Supporting STEM Learning by Redesigning the Textbook

Creating High-Completion CS Online Learning Using Educational Psychology Principles

Mark Guzdial Barbara Ericson Briana Morrison Miranda Parker
Neeti Pathak Matthew Moldavan Kantwon Rogers

1 Redesigning Learning by Redesigning the Textbook

In computer science, research results have surprisingly little impact on teacher practices [6]. Computer science teachers are influenced by practical concerns. They care about what students say [5]. They care about having textbooks match the curriculum that they want to teach [11].

The hypothesis underlying our *CSLearning4U* project is that we can improve learning by improving textbooks. By redesigning the textbook, we can redesign the learning experience [15]. Even more important for our purposes, we aim to design textbooks to meet the learning needs of a particular populations.

The goal of the CSLearning4U project is to develop effective and more efficient computer science teaching and learning in order to serve professional development for high school teachers. Computer science education today is based on apprenticeship, where bits of instruction (in lecture, videos/MOOCs, or books) precede hours of student practice at a development environment. Our studies of high school teachers suggest that they need a different model:

- High school teachers are working professionals with significant time constraints. They often fail at on-line learning simply because of the external demands on their lives [8].
- The apprenticeship of computer science education match the activities of software developers, but not high school computer science teachers [16].
- The support for the teacher community is critical. Teachers who develop a sense of identity as computer science teachers are more likely to be retained, to join professional organizations, and to seek out additional professional development [25, 20]. Since few states offer computer science teacher certification, having a community with role models is an important part of developing that sense of identity [24].
- Teachers need a sense of confidence and self-efficacy to succeed as computer science teachers. Developing that confidence is related to computer science content knowledge and pedagogical content knowledge (*how* to teach) [24, 26, 23].

To meet the learning needs of high school CS teachers means that we change how we teach and what we teach.

Our goal is to support students and teachers learning the new CS Principles curriculum through innovative educational technology whose design is informed by educational psychology principles [12]. We are developing a new model for online learning using a dynamic, web-based book (or ebook), informed by principles of educational psychology. We believe that ebooks can have higher completion rates than a MOOC, with learning comparable to face-to-face classes.

While our focus is on computer science, the same approaches might be useful in other STEM disciplines. Our work uses educational psychology principles to design ebooks that meet the particular needs of our learners, high school computer science teachers. Different STEM learners have different learning needs. Our approach to design and the features we include in our ebooks may be useful in the design of textbooks for other STEM learners.

2 Designing an Ebook to Meet Learner Needs

We create a more efficient practice by drawing on educational psychology. We call our approach the *examples+practice model*, because we draw most significantly on the literature on creating effective examples [29, 31, 4] and on practice

(exercises used to encourage reflection on examples and develop skills) based on cognitive tutors research [3, 9, 2]. Our ebook for prospective teachers of the new AP CS Principles (CSP) course that incorporates findings from educational psychology: worked examples interleaved with practice activities, dual modality presentations, low cognitive load exercises including Parsons Problems, and social support from small groups.

We have described the structure of our ebooks in other publications [13, 14, 12]. Here, we describe one example. Figure 1 shows Python code running in the book, accessed from a Web browser. Note that the code is segmented by subgoal labels, which are common across multiple programs. The subgoal labels have been shown to lead to better learning and retention in several experiments[22, 21].

Several of the programs are challenges in the student version of the ebook. The teacher version of the ebook includes answers in a separate tab. There is support for discussion about each problem, also available in a separate tab.

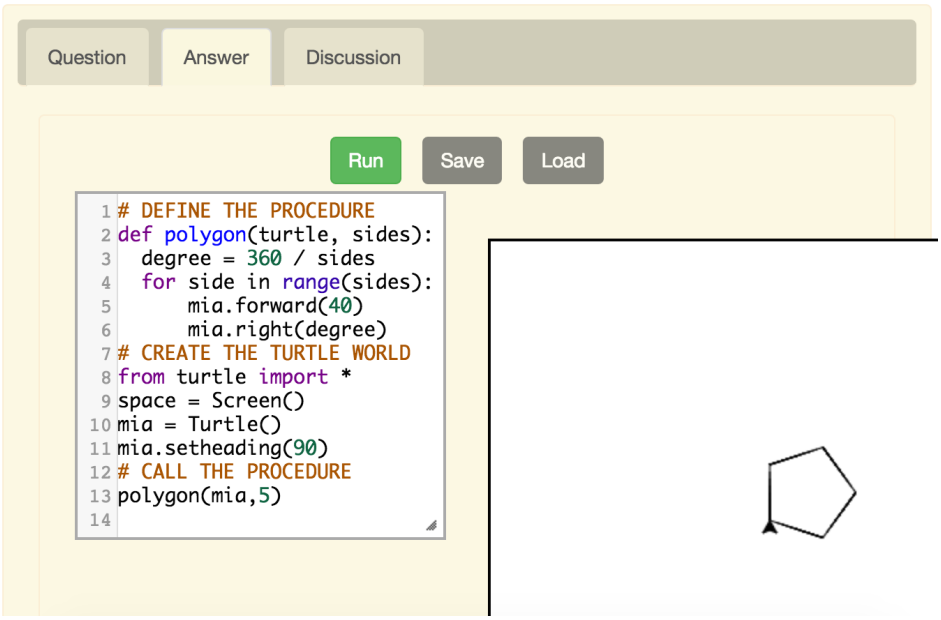

Figure 1: Executable code with subgoal labels, and answer and discussion tabs

Besides the discussion tabs, our teacher ebook includes support for small reading groups. A reading group can set a schedule for completing chapters of the book. The design goal is to create social pressure to complete the book and support a sense of community.

3 Results from our Ebook Studies

We have conducted several studies of our ebook. The first was a usability survey comparing three different ebook platforms: Runestone Interactive, Zyante, and CS Circles. Runestone did well in the survey, just beating out Zyante, and both were far ahead of CS Circles [13].

One of our surprises in use of the ebook has been the wide range of variability among students even on the low cognitive load practice activities. Computer science teachers expect a wide range of variability in coding activity [1, 19]. Our ebooks use a form of practice called *Parsons Problems* [27] where readers shuffle existing lines of code into a correct order. While many readers solve these problems quickly, a significant percentage of students take dozens of trials to solve them correctly [14]. Even in low cognitive load activities, we are challenged to support a wide range of student ability.

We conducted a trial with prospective high school computer science teachers. Ten teachers qualified for our study (i.e., scored less than 40% on a pretest). We asked them to read eight chapters of the ebook. Every two chapters, there was a post-test on those two chapters.

Some of our findings [13] include:

- 50% of the study participants finished all 8 chapters. That's good and higher than expected from studies of teachers learning in MOOCs [28], but isn't directly comparable to MOOC studies because we did offer them a

$50 gift card for completing.

- As we expected and designed for, teachers read the books in chunks of time when they could fit it in. Figure 2 describes one teacher's progression through the ebook, where each point represents use of a practice exercise. We see use of several exercises on one day, then a few more on the next day, and so on. Teachers are good learners who actively self-monitor and sometimes review or repeat prior exercises.

- Most importantly, those teachers who used the book (e.g., did the Parsons Problems, ran the code examples, answered multiple choice questions, etc.), reported increased confidence in teaching CS and they performed well on the post-tests. This is likely the first time that a study has shown improved learning and confidence by teachers without relying on program writing as the main practice activity. We are showing that writing code is not the only way to learn CS for high school teachers. Many CS teachers believe that apprenticeship is the only way to learn CS [18, 6, 7], but we believe that we can successfully use a range of pedagogical practices.

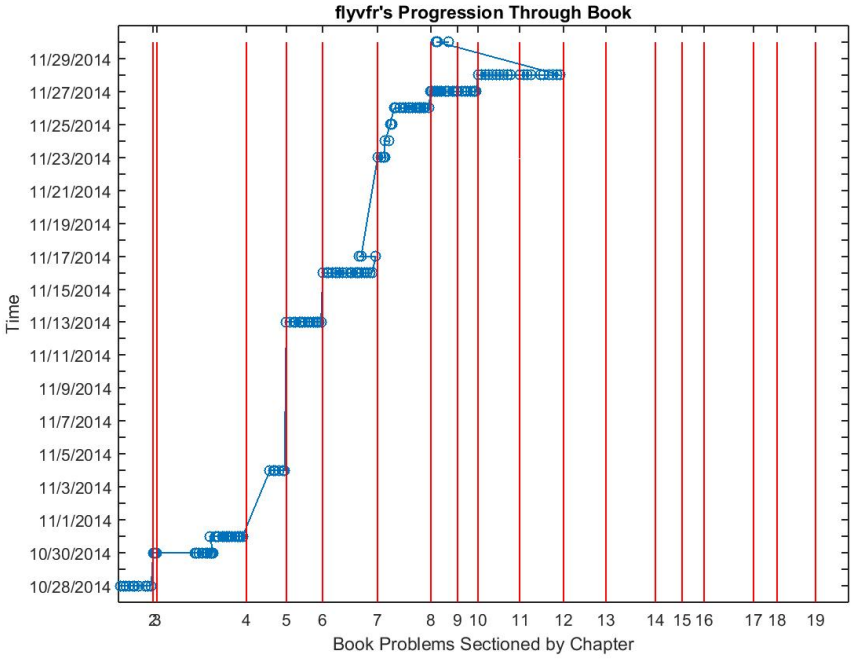

Figure 2: Charting a teacher's progress through the ebook exercises

3.1 Challenges to Conducting This Work

Our work aims to compare ebook use between teachers and students and between offline and online use. In one study, we aim to compare our CSP ebook to a face-to-face CSP class and to a set of students interested in studying CSP online. We plan to compare these students for learning outcomes and for completion rates. In a second study, we plan contrast our two CSP ebooks, one for teachers and one for students. By comparing these populations, we can develop a better understanding of how non-traditional and traditional students use e-books differently. We are building upon on our previous work in establishing valid and reliable measures of CS learning [30] to carefully measure reader's learning gains. We hypothesize that our approach will lead to higher completion rates, with significant learning gains, especially for underprepared learners.

It has been surprisingly difficult to get access to a CS Principles (CSP) face-to-face classroom in which we can trial our new interventions. Public high schools in our area are reluctant to provide access to their students for studies. Few universities are currently offering CSP.

We are finding that studying online learning is challenging in ways we had not expected. We just completed a larger study with many more teachers in Summer 2015, but are finding that the log data is difficult to trace. Teachers frequently forgot their usernames and simply created new ones, making log file analysis difficult. In our studies of

students, we find that teachers sometime have students reuse usernames, so that log entries for different students are indistinguishable.

4 Conclusion: Ebooks are a Promising Alternative to MOOCs

MOOCs have proven to be a successful form of professional development, especially for well-educated, affluent majority males [17, 10]. However, they are unlikely to be the best form of education for all students. In particular, the success rates with teachers are not particularly high [28]. We believe that ebooks are a promising medium for meeting the specific needs of computer science teachers.

More generally, we believe we should be designing textbooks to meet needs of learners. Our work with ebooks demonstrates the approach with high school teachers as our target learning population. We are creating a model for how new kinds of interactive textbooks might be developed for other STEM disciplines.

References

[1] A. Ahadi and R. Lister. Geek genes, prior knowledge, stumbling points and learning edge momentum: parts of the one elephant? In *ICER '13: Proceedings of the ninth annual international ACM conference on International computing education research*, pages 123–128, New York, NY, USA, 2013. ACM.

[2] V. Aleven, B. M. Mclaren, J. Sewall, and K. R. Koedinger. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2):105–154, 2009.

[3] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–208, 1995.

[4] R. K. Atkinson, S. J. Derry, A. Renkl, and D. Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, 70(2):181–214, 2000.

[5] L. Barker and J. Gruning. The student prompt: Student feedback and change in teaching practices in postsecondary computer science. In *Proceedings of the 2014 Frontiers in Education Conference*, 2014.

[6] L. Barker, C. L. Hovey, and J. Gruning. What influences CS faculty to adopt teaching practices? In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, pages 604–609, New York, NY, USA, 2015. ACM.

[7] L. J. Barker, K. Garvin-Doxas, and M. Jackson. Defensive climate in the computer science education. In D. Knox, editor, *The Proceedings of the Thirty-third SIGCSE Technical Symposium on Computer Science Education, 2002*, pages 43–47. ACM, 2002.

[8] K. Benda, A. Bruckman, and M. Guzdial. When life and learning do not fit: Challenges of workload and communication in introductory computer science online. *ACM Transactions on Computing Education*, 12(4):1–38, 2012.

[9] H. Cen, K. R. Koedinger, and B. Junker. *Learning factors analysis: A general method for cognitive model evaluation and improvement*, pages 164–175. Springer-Verlag, Berlin, 2006.

[10] G. Christensen, A. Steinmetz, B. Alcorn, A. Bennett, D. Woods, and E. J. Emanuel. The MOOC phenomenon: Who takes massive open online courses and why? *Social science research network*, Abstract 2350964, 2013.

[11] N. B. Dale. Textbooks: How we choose them, how we use them, shall we lose them? In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*, ACE '10, pages 3–8, Darlinghurst, Australia, Australia, 2010. Australian Computer Society, Inc.

[12] B. Ericson, M. Guzdial, B. Morrison, M. Parker, M. Moldavan, and L. Surasani. An ebook for teachers learning cs principles. *ACM Inroads*, 6(4):84–86, Nov. 2015.

[13] B. Ericson, S. Moore, B. Morrison, and M. Guzdial. Usability and usage of interactive features in an online ebook for CS teachers. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, WiPSCE '15, pages 111–120, New York, NY, USA, 2015. ACM.

[14] B. J. Ericson, M. J. Guzdial, and B. B. Morrison. Analysis of interactive features designed to enhance learning in an ebook. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ICER '15, pages 169–178, New York, NY, USA, 2015. ACM.

[15] S. Esper, B. Simon, and Q. Cutts. Exploratory homeworks: An active learning tool for textbook reading. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research*, ICER '12, pages 105–110, New York, NY, USA, 2012. ACM.

[16] M. Guzdial. *Learner-centered design of computing education: Research on computing for everyone.* Synthesis Lectures on Human-Centered Informatics. Morgan and Claypool, 2015.

[17] S. Kolowich. The MOOC 'revolution' may not be as disruptive as some had imagined. *The Chronicle of Higher Education*, 2013.

[18] C. Lewis, M. H. Jackson, and W. M. Waite. Student and faculty attitudes and beliefs about computer science. *Commun. ACM*, 53(5):78–85, May 2010.

[19] R. Lister. Geek genes and bimodal grades. *ACM Inroads*, 1(3):16–17, Sept. 2011.

[20] A. L. Luehmann. Identity development as a lens to science teacher preparation. *Science Education*, 9(5):822–839, 2007.

[21] L. E. Margulieux, M. Guzdial, and R. Catrambone. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research*, ICER '12, pages 71–78, New York, NY, USA, 2012. ACM.

[22] B. B. Morrison, L. E. Margulieux, and M. Guzdial. Subgoals, context, and worked examples in learning computing problem solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ICER '15, pages 21–29, New York, NY, USA, 2015. ACM.

[23] B. B. Morrison, L. Ni, and M. Guzdial. Adapting the disciplinary commons model for high school teachers: improving recruitment, creating community. In *Proceedings of the ninth annual international conference on International computing education research*, ICER '12, pages 47–54, New York, NY, USA, 2012. ACM.

[24] L. Ni. *Building professional identity as computer science teachers: Supporting high school computer science teachers through reflection and community building.* PhD thesis, Georgia Institute of Technology, Atlanta, GA, December 2011.

[25] L. Ni and M. Guzdial. Prepare and support computer science (CS) teachers: Understanding CS teachers' professional identity, 2011.

[26] L. Ni, M. Guzdial, A. E. Tew, B. Morrison, and R. Galanos. Building a community to support HS CS teachers: the disciplinary commons for computing educators. In *SIGCSE '11: Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 553–558, New York, NY, USA, 2011. ACM.

[27] D. Parsons and P. Haden. Parson's programming puzzles: A fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, ACE '06, pages 157–163, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[28] C. Spradling, D. Linville, M. P. Rogers, and J. Clark. Are MOOCs an appropriate pedagogy for training K-12 teachers computer science concepts? *Journal of Computer Science in Colleges*, 30(5):115–125, May 2015.

[29] J. Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12:257–285, 1988. Theory behind worked examples.

[30] A. E. Tew and M. Guzdial. The FCS1: a language independent assessment of CS1 knowledge. In *SIGCSE '11: Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 111–116, New York, NY, USA, 2011. ACM.

[31] J. G. Trafton and B. J. Reiser. *The contributions of studying examples and solving problems to skill acquisition*, pages 1017–1022. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1993.