Integrating Computer Science in Science: Considerations for Scale

Ari Krakowski, Lawrence Hall of Science, University of California, Berkeley, akrakowski@berkeley.edu
Eric Greenwald, Lawrence Hall of Science, University of California, Berkeley, eric.greenwald@berkeley.edu
Jake Duke, Lake Washington School District, Richmond, WA, jacduke@lwsd.org
Meghan Comstock, Pennsylvania University Graduate School of Education, mccom@gse.penn.edu
Natalie Roman, Lawrence Hall of Science, University of California, Berkeley, nroman@berkeley.edu

Abstract: While the push to integrate computer science (CS) within core subjects shows promise for broadening access to CS learning, research is needed to understand how such approaches can be realized at scale. Our ongoing research aims to design learning sequences that integrate CS and science learning at scale, and interrogate that design with practitioners. We report emerging design principles that can inform future efforts to integrate CS and disciplinary learning in required courses like science.

Introduction

Facility with foundational practices in computer science (CS) is increasingly recognized as critical for the 21st century workforce. Developing this capacity and broadening participation in CS disciplines will require learning experiences that can engage a larger and more diverse student population (Margolis et al., 2008). One promising approach involves including CS concepts and practices in required subjects like science. Yet, research on the scalability of educational innovations consistently demonstrates that their successful uptake in formal classrooms depends on teachers' perceived alignment of the innovations with their goals and expectations for student learning, as well as with the specific needs of their school context and culture (Blumenfeld et al., 2000; Penuel et al., 2007; Bernstein et al., 2016). Research is nascent, however, about how exactly to achieve this alignment and thereby position integrated instructional models for uptake at scale. To contribute to this understanding, we are developing and studying two units for core middle school science classrooms, known as Coding Science Internships. The units are designed to support broader participation in CS, with a particular emphasis on females, by expanding students' perception of the nature and value of coding. CS and science learning are integrated through a simulated internship model, in which students, as interns, apply science knowledge and use computer programming as a tool to address real-world problems. In one unit, students gain first-hand experience with sequences, loops, and conditionals as they program and debug an interactive scientific model of a coral reef ecosystem under threat. The second unit engages students in learning concepts related to data analysis and visualization, abstraction, and modularity as they code data visualizations using real EPA air quality data. A core goal for both units is to provide students experience with some of the increasingly prevalent ways that computer science is integrated into the work of scientists.

Methods

Data collection and analyses are grounded in principles from Design-Based Research (Barab, 2014) and Design-Based Implementation Research (Penuel et al., 2011), emphasizing iterative design and inquiry in collaboration with practitioners to understand what instructional design features position a CS-in-science integrated model for effective and scalable implementation. Reported findings focus on practitioner perceptions regarding implementation feasibility and alignment with school and district learning goals and are drawn from: a) iterative classroom piloting and observations of both instructional units in 19 classrooms, with 475 students; b) student survey and assessment data (n=391) from a research trial of the first of two 10-lesson instructional units; c) teacher surveys and reflection logs (n=10); d) teacher and district administrator interviews (n=8); and e) development team and stakeholder meeting notes from the first two years of the project.

Emerging Design Principles

Attending to time constraints. Teachers and administrators echoed the well-recognized feasibility constraint of *time* as an exceedingly precious commodity: the school day is limited and subject area teachers must contend with voluminous content standards and competing priorities for instructional time. Responsive to these constraints, we limited each instructional sequence to a unit of ten 45-minute lessons, which we have found to be feasible for uptake at scale in middle school classrooms (Jen et al., in review). In addition, we found that uptake of integrated instructional approaches was facilitated by attention to the implementation ecosystem, through supporting teachers and administrators in determining the timing of instructional sequences such that they interleave within and augment the school's adopted science curriculum progression.

Designing for face validity. Our experiences during recruitment and implementation suggest uptake in science classrooms will depend in part on whether teachers and administrators perceive CS integration to be legitimate science with instructional value. CS learning must *do work* for formal science classrooms, rather than simply "add coding" to the already ambitious learning goals set forth by science standards. Documentation that demonstrated alignment with science content (e.g., standards crosswalks with CS-in-science instructional sequences) proved invaluable for recruitment. Participating teachers also valued the opportunity to implement curricular sequences that demonstrated learning gains in computational thinking (CT), which represents a pivotal link between science and CS standards (Greenwald & Krakowski, 2020). This is especially salient because science teachers may recognize CT as a science and engineering practice in science standards, yet remain uncertain about how best to promote CT learning in concert with disciplinary learning.

Supporting teachers to navigate unfamiliar content and its integration. First, we found that instructional design features are critical to support the development of teacher capacity and confidence with CS. In particular, teachers in our study, most of whom identified as CS novices, consistently reported the value of extensive just-in-time teacher resources, including educative curriculum materials with background information and support documentation for using and supporting student engagement with the coding environment. Teachers also indicated that the simulated internship model itself allowed them to play more of a "guide on the side" role, shifting the onus of expertise and providing an opportunity to learn alongside their students. Second, we found that supports for differentiation are essential with integrated approaches in that students varied in both science and, especially, coding prior experiences. Accordingly, teachers regularly identified the differentiated resources within the units as critical to support the diverse learning needs in their classrooms. In addition to activity-based supports, we structured the design of coding tasks into clusters along a conceptual progression. This approach enabled all students to engage with core concepts for each cluster, while providing additional more scaffolded as well as more challenging tasks within each cluster to enable students to work at their own pace and in accordance with their prior coding experience.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1657002. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Barab, S. A. (2014). Design-based research: a methodological toolkit for engineering change. *Handbook of the Learning Sciences*, 2, 233-270.

Bernstein, D., Drayton, B., McKenney, S., & Schunn, C. (2016). Designing science curriculum for implementation at scale: Considerations for diverse and resource-limited settings. Singapore: International Society of the Learning Sciences.

Blikstein, P., & Wilensky, U. (2009). An atom is known by the company it keeps: A constructionist learning environment for materials science using agent-based modeling. *International Journal of Computers for Mathematical Learning*, 14(2), 81-119.

Blumenfeld, P., Fishman, B. J., Krajcik, J., Marx, R. W., & Soloway, E. (2000). Creating usable innovations in systemic reform: Scaling up technology-embedded project-based science in urban schools. *Educational psychologist*, 35(3), 149-164.

Greenwald, E. and Krakowski, A. (2020, March) Coding Science Internships: Broadening Participation in Computer Science by Positioning Coding as a Tool for Doing Science. In *Proceedings of the 50th SIGCSE Technical Symposium on Computer Science Education*.

Jen, T., Morales, C., Greenwald, E., Krakowski, A., Montgomery R., Loper, S. (2020) Enacting ambitious engineering curriculum in science classrooms: Examining teachers' implementation of Virtual Engineering Internships. Manuscript submitted for publication.

Margolis, J. (2010). *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press.

Penuel, W. R., Fishman, B. J., Haugan Cheng, B., & Sabelli, N. (2011). Organizing research and development at the intersection of learning, implementation, and design. *Educational researcher*, 40(7), 331-337.

Penuel, W. R., Roschelle, J., & Shechtman, N. (2007). Designing formative assessment software with teachers: An analysis of the co-design process. *Research and practice in technology enhanced learning*, 2(01), 51-74.