Toward a Computer Science Learning Progression: Investigating the Role of Adaptive Learning Environments for K–12

Kristy Elizabeth Boyer
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
keboyer, lester, bwmott, wiebe@ncsu.edu

James Lester
Computer Science

Bradford Mott
Computer Science

Eric Wiebe
STEM Education

1. INTRODUCTION

Creating a strong computing workforce is critical for maintaining our nation’s technological and scientific strength in a global society. The overall enrollment of students in computing disciplines has recently declined, and the nation faces a severe projected national computing workforce shortage (Zweben, 2013). The shortage is symptomatic of a broader problem: exceptionally low numbers of U.S. students in STEM disciplines. Policy reports on the state of the STEM workforce in the U.S. make it clear that addressing the shortage in the STEM workforce will require a multi-pronged approach at all grade levels. In response to this crisis, action is being taken along several fronts. The National Science Foundation has invested in numerous projects aimed at broadening participation in computing, many of demonstrate promising results for increasing the number and diversity of students who enter computing disciplines. Further, legislation has been proposed to strengthen K-12 computer science education. Many argue that if the U.S. is to maintain its economic leadership and compete in the new global economy, today’s K-12 students must be better prepared and encouraged to enter STEM careers. Recent years have thus seen an increased emphasis on introducing computer science to K-12 learners. The motivation for these activities is rooted in part in an understanding of how students eventually choose and succeed with an undergraduate major: previous positive experience with a subject matter plays a pivotal role (Arpaci-dusseau et al., 2013; Guzdial et al., 2010). This point has particular significance within the field of computer science; for example, a very small number of U.S. students take the traditional AP CS exam, and there is a troubling participation gap seen in that small body of test-takers (Arpaci-dusseau et al., 2013). The discrepancy corresponds to similarly disappointing statistics for students who earn bachelor degrees in computer science (Zweben, 2013).

Students begin their career trajectory in early K-12 years, in both their stated aspirations and their development of subject-specific skills (Lent et al., 1994). Researchers have looked specifically at how the underproduction and underrepresentation issues in undergraduate computer science departments may be traced back to lack of exposure as early as middle school (Shashaani, 1994; Webb, 2011). In the U.S., any K-12 computer science intervention must address the challenge that computer science is essentially absent from standard public education curricula. Numerous successful interventions have addressed this challenge through after-school, summer, or enrichment-based activities. For example, at the middle school level these curricula often use various visual programming languages (Lewis, 2010) or digital games (Webb et al., 2012), which hold broad appeal, and newer efforts focused at elementary school have also emerged (Gregg et al., 2012; Klopfer & Scheintaub, 2008; Lewis & Shah, 2012). Along with the curricular interventions, we are seeing efforts to formalize assessment of computer science or computational thinking at the K-12 level (Werner et al., 2012).

In order to develop a computer science learning trajectory for K-12 that can eventually permeate the U.S. educational system, several important research questions should be investigated:

---

1 CSTEM – http://www.cstem.org
3 House Resolution 2536 – https://www.govtrack.us/congress/bills/113/hr2536
1. How can computer science learning objectives be aligned with new curricular standards for K-12?
2. How can computer science be successfully taught by teachers who are certified in other subjects?
3. How can we build learning activities that are highly engaging and developmentally appropriate for students?
4. How can adaptive learning technologies support investigation of these research questions?

2. BACKGROUND

Middle School Computing Interventions. Recent years have seen an increase in the number of middle school computing interventions, many facilitated by visual or graphical programming languages. Alice 3D has been used successfully in middle school interventions for elementary, middle, and high school students (Rodger et al., 2010) to integrate computing within the context of a wide variety of subjects such as math, science, and language arts (Rodger et al., 2009a), and to help students understand what their future careers in computing fields might look like (Webb & Rosson, 2011). Scratch programming has also been used extensively in middle school programs (Lewis, 2010; Sivilotti & Laugel, 2008). Specialized interventions using Scratch have focused on students with physical or behavioral disabilities (Adams, 2010), targeted urban youth (Maloney et al., 2008), and incorporated culturally relevant themes for specific minority groups such as Hispanic students (Franklin et al., 2011). Along with Alice and Scratch, other languages such as Logo (Lewis, 2010) and more recently Kodu (Stolee & Fristoe, 2011) have also been used in middle school interventions, while some interventions focus very little on a programming language, such as a program that incorporates crafts into engineering and computing to engage young women of low socioeconomic status (Marcu et al., 2010).

Computational Thinking: A Door to K–12. Central to many STEM and ICT-intensive careers are core computer science concepts and habits of mind (CWCT, 2010; NRC, 2008). A strong computing and information technology workforce must not only be capable of deploying computing technologies, but also be knowledgeable of the underlying structures and mechanisms that make these technologies useful for problem solving (CSTA, 2011). The skills needed to prepare learners for engaging in creative problem solving with computing tools is termed “computational thinking” (CWCT, 2010; Wing, 2006). Historically, K–12 computer science curricula in the United States have been very limited (Fossati & Guzdial, 2011; Ni, Guzdial, Tew, Morrison, & Galanos, 2011). Programming-centric curricula, such as the AP Computer Science course for high school, have suffered from declining enrollment and are currently utilized on a very limited basis, especially among underrepresented groups in computing (Cuny, 2011). Because of this strong tension between workplace demand and declining enrollment, a novel AP Computer Science curriculum, CS Principles, has emerged as a complement to traditional programming-centric courses (Computer Science Principles, 2013). This new curriculum frames both K–12 computer science principles and the core tenets of computational thinking (Astrachan et al., 2011). The Next Generation Science Standards framework (NRC, 2011) and first drafts of those standards (Achieve, 2012) also present computational thinking as foundational to science and engineering practices and have incorporated it throughout the K-12 grade range. As articulated in the framework, computational thinking is not only foundational to computer science but also a key bridge to bringing large-scale data analysis and mathematical thinking into the STEM classrooms. This whitepaper presents a case study aligning a CS Principles topic, Big Data, with these emerging K-12 curricular standards.

3. PRELIMINARY RESULTS: BIG DATA CASE STUDY

Big Data is one of the foci of the proposed AP Computer Science Principles course (Computer Science Principles, 2013). Although a full formal definition of “Big Data” is still emerging, the phenomenon is often characterized by collection at high velocity, storage in a variety of structures, and large volume (Crawford, 2011). In addition to its widespread societal importance and increasing implications for the computer science workforce, Big Data holds great promise as a focus for K-12 computer science because of its rich interplay with other important CS Principles including algorithms, abstraction, and the Internet. Big Data also represents an area with strong correspondence between Computer Science Principles and emerging
U.S. national curricular standards such as the Common Core in Mathematics (CCSSI, 2012) and the Next Generation Science Standards (Next Generation Science Standards, 2013; NRC, 2011). In addition to its strong links to other core subjects, the topic of Big Data has great flexibility in incorporating other highly engaging application domains, including those that are relevant to children’s everyday lives. For instance, an excellent example of Big Data can be seen within social networks and their massive collections of structured and unstructured data. This case study examines the alignment of the CS Principles-based curriculum with current standards for middle school grade levels. We first analyze the Common Core State Standards for English and Mathematics and the Next Generation Science Standards.

3.1 Common Core State Standards
Beginning in 2009, the Common Core State Standards Initiative created a set of curricular standards intended to align the previously separate curricula of individual states within the U.S. According to its mission statement, these standards “are designed to be robust and relevant to the real world, reflecting the knowledge and skills that our young people need for success in college and careers” (CCSSI, 2012). Common Core for Mathematics laid out a series of high-level practices through which students are expected to find meaning in problems, use abstract and quantitative reasoning, construct arguments, critique the reasoning of others, and model with mathematics. Similarly, the CS Principles curriculum frames seven Big Ideas around the core computational thinking practices of connecting computing, developing computational artifacts, abstracting, analyzing problems and artifacts, communicating, and collaborating (Computer Science Principles, 2013). Table 1 highlights some important parallels between the CS Principles Computational Thinking (CT) Practices and the practices for the Common Core in Mathematics.

<table>
<thead>
<tr>
<th>CS Principles Computational Thinking Practices</th>
<th>Common Core Practices for Mathematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1: Connecting computing</td>
<td>MP5: Use appropriate tools strategically</td>
</tr>
<tr>
<td>P2: Developing computational artifacts</td>
<td>MP4: Model with mathematics</td>
</tr>
<tr>
<td>P3: Abstracting</td>
<td>MP2: Reason abstractly and quantitatively</td>
</tr>
<tr>
<td>P4: Analyzing problems and artifacts</td>
<td>MP1: Make sense of problems and persevere in solving them</td>
</tr>
<tr>
<td>P5: Communicating</td>
<td>MP3: Construct viable arguments and critique the reasoning of others</td>
</tr>
<tr>
<td></td>
<td>MP6: Attend to precision</td>
</tr>
<tr>
<td>P6: Collaborating</td>
<td></td>
</tr>
</tbody>
</table>

At a high level, both CS Principles and the Common Core include big ideas on data. While the Common Core standards relating to data do not map directly to the corresponding CS Principles Big Idea, we see the opportunity for a curriculum that serves the goals of both CS Principles and the Common Core in a complementary fashion. Table 2 shows Learning Objectives for CS Principles and selected standards from the Common Core that relate to data.

3.2 The Next Generation Science Standards
Similar to the Common Core, the precursor Framework report and Next Generation Science Standards in the U.S. (Next Generation Science Standards, 2013; NRC, 2011) does not explicitly address computer science, but holds great promise for the development of complementary curricula. The Framework report states that elements of both computational and mathematical thinking are central to K-12 science education, and these are addressed through the integration of science and engineering practices and crosscutting big ideas into the study of science concepts (NRC, 2011). These science and engineering practices recognize the importance of identifying student outcomes that go beyond memorizing narrowly focused scientific facts; they provide a broader interpretation of science inquiry that includes computational approaches. The Next Generation Science Standards stated practices include both analyzing and interpreting data, and using mathematical and computational thinking (Next Generation Science Standards, 2013). The Framework was clear that the eventual Standards documents needed to address how mathematical and computational tools and techniques are deployed in the service of
analyzing large data sets: “Such data sets extend the range of students’ experiences and help to illuminate this important practice of analyzing and interpreting data” (NRC, 2011). As this statement illustrates, there is great promise for creating curricula in which the power of computer science is illustrated within the context of widely required subject matter.

Table 2. Goals for Data in CS Principles and Common Core

<table>
<thead>
<tr>
<th>CS Principles Big Idea 3: Data and information facilitate the creation of knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1 Use computers to process information to gain insight and knowledge</td>
</tr>
<tr>
<td>3.1.2 Collaborate when processing information to gain insight and knowledge</td>
</tr>
<tr>
<td>3.1.3 Communicate insight and knowledge gained from using computer programs to process information</td>
</tr>
<tr>
<td>3.2.1 Use computing to facilitate exploration and the discovery of connections in information</td>
</tr>
<tr>
<td>3.2.2 Use large data sets to explore and discover information and knowledge</td>
</tr>
<tr>
<td>3.3.1 Analyze the considerations involved in the computational manipulation of information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Common Core Standards that relate to data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade 6. SP.A.1 Recognize a statistical question as one that anticipates variability in the data related to the question and accounts for it in the answers</td>
</tr>
<tr>
<td>Grade 6. SP.B.4 Display numerical data in plots on a number line, including dot plots, histograms, and box plots</td>
</tr>
<tr>
<td>Grade 6. SP.B.5 Summarize numerical data sets in relation to their context</td>
</tr>
<tr>
<td>Grade 7. SP.A.1 Understand that statistics can be used to gain information about a population by examining a sample of the population</td>
</tr>
<tr>
<td>Grade 7. SP.A.2 Use data from a random sample to draw inferences about a population with an unknown characteristic of interest.</td>
</tr>
<tr>
<td>Grade 7. SP.B.4 Use measures of center and measures of variability for numerical data from random samples to draw informal comparative inferences about two populations</td>
</tr>
<tr>
<td>Grade 8. SP.A.1 Construct and interpret scatter plots for bivariate measurement data to investigate patterns of association between two quantities</td>
</tr>
<tr>
<td>Grade 8. SP.A.4 Understand that patterns of association can also be seen in bivariate categorical data by displaying frequencies and relative frequencies in a two-way table.</td>
</tr>
</tbody>
</table>

4. CONCLUSION

Integration of computer science into K-12 is essential to developing the next generation of learners who will thrive and become leaders in a global society centered on ubiquitous computational technology. This whitepaper takes the position that computer science education researchers should invest effort in investigating research questions leading to a K-12 computer science learning progression. This investigation bridges an interdisciplinary gap between computer science and education, and as such, must address several important considerations. First, learning environments investigated at the K-12 level should take a holistic approach that includes the pedagogical and logistical context as well as the broader social and political climate. Second, the computer science education research community should leverage its expertise at designing and building innovative technology to build and investigate the impact of adaptive, individualized instruction that supports learners in a tailored way. Finally, all of these endeavors should be conducted in a student-centered approach. It is hoped that this work can substantially increase access to, and interest in, computer science for K-12 learners.

REFERENCES


