CS Teacher Experiences with Educational Technology, Problem-Based Learning, and a CS Principles Curriculum

George Veletsianos
Royal Roads University
veletsianos@gmail.com

Bradley Beth
University of Texas at Austin
bbeth@cs.utexas.edu

Calvin Lin
University of Texas at Austin
lin@cs.utexas.edu

ABSTRACT

Little is known about how K–12 Computer Science (CS) teachers use technology and problem-based learning (PBL) to teach CS content in the context of CS Principles curricula. Significantly, little qualitative research has been conducted in these areas in computer science education, so we lack an in-depth understanding of the complicated realities of CS teachers' experiences. This paper describes the practices and experiences of six teachers' use of technology that was implemented to support PBL in the context of a dual enrollment CS Principles course.

Results from an early offering of this course suggest that (1) while CS teachers used technology, they did not appear to use it to support student inquiry, (2) local adaptations to the curriculum were largely teacher-centric, and (3) the simultaneous adoption of new instructional practices, technologies, and curricula was overwhelming to teachers. This paper then describes how these results were used to modify the curriculum and professional development, leading to increased teacher satisfaction and student success in the course.

1. INTRODUCTION

Most computer science curricula focus either on coding or technology applications (e.g., how to use the Microsoft suite of office productivity tools) rather than on the broader principles of computing and how computing impacts our world. For instance, recently, Code.org, a 501(c)(3) non-profit organization based in the US, launched several initiatives aiming to expand participation among underrepresented groups in computing. The goal of these initiatives is to increase exposure to computer programming in K–12 schools, and includes examples such as the following:

- The *Hour of Code*, a nationwide effort to introduce computer programming to students through participation in a variety of online tutorials. These tutorials are designed to give a brief overview of programming in the space of one hour and are targeted primarily toward K–8 students.

- Code.org promotional videos featuring industry leaders (e.g., founders of Microsoft, Facebook, Twitter, etc.), politicians, and celebrities advocating for learning computer science through writing code. These high-profile initiatives seek to encourage individuals to consider CS-related fields, but they equate computer science with *writing code.*

Computer Science courses that focus on computer programming or applications are not engaging high school students [26]. To address its shortage of students and diversity, computer science education needs to address both the content and the pedagogies of introductory computer science courses. The paucity of K–12 CS courses that feature rich content, student-centered practices, and technology-enhanced environments is due in part to the lack of exemplary models and research. As a result, the CS education community lacks knowledge of teacher experiences with CS courses that make use of student-centered pedagogies. This paper addresses these gaps in the literature.

In particular, this paper makes the following contributions:

- We present the experiences of six high school teachers who implemented a dual enrollment CS Principles curriculum in 2012.

- We report three main findings: (1) Teachers made use of technology, but not to support student inquiry; (2) teachers largely did not adopt PBL pedagogies, except in a module that focused on programming; and (3) teachers were largely overwhelmed by the simultaneous adoption of a new curriculum, a new pedagogy, and new learning technology.

- We describe the subsequent changes to the curriculum and provide anecdotal evidence that these changes were effective in both improving teacher satisfaction with course content and pedagogy and increasing student completion rates for college credit.

The remainder of this paper is organized as follows. We first describe the course that the teachers implemented. Next we review relevant literature and the methods used to conduct this investigation. We conclude by presenting our findings and their implications.

SIGCSE '16, March 02 - 05, 2016, Memphis, TN, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ISBN 978-1-4503-3685-7/16/03…$15.00

DOI: http://dx.doi.org/10.1145/2839509.2844645

2. BACKGROUND

Thriving in Our Digital World is an introductory high-school computer science course that uses an inquiry-based curriculum and a blended learning environment to improve the accessibility, rigor, and relevance of computer science. The course content is organized by a "framework of the principles and practices of computing" called CS Principles. Developed by the NSF and College Board, CS Principles covers a broader array of computer science than simply coding, including big ideas like programming and algorithms, but also other topics like data analysis, digital manipulation, and ethics. The course also strives to show students that computing is applicable to both local and global issues, and a variety of different fields. The curriculum uses student-centered practices and inquiry-driven pedagogies, like problem-based learning (PBL) based on the work of Krajcik and Blumenfeld [18] and the Jasper Woodbury Problem Solving Series [3].

The structure of *Thriving in Our Digital World* is modular, with a separate, overarching project driving each module's content [4]. Course modules begin with the presentation of a driving problem that situated the learning activities that follow. Students engage in authentic activities, collaborate with peers, and use technology to facilitate learning as they develop a product to address the question/challenge. Each problem is 'launched' via an anchor video [19] and students collaborate in small groups to develop their final solutions. Throughout each module, teachers also present content, provide mini-lessons and tutorials, implement formative assessments (e.g., peer collaboration evaluations), and orchestrate activities intended to support students in completing their projects (e.g., rubric checks). The learning environment itself is blended, with course materials (e.g., videos, guiding activities, rubrics, reading assignments, etc) organized within a Learning Management System (LMS), and face-to-face meetings occurring according to schools' regular bell schedules.

There are several reasons why this course appealed to high school administrators and teachers, including (1) its dual credit format (i.e. students who pass this course are able to apply the credits they earn towards their college education in the state in which this course was being developed), and (2) the nature of the content. The course has been designed in an iterative manner over a five-year period; this paper focuses on teachers' initial experiences with embedded learning technologies in the course and how these experiences informed subsequent course revisions.

3. RELATED WORK

Even though there is abundant literature on technology integration and inquiry learning, we do not have a clear understanding of how blended environments support inquiry learning specifically, especially in computer science education. Technology integration has been described as a "wicked problem," one which is complex and involves "the convoluted interaction of multiple factors, with few hard and fast rules that apply across contexts and cases" [22]. Mere access to technology does not necessarily result in effective technology integration [9], and efforts at technology integration are often limited to activities that aim at making known instructional approaches faster or cheaper [15]. Nevertheless, empirical research suggests that technology integration can create conditions that positively affect learning [3, 8, 16, 29]. A second-order meta-analysis of 25 meta-analyses and 1,055 individual studies for example, showed that learning outcomes are significantly and positively affected by learning technology integration in experimental research [27].

Effectively integrating instructional technology and inquiry-learning pedagogies into classroom instruction can be challenging. Numerous researchers have sought to articulate what teachers need to know in order to effectively integrate technology in their classrooms. One of the most popular frameworks over the last decade is the Technological Pedagogical And Content Knowledge (TPACK) framework [22]. Though Brantely-Dias and Ertmer [6] argue that the TPACK construct is both "too vague and too intricate" the framework purports that teachers should be knowledgeable of (1) their content area, (b) pedagogies that support learning of said content, and (3) how technology can support or facilitate content-specific pedagogies, and (4) the interactions between these three knowledge areas. This suggests that TPACK proficiency is more challenging to acquire than technological, pedagogical, or content knowledge proficiency alone. Furthermore, achieving TPACK proficiency may even be more difficult for teachers who adopt pedagogical approaches that are unfamiliar to them citees06. For example, PBL approaches invite teachers to take on unfamiliar roles (e.g., facilitating collaboration) and engage in unfamiliar activities (e.g., scaffolding problem-solving). Blended learning environments introduce additional challenges, too [11]. For instance, online course calendars must adjust to a variety of local events (e.g., state-mandated testing), and the amount of digital work produced by students can be prolific. Faced with such obstacles, teachers are likely to require tremendous support as they learn new pedagogical approaches that may conflict with pre-existing beliefs and practices [3].

While computer science teachers are likely to be proficient in their content area, the lack of CS-specific teaching methods courses and CS teaching certification programs across the U.S. [12, 30] suggests that CS teachers may lack content-specific pedagogical knowledge. This problem may be exacerbated by the fact that research on content-specific pedagogies for computer science education is minimal [14]. Likewise, research on the use of inquiry-based pedagogies in K–12 computer science classrooms is scant. An analysis of computer science education research shows that the plurality of computer science education research articles were program descriptions or contained only anecdotal evidence [24]. Our review of the literature on CS teachers' experiences in K–12 classrooms revealed a scarcity of relevant articles. In particular, we were able to locate empirical articles describing strategies that CS teachers employ to make CS interesting to students [5] and noting that it may be productive for teachers to view their interactions with students as connecting with a different culture [17]. We also located articles explaining how CS teachers experience the success of their students [7] and illustrating how they perceive the use of multiple choice assessments [25]. However, evidence of CS educators' experiences with inquiry learning in blended learning environments was minimal. Thus, more empirical research that examines CS educators' experiences and practices is necessary especially at a time when the CS community is seeking to increase the number of CS teachers and the effectiveness and diversity of CS education [2].

4. RESEARCH QUESTIONS

The field lacks an understanding of CS teacher practices and experiences with inquiry-driven learning and blended environments. This research project attempts to address this need by answering the following research question: How did teachers respond to a student-centered and technology-rich curriculum?

5. METHODOLOGY

5.1 Participants

In this initial pilot offering, six male computer science teachers were trained to teach *Thriving in Our Digital World* through a two week summer professional development and taught it in the following school year. Each agreed to participate in this study under the assigned pseudonyms Bob, Ken, Mark, James, Tyler, and Sam. As this was the first year of implementation, we targeted teachers with prior experience teaching high school computer science coursework.

Bob and Ken teach in small rural high schools, while Mark, James, Tyler, and Sam teach at high schools located in a large metropolitan area. Bob and Ken teach at predominantly white schools and have taught courses related to computer science for 15 and 20 years respectively. Mark has taught computer science related courses at an economically disadvantaged, ethnically diverse high school for five years. James has taught computer science at a magnet high school for less than five years, and at the same magnet school, Sam was teaching computer science for the first time. Tyler has taught computer science at a technology-based high school for less than five years. Tyler's school is infused with technology and utilizes a problem-based learning model school-wide. Tyler was an experienced problem-based teacher, whereas all other participants were novices to inquiry learning. Participants' classrooms each have a digital projector or interactive whiteboard, wireless Internet connections, and at least a 1:1 computing device to student ratio (desktop computer, laptop, or tablet). Class sizes ranged in size from 9 to 26 students.

5.2 Data Sources

To address the research questions, we collected qualitative data from classroom observations, teacher interviews, and student focus groups. Most computer science education research is quantitative in nature and frequently relies on experimental designs [24]. Qualitative methods are infrequently used in computer science education research [1, 10, 23], yet are capable of generating significant insights into the activities, practices, perceptions, ways of thinking, and overall lived experiences of computing education stakeholders. Our study is informed by six different classroom cases that include 14 one-to-one participant interviews, 28 observations of participants' classrooms, and numerous researcher field notes. Semi-structured teacher interviews were conducted at the beginning, in the middle, and at the end of the school year. Classroom observations occurred more regularly. Each interview lasted between 30 and 60 minutes, was audio-recorded and transcribed.

5.3 Data Analysis

Working within the interpretive research paradigm [21], data was analyzed using the constant comparative method [13]. Two researchers independently read and coded the data. With each new data that entered the analysis (e.g., a sentence, an interview), each researcher compared the data to existing codes and created new codes if necessary. After independent analyses were complete, researchers compared notes, discussed codes, and re-analyzed the data independently. Next, codes were combined to form themes and this process continued until consensus was reached and it was felt that data saturation[1] had been reached. Though the results presented are bound by the context within which the project was implemented, the results are presented using rich descriptions so as to enable readers and other researchers to judge the extent to which these results apply to their own contexts (c.f. Merriam [20]).

6. FINDINGS

Results from this investigation are presented in three themes. These themes capture some, but not all, of the experiences/activities of these six teachers. These themes are reported because they are novel in the CS education literature and bear significant implications for CS education.

6.1 Use of Technology in the Classroom

Although technology was integrated heavily into instruction as a result of the design of *Thriving in Our Digital World*, teachers did not use the technology scaffolds as prescribed. The learning technologies integrated into the curriculum were designed to be flexible so that order teachers could alter them as necessary (e.g., create/delete/moderate discussion board threads), and this flexibility resulted in both desired outcomes and unintended consequences.

Most integration efforts either replicated or amplified classroom practices, but they were rarely entirely transformative. At each school, students were online for the entirety of almost every class, so most work was completed and submitted digitally and asynchronously. Throughout the duration of the course all teachers accessed static content and assignment pages weekly, and in some cases daily. Some teachers altered these materials to better suit their needs. For instance, the online quiz feature of the blended environment was instrumental to Ken's course implementation, and he felt that the online quizzes helped learners succeed in the course. Ken created a static webpage within the LMS that contained a calendar of his class's daily activities and more:

> "I made up my own quizzes, and had [students] do additional stuff to get them prepared for that test."

Teachers also used digital tools in teacher-centric ways, such as when James and Sam requested or created supplementary Powerpoint presentations to deliver content to students on a regular basis. These efforts helped students and teachers remain organized, reduced paper usage in the classroom drastically, and provided another venue for access to course materials, but none of these activities transformed the teaching and learning process.

In general, most teachers utilized very few of the technology scaffolds to support inquiry-based and PBL learning pedagogies as intended, though many alternative integration efforts could clearly support PBL. Teachers tend to feel more comfortable integrating technologies that they are familiar

[1]Saturation is defined as the point at which researchers felt that additional analyses were yielding no new insights.

with than those that are strange to them. For instance, James already used Google Drive for classroom management and organization, but for *Thriving in Our Digital World*, he integrated into this course as a collaboration hub for groups, too. This helped James achieve his "really big goal that whatever we cover in class is easily available to students outside of class without having to use...district servers and stuff, which can be hard for students to access." Additionally, his students authored weekly blogs about the course, as they did in his other classes, allowing students to reflect on their learning. There was little need for him to facilitate online discussions via the discussion forums or to have groups collaborate using the various tools in the LMS, because he was already achieving the major goals he had set out to achieve using alternative means. Thus, use of these tools was rare, not just for James, but for the rest of the teachers as well. At the beginning of the school year, all teachers used the Discussions feature in the LMS to facilitate online discussions and group collaboration, but their usage practically ended after the initial module, with some teachers, like Ken, deleting the links to the discussions from the course outline, and other teachers, like Sam, avoiding them.

The blended environment seemed to support technology integration, but not inquiry learning or PBL instruction specifically.

6.2 Use of Pedagogical Innovation

Teachers often appeared to adapt the curriculum to meet instructional goals they deemed worthwhile, sometimes in ways that diverged from inquiry-based pedagogies. For instance, teachers integrated more lectures, homework, and quizzes than found in the original curriculum. James said that his "teaching philosophy centers around the idea that I want to get information across to the students as quickly as possible," so most of his course adoptions "centered around presentations." However, while many activities created by the teachers were teacher-centric, their pedagogical practices were inquiry-based and student-centered during the programming module of the pilot curriculum, which is the one module composed of the content most commonly associated with computer science instruction. Tyler, Bob, James, Ken, and Sam introduced pair programming, peer tutoring, and one-on-one tutorials to support self-paced learning activities in this module. The Programming activities required students to produce learning artifacts and included increased elements of student choice (e.g., program a virtual keyboard or a virtual paint palette) more often than in other modules. Students also progressed through the Programming module curriculum mostly at their own pace. According to Ken, "In the Programming unit [students] were able to just jump in and go, and I did not deliver a lesson, not once during that unit. I just didn't have to." Perhaps, teachers implemented more inquiry learning activities during this particular module, because they were more familiar with the content, they had greater experience in teaching programming, or because they simply enjoyed programming itself. Tyler for example said that he "did a lot of cheerleading for the Programming (module)" that he did not do for the previous modules.

6.3 Overwhelming Teachers with Changes

The simultaneous adoption of a pilot curriculum, PBL pedagogies, and blended learning environments appeared to overwhelm these teachers. Barriers to technology integration (e.g., classroom management skills), to the adoption of pilot curricula (e.g., learning new content), and to teaching in blended environments (e.g., organizing course materials and calendars) combined forces to inhibit instruction and exacerbate the challenges that teachers faced in their classrooms. For example, the challenges of classroom management associated with technology integration amplified the challenges of classroom management using student-centered pedagogies. For instance, James and Sam reported that they "got burnt out" by the quantity of online discussions, which they felt obligated to grade.

The adoption of new pedagogies, new content, and a new online learning environment required significant amounts of teacher preparation. All teachers mentioned spending more time preparing for this course than any other course, often struggling to stay ahead of the students. "I don't know how a busy person with a lot of kids [students] could teach this course," Bob conjectured. Mark said that "it can be overwhelming when you're throwing something new in there [that] you really haven't had a chance to get used to." James explained that he "probably went over (the lesson) the night before or the day of, hoping that I didn't find anything that was too wrong to be able to fix in that amount of time." Teachers had originally anticipated that the pilot curriculum would be ready to be implemented, but adopting curricular materials to local contexts took time. For instance, Tyler initially thought, "Great! You guys have done all the work for me," only later to realize "that there is a lot of work in taking others' stuff, and fitting it into what we usually do."

Teachers also yearned to learn the new content associated with the curriculum and explore the online materials prior to face-to-face class sessions and this imposed additional time demands. Sam "would love to have sat in on a course from someone that developed (this course's curriculum)," so that he could have discussions about the content and potential questions that his students might have had. Mark "didn't have enough time last summer to really get into (the curriculum), and be a month ahead of the kids." He would have preferred an opportunity to "turn [himself] into a student and do every single thing...that [he] would ask [his] students to do." Unfamiliarity with the content and pedagogies of the pilot curriculum hindered implementations, because teachers had to spend time learning the content and the instructional approaches associated with it. Furthermore, integrating new technologies and new pedagogies also required additional preparatory time. Bob felt the curriculum was "pretty open ended and it's hard on me to grade that kind of stuff."

None of these experiences happened in isolation, and as a result they compounded each other. The difficulties associated with simultaneous paradigm shifts are noteworthy and computer science developers should take into account the challenges that teachers may face when asked to engage in activities that require rethinking of content, pedagogy, and technology integration.

7. INNOVATIONS

These findings from the initial pilot of *Thriving in Our Digital World* have informed subsequent refinements to the course through (1) significant revisions in course design (further detailed in Veletsianos et al. [28]) and (2) an enhanced professional development sequence. The targeted areas of revision may be distilled to the following:

- **Daily usage of technology scaffolds is integrated into course expectations.** Rather than providing collaborative tools as course resources, explicit usage of discussion boards, wikis, and collaborative workspaces (e.g., Google docs) is detailed in the provided lesson plans, scope and sequence, and suggested assessments. Teachers are provided with scripted discussion seed questions and responses. Students are assessed on their use of collaborative spaces to create and submit group projects.

- **Pedagogical innovation is an explicit objective of professional development.** Rather than focusing solely on the student experience within the LMS, *Thriving in Our Digital World* professional development now devotes a significant portion of time to developing teacher-created artifacts. Teachers are provided guided practice in tailoring stock exercises to their student populations and sharing those innovations with other teachers remotely.

- **Professional development is approached holistically, emphasizing the overall course experience.** The initial professional development sequence preceding the pilot year of implementation was presented in a discrete subject format, where each of the course components were addressed in relative isolation. Our revised professional development curriculum now better integrates course components such as PBL, blended learning, and the course content through an extensive linear introduction to the course materials. This revision borrows heavily from the introductory module provided to students, explicitly introducing each of the core course components in the shared context of a unit project.

Although the effects of these innovations have not been directly studied, anecdotal evidence suggests that increases in teacher satisfaction and student success may be due in part to these targeted revisions. Teacher focus groups, surveys, and observation data analyzed by an external evaluator suggest that teacher confidence in both teaching the course content and implementing the course pedagogies (primarily PBL) greatly increased from year one to two of implementation. Whereas a majority of teachers expressed doubt over the efficacy of PBL and its connection to course content in the initial course offering, nearly all of the teachers in the subsequent year were overwhelmingly positive.

Additionally, student success as measured by completion rates for college credit rose dramatically from year one (72%) to two (88%). This increase is sustained over year three, where 86% of students completed the course with college credit.

8. CONCLUSION

This project examined the experiences and practices of a group of teachers asked to make rapid changes to their beliefs and practices about pedagogy, technology integration, and computer science content. Though past research shows that technology integration can support PBL, these cases reminds us that PBL and technology integration may also complicate each other under less favorable circumstances, such as when teachers are simultaneously responding to new pedagogies, curriculum, or technologies.

A significant implication of this research is the recognition that even though computer science teachers may be experts in computing, they may not necessarily be proficient in the use of educational technology. Teachers' expertise in computing, like software development or systems analysis, did not necessarily equate to expertise in technology integration for achieving desired learning outcomes. For instance, this group of teachers seemed unfamiliar with a few of the more commonplace learning technologies that were heavily integrated into the curriculum: Bob, Ken, and Mark lacked prior experience with social networking sites and had never used online collaborative documents prior to being asked to implement this curriculum. Computer science teachers likely need to develop their knowledge of how technology can enhance learning experiences and how it can support pedagogical innovation.

Computer science teachers tasked with implementing novel pedagogical practices require extensive support and assistance. Increased teacher satisfaction and student success after informed revisions to *Thriving in Our Digital World* curriculum and professional development bear this out. Future initiatives would benefit from considering teachers' beliefs, practices, content knowledge, technology knowledge, and readiness to adopt new pedagogical approaches. Equally important, given the dearth of research in CS teacher experiences and CS teacher preparation programs, more research on CS teachers' pedagogical practices and technology integration activities is both pressing and necessary.

9. ACKNOWLEDGMENTS

Thriving in Our Digital World is based upon work supported by NSF Grant DRL-1441009, NSF Grant CNS-1138506, and by the University of Texas at Austin's OnRamps college readiness program.

References

[1] V. L. Almstrum, O. Hazzan, M. Guzdial, and M. Petre. Challenges to computer science education research. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, St. Louis, Missouri, USA, 2005. ACM.

[2] O. Astrachan, J. Cuny, C. Stephenson, and C. Wilson. The CS10K project: mobilizing the community to transform high school computing. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, page 85. ACM, 2011.

[3] B. Barron, D. Schwartz, N. Vye, A. Moore, A. Petrosino, L. Zech, The Cognition, and Technology Group at Vanderbilt. Doing with understanding: Lessons from research on problem- and project-based learning. *The Journal of the Learning Sciences*, 7(3 and 4):271–311, 1998.

[4] B. Beth and C. Lin. Thriving in Our Digital World. *CSTA Voice*, 11(3):2–3, July 2015.

[5] J. Black, J. Brodie, P. Curzon, C. Myketiak, P. W. McOwan, and L. R. Meagher. Making Computing

Interesting to School Students: Teachers' Perspectives. pages 255–260. Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, 2013.

[6] L. Brantley-Dias and P. A. Ertmer. Goldilocks and TPACK: Is the Construct "Just Right?". *Journal of Research on Technology in Education*, 46(2):103–128, 2013.

[7] A. Carbone, L. Mannila, and S. Fitzgerald. Computer science and IT teachers' conceptions of successful and unsuccessful teaching: A phenomenographic study. *Computer Science Education*, 17(4):275–299, 2007.

[8] L. ChanLin and K. Chan. PBL approach in web-based instruction. *Journal of Instructional Psychology*, 31:2, 2004.

[9] L. Cuban, H. Kirkpatrick, and C. Peck. High Access and Low Use of Technologies in High School Classrooms: Explaining an Apparent Paradox. *American Educational Research Journal*, 38(4):813–834, 2001.

[10] K. Deibel. Studying our inclusive practices: Course experiences of students with disabilities. In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITiCSE '07, pages 266–270, New York, NY, USA, 2007. ACM.

[11] R. Donnelly. Harmonizing technology with interaction in blended problem-based learning. *Computers and Education*, 54(2):350–359, 2010.

[12] J. Gal-Ezer and C. Stephenson. The current state of computer science in U.S. high schools: A report from two national surveys. *Journal for Computing Teachers*, Spring:1–5, 2009.

[13] B. Glaser and A. Strauss. *The discovery of grounded theory: Strategies for qualitative research.* Aldine Publishing Company, Chicago, 1967.

[14] M. Guzdial. Learning how to prepare computer science high school teachers. *Computer*, 44(10):95–97, 2011.

[15] J. Hughes, R. Thomas, and C. Scharber. Assessing Technology Integration: The RAT - Replacement, Amplification, and Transformation - Framework. *Paper presented at the Society for Information Technology and Teacher Education International Conference 2006, Orlando, Florida, USA*, 2006.

[16] S. Judge, K. Osman, and S. Yassin. Cultivating communication through PBL with ICT. *Procedia - Social and Behavioral Sciences*, 15:1546–1550, 2011.

[17] Y. Kolikant. Computer science education as a cultural encounter: a socio-cultural framework for articulating teaching difficulties. *Instructional Science*, 39(4):543–559, 2011.

[18] J. Krajcik and P. Blumenfeld. Project-Based Learning. In R. K. Sawyer, editor, *The Cambridge Handbook of the Learning Sciences*, pages 317–330. Cambridge University Press, Cambridge, UK, 2006.

[19] D. Kumar. Approaches to interactive video anchors in problem-based science learning. *Journal of Science Education and Technology*, 19(1):13–19, 2010.

[20] S. Merriam. What can you tell from an N of 1?: Issues of validity and reliability in qualitative research. *PAACE Journal of Lifelong Learning*, 4:51–60, 1995.

[21] S. Merriam. *Qualitative research in practice: Examples for discussion and analysis.* Jossey-Bass, San Francisco, CA, 2002.

[22] P. Mishra and M. Koehler. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6):1017–1054, 2006.

[23] L. Postner and R. Stevens. What resources do CS1 students use and how do they use them? *Computer Science Education*, 15(3):165–182, 2005.

[24] J. J. Randolph, G. Julnes, E. Sutinen, and S. Lehman. Computer science education research at the crossroads: a methodological review of computer science education research, 2000–2005. *Journal of Information Technology Education*, 7:135–162, 2007.

[25] S. Shuhidan, M. Hamilton, and D. D'Souza. Instructor perspectives of multiple-choice questions in summative assessment for novice programmers. *Computer Science Education*, 20(3):229–259, 2010.

[26] C. Simard, C. Stephenson, and D. Kosaraju. Addressing Core Equity Issues in K-12 Computer Science Education. *Report from The Anita Borg Institute, CSTA, and The University of Arizona*, 2010.

[27] R. M. Tamim, R. M. Bernard, E. Borokhovski, P. C. Abrami, and R. F. Schmid. What Forty Years of Research Says About the Impact of Technology on Learning: A Second-Order Meta-Analysis and Validation Study. *Review of Educational Research*, 81(1):4–28, 2011.

[28] G. Veletsianos, B. Beth, C. Lin, and G. Russell. Design Principles for Thriving in Our Digital World, a High School Computer Science Course. *Journal of Educational Computing Research*, in press.

[29] A. Walker, M. Recker, M. Robertshaw, J. Olsen, H. Leary, L. Ye, and L. Sellers. Integrating technology and problem-based learning: A mixed methods study of two teacher professional development designs. *The Interdisciplinary Journal of Problem-based-learning*, 5(2):70–94, 2011.

[30] A. Yadav and J. Korb. Learning to teach computer science. *Communications of the ACM*, 55(11):31, 2012.