and enable new lines of inquiry, computer science education remains far short of this broad new literacy vision at many higher education institutions. This is true whether measured in de jure terms—how often is computer science listed as a general education requirement—or in de facto terms—what percentage of non-majors take a computer science course. The situation is perhaps even more bleak at the K-12 level, where the Computer Science Teacher’s Association (CSTA) reported in 2013 that only 22 states even had any kind of certification or endorsement of teachers to teach computer science. [5]
CS education has long recognized non-major students as having distinct motivations, demographic profiles, prior knowledge, and needed skillset outcomes, compared to major students. Specifically tailored pedagogical approaches and course topic design have been proposed to meet the needs of these students. [3, 6, 7, 8] Computational thinking is one formulation of the kind of computing everyone should know. [9] And the nationwide, multi-institution-piloted CS Principles course was developed in part to meet the need for a first computing course that could also serve students well as a terminal course. [10]

2. New Questions about our Role
This existing work can inform us, but new thinking will be needed to address the profound change to our field that will occur when it is not simply that some number of non-majors and K-12 students wish to learn programming, but that the center of gravity of who is doing computing on university campuses and in industry is no longer concentrated in CS departments and explicitly tech sector industry. This raises organizational issues for departments that will impact curriculum design, and community issues for CS education research:

- If other departments teach computing, how does that change the mission and makeup of the CS education research community? How does it change our influence?
- Assuming non-CS faculty who are doing CS in their research will train graduate and undergraduate students in their work, what will inform the content and pedagogy of this training? How can we ensure that the CS education community has influence with these faculty, or even includes them?
- What happens when those students in turn educate students, creating academic lineages of people doing CS that are multi-generation disconnected from a CS department?

These guiding questions suggest several studies that could be conducted in the immediate future:

- Initial characterization of the phenomenon: take a sampling of a number of institutions, and search faculty research profiles to characterize the prevalence of faculty “doing CS” outside of CS departments. A key assumption is that these faculty and their work may not be known to CS department faculty. In which fields do we find these faculty? Are they incorporating CS topics or methods into classes taught? Are they sending students to CS departments for coursework?
- Efficacy studies of different models for managing pervasive computing: enlist departments in piloting different models of division of labor in teaching computer science. For example, the UCSD Computer Science Department investigated creating customized CS1 courses to appeal to each of several different campus communities whose students needed these skills (natural sciences, social sciences, arts, etc.). A proof of concept CS1 course specifically targeted to science majors and taught within the CS department proved much more successful than the course taught in the non-CS department that it replaced. [6] Does the experience of other institutions currently experimenting with these issues corroborate or differ from this experience?

3. Disseminating Pedagogical Innovations
CS education research has identified several best practices that have significant positive impact on student learning [13], representation of women [17], drop and fail rates in a given course [14], and longer-term persistence in the major. [15] Further, many of these pedagogical practices have been successfully exported to newly adopting faculty beyond the initial creator-researcher cohort. [16] These best practices include:

- Peer Instruction [12, 13, 14, 15]
- Pair Programming [15]
- Media Computation [7, 15]
- Harvey Mudd model for promoting gender balance [17]

A critical need now exists to grow our expertise, infrastructure, and production throughput, in the dissemination of these best practices.
3.1 New Questions

Dissemination of research findings and best practices is a process not yet well understood, and therefore should not be viewed by the CS education research community as simply a task to be completed after the real research is done, but a subject of research inquiry itself. Questions that need answering about the process of diffusion of innovation [18] include:

- What are the best ways of identifying and reaching faculty who teach or are preparing to teach computing, especially in a new environment where computing permeates all fields?
- Workshops are a common method of dissemination of new pedagogical practices, yet adoption rates by workshop attendees are often low, even when attendees feel the workshop was successful and inspiring. Are there critical success factors for workshops? Are there different, more effective ways of training new faculty?
- Given an assumption that one of the most effective moments to change pedagogy is with new, first-time teachers, are there ways we can take unique advantage of the current rapid pace of expansion of our field to maximize reach of dissemination activities and lock in a generation of high-quality pedagogy?
- What are the success factors for creating and sustaining online communities for providing ongoing support to faculty who are adopting new pedagogical practices?

ACKNOWLEDGEMENTS

My thanks to the coauthors with whom I have collaborated on work related to the topics discussed here: Leo Porter, Beth Simon, Christine Alvarado, Daniel Zingaro, and others.

Additional thanks to Mark Guzdial, whose blog and social media posts brought to my attention some of materials cited here, and which have been influential in my thinking.

REFERENCES


