Considering theory in the design of CS education infrastructure: Three framings of computational thinking

Chris Proctor
cproctor@stanford.edu
Stanford University
Palo Alto, California

ABSTRACT

The development of computer science (CS) education infrastructure at the K12 and university level has largely sidestepped theoretical questions, positioning traditional cognitive approaches to computational thinking as unproblematic. This presentation considers how recent work organizing the theory space of computational thinking (into cognitive, situated, and critical computational thinking) might influence the nature of K12 CS education research questions and infrastructure used for teaching and assessing learning. An analysis of Unfold Studio, a platform for middle- and high-school literacy-based computer science education, illustrates how infrastructure could support theoretically-grounded pedagogy and research.

CCS CONCEPTS

• **Social and professional topics → K-12 education**.

KEYWORDS

Computational thinking, literacy, pedagogy, programming

ACM Reference Format:
Chris Proctor. 2019. Considering theory in the design of CS education infrastructure: Three framings of computational thinking. In *SPLICE: Computing Science Education Infrastructure: From Tools to Data, August 11, 2019, Toronto, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The questions of what should be taught in K12 computer science education and how have been coalescing around the concept of computational thinking [4, 13]. These have been contested issues. Some have argued for the importance of a unified approach to K12 CS education [1], while others have suggested that computational thinking has expanded to the point of incoherence [2]. The development of CS education infrastructure at the K12 and university level has largely sidestepped the question, positioning traditional cognitive approaches to computational thinking as unproblematic. In this presentation, we summarize recent work organizing the theory space of computational thinking, present a case study of a K12 CS education project explicitly positioned with respect to these framings, and reflect on how CS education infrastructure cold be developed with attention to various framings of what computer science learning looks like.

SPLICE '19, August 11, 2019, Toronto, CA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 THREE FRAMINGS OF COMPUTATIONAL THINKING

The idea of atheoretical infrastructure is tempting. Nelson and Ko [7] argue that while theory can be generative in CS education design research, theory can also impede progress in three ways: (1) research attention can be divided between creating better designs and generating theory, (2) reliance on general educational theory could detract from developing CS-specific theory, and (3) theory bias in peer review could unfairly privilege some research. Kafai, Proctor, and Lui [6] argue against the implicit suggestion that design research for computing education could choose to do without educational theory. Educational theory can play productive roles in design, such as organizing the search space of possible designs and as a heuristic, but theory also plays an essential role in defining the outcomes of educational research and practice. Design conjectures hypothesize relationships between tools, curriculum, and pedagogies the visible manifestations of mediating processes such as observable behaviors and learner-produced artifacts. Theoretical conjectures, in turn, hypothesize the relationship between these and outcomes such as various conceptions of learning [11].

Kafai, Proctor, and Lui [6] propose a framework to organize the theory space of computational thinking: cognitive, situated, and critical. Cognitive computational thinking focuses on individual learners and considers learning in terms of specified skills, competencies, and knowledge. Situated computational thinking sees learning as distributed in communities of practice, and considers learning in terms of participation and identity development. Critical computational thinking considers the role of learning in hierarchical power relationships across society, and considers learning in terms of action oriented toward justice. These framings represent different units of analysis, epistemological commitments, and stakeholders in K12 education. (Recent calls to reframe computational thinking as computational literacy [3, 5, 9] are based in the recognition that the construct can span these framings.) Recognizing these perspectives will be a necessary step to integrating CS into K12 education [8], and to broadening access at the university level.

3 OPERATIONALIZING THEORY IN UNFOLD STUDIO

Over several years of a design-based research process, Proctor and Blikstein [9] developed Unfold Studio, a free open-source web application for interactive storytelling, designed to support critical computational literacies. The web application was developed in conjunction with pedagogical strategies and curriculum designed to support students in representing, analyzing, and sharing their lived social experiences. The curriculum is centered around writer's workshop (open-ended writing/programming on topcis of students' choice) with periodic mini-lessons introducing computational concepts as they become relevant and interesting to students. Unfold Studio was also designed as a research instrument to answer questions spanning the three framings of computational thinking presented above. The following subsections explain how research questions are grounded in each of the three framings of computational thinking and how the application is instrumented to answer the questions.

3.1 Cognitive: How well did individual students learn to program?

A cognitive view of computational thinking will be primarily interested in individual students' competencies. Sequential code snapshots produce diffs showing the progress of individual stories. Each diff is composed of ops, additions, deletions, and changes to adjacent lines. Classifying ops and diffs allows assessment of the complexity of student code, as well as the student's ability to correct errors. Additionally, the curriculum unit features several assessments of programming skill. Students are asked to implement several puzzles, and are given pseudocode showing the author's intent and a broken story, and asked to fix it (based on the Fairy performance assessment [12]).

3.2 Situated: How did participants draw on literacy practices in developing computational thinking skills?

A situated view of computational thinking will be interested in the extent to which students participate in a community producing and consuming computational artifacts–using computational thinking in personally-meaningful ways for a real audience. Similar to Scratch, Unfold Studio allows users to see each other's shared stories, love stories, collect stories in books, and follow other users via an activity feed. The question of which interactions are computational, or at least which affect students' trajectories of learning computational thinking, can be answered by associating participation data with performance on the assessments described in the previous section, grounded by qualitative analysis of what students are writing and their accounts of why. Furthermore, social network analysis can show how practices such as using certain syntactic structures, spread through the community.

3.3 Critical: How did participants connect classroom literacy practices to their broader literacies via identity and voice?

A critical view of computational thinking will be interested in the extent to which students are able to deploy computational thinking as a resource in their existing literacy practices. Does computational thinking help them get respected at school, help them navigate multiple in-school and out-of-school identities, or help them deal with inequities related to gender, race, and social class? Here, the development and uptake of individual stories is particularly important. Qualitative analysis of successive story versions, backed by students' commit messages reflecting on their writing, shows the ways in which students bring these concerns into the classroom and make rhetorical use of the computational media to position themselves, introduce new perspectives, and challenge stereotypes. Logs of other students' interactions with these stories, including the sequence of choices they made as they played them, serve as evidence that these texts have an impact on changing the local culture. For example, Proctor and Garcia [10] analyze how a high school author helps her peers understand the dynamics of peer sexual pressure by strategically presenting players with choices.

4 DISCUSSION

This talk illustrates cognitive, situated, and critical views of computational thinking, showing how the infrastructure of Unfold Studio supports research questions framed in each view. It also argues that developers of K12 CS education infrastructure ought to consider how they frame computational thinking, and the kinds of data they collect and present. Infrastructure operationalizes theoretical constructs, regardless of whether this they are acknowledged. One of the difficulties in sharing K-12 curriculum has historically been that different schools serve very different populations, and have different kinds of educational goals. As we continue the important work of developing scalable infrastructure and sharable ontologies for curriculum, it would be valuable to be explicit in articulating theoretical stance.

REFERENCES

[1] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Inroads* 2, 1 (2011), 48–54.
[2] Peter J Denning. 2017. Remaining trouble spots with computational thinking. *Commun. ACM* 60, 6 (2017), 33–39.
[3] Andrea A DiSessa. 2001. *Changing minds: Computers, learning, and literacy.* Mit Press.
[4] Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher* 42, 1 (2013), 38–43. https://doi.org/10.3102/0013189X12463051
[5] Sharin Rawhiya Jacob, Mark Warschauer, University of California, Irvine, and University of California, Irvine. 2018-08-24. Computational Thinking and Literacy. 1, 1 (2018-08-24). https://doi.org/10.26716/jcsi.2018.01.1.1
[6] Yasmin B. Kafai, Chris Proctor, and Deborah Lui. 2019. From theory bias to theory dialogue: Embracing cognitive, situated and critical framings of computational thinking for K-12 CS education. In *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '19).* ACM, New York, NY, USA. http://chrisproctor.net/publications/kafai_2019_theory_dialogue
[7] Greg L. Nelson and Andrew J. Ko. 2018. On Use of Theory in Computing Education Research. In *Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18).* ACM, New York, NY, USA, 31–39. https://doi.org/10.1145/3230977.3230992 event-place: Espoo, Finland.
[8] Chris Proctor, Maxwell Bigman, and Paulo Blikstein. 2019. Defining and designing computer science education in a k-12 public school district. 7.
[9] Chris Proctor and Paulo Blikstein. 2019. Unfold Studio: Suporting critical literacies of text & code. *Information and Learning Science* 1, 2 (2019).
[10] Chris Proctor and Antero Garcia. 2019. Student voices in the digital hubbub. In *Giving student voice due weight: Possibilities and challenges in USA and New Zealand*, L Hogg and K Stockbridge (Eds.). http://chrisproctor.net/publications/proctor_2019_voice
[11] William Sandoval. 2014. Conjecture mapping: An approach to systematic educational design research. *Journal of the learning sciences* 23, 1 (2014), 18–36.
[12] Linda Werner, Jill Denner, Shannon Campe, and Damon Chizuru Kawamoto. [n.d.]. The fairy performance assessment: measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (2012). ACM, 215–220.
[13] Jeannette M. Wing. 2006-03. Computational Thinking. *Commun. ACM* 49, 3 (2006-03), 33–35. https://doi.org/10.1145/1118178.1118215