Article

Design Principles for *Thriving in Our Digital World*: A High School Computer Science Course

Journal of Educational Computing Research
2016, Vol. 54(4) 443–461

DOI: 10.1177/0735633115625247
jec.sagepub.com

George Veletsianos[1], Bradley Beth[2], Calvin Lin[2], and Gregory Russell[3]

Abstract

Thriving in Our Digital World is a technology-enhanced dual enrollment course introducing high school students to computer science through project- and problem-based learning. This article describes the evolution of the course and five lessons learned during the design, development, implementation, and iteration of the course from its first through third year of implementation. The design principles that we describe have guided our design endeavors and may be helpful to instructional designers, learning technologists, and others who are engaged in the design and development of in situ interventions to improve the teaching and learning of computer science.

Keywords

computer science education, design principles, instructional design, design-based research, learning environments

The teaching of computer science (CS) in K-12 is experiencing rapid growth and interest, with worldwide initiatives to introduce its formal and informal

[1]School of Education and Technology, Royal Roads University, Victoria, BC, Canada
[2]Department of Computer Science, The University of Texas at Austin, TX, USA
[3]Oracle Inc., USA

Corresponding Author:
George Veletsianos, School of Education and Technology, Royal Roads University, Victoria, BC, Canada.
Email: veletsianos@gmail.com

instruction and dissemination, including efforts in France (Baron, Drot-Delange, Grandbastien, & Tort, 2014), New Zealand (Bell, Andreae, & Robins, 2014), and the United Kingdom (Brown, Sentance, Crick, & Humphreys, 2014). Improvements in CS education are often cited as potential solutions to low enrollments in CS across the United States (Gal-Ezer & Stephenson, 2009; Ryoo, Margolis, Lee, Sandoval, & Goode, 2013; Simard, Stephenson, & Kosaraju, 2010), especially among women and ethnic minorities (Girl Scouts Research Institute, 2012; Simard et al., 2010). Many efforts to disseminate CS resources and best practices have encountered limited success beyond local implementations (Simard et al., 2010), and CS teachers often find themselves without the materials, resources, or support necessary to improve learning, engagement, and equity in their classrooms. Significantly, Almstrum, Hazzan, Guzdial, and Petre (2005) note that "too much of the research in computing education ignores the hundreds of years of education, cognitive science, and learning sciences research that have gone before us" (pp. 191–192).

To address these issues, the U.S. National Science Foundation supports efforts to develop a knowledge base for computing education in K-12 and to broaden participation and education in computing, including collaborations with experts in other content areas (e.g., Gilbert, 2006). Many of these initiatives focus on the development of CS education resources and supports that will directly impact teaching and learning in CS classrooms. For example, educators from the Santa Fe Institute and the University of New Mexico have developed *New Mexico Computer Science for All* with an interdisciplinary approach to prepare high school STEM teachers to teach CS through computerized modeling and simulation (Astrachan, Osborne, Lee, Beth, & Gray, 2014). In partnership with the College Board, CS educators, researchers, and experts have developed an introductory course called *AP Computer Science: Principles* that aspires to "introduce students to the central ideas of computer science, to instill ideas and practices of computational thinking, and to have students engage in activities that show how computing changes the world" (The College Board, 2014). A parallel curriculum, *Exploring Computer Science*, focuses on engaging students with and training teachers to teach inquiry-based CS (Exploring Computer Science, 2015). Each of these initiatives attempts to improve CS education and access at the K-12 level by addressing pedagogical and instructional issues.

The U.S. National Science Foundation's funding of CS education has encouraged the computing education community to partner with education researchers. Unfortunately, computing education research suffers from a problem identified in education research, namely, that in situ interventions addressing educational problems may be best examined and addressed using design-based research (DBR) rather than predictive research (Reeves, 2006, 2011). Thus, we believe that detailed descriptions of the design decisions and principles used to develop and refine interventions will be helpful to computing education designers and researchers as they seek to develop new learning environments and to refine existing ones.

In this article, we describe the design, development, evolution, and design principles that were developed for a dual enrollment CS course called *Thriving in Our Digital World* (TODW), implemented in 5, 13, and 14 high schools in Years 1, 2, and 3, respectively. Our work lies at the intersection of learning design, learning technologies, and CS education. We first provide a general description of the course. Next, we describe the initial design efforts to create the course, the rapid development of the pilot initiative, and the lessons learned from the implementation and evolution of the initiative.

Thriving in Our Digital World

TODW is a dual enrollment course that introduces CS to high school students. The main features of the course are as follows:

- It is guided by *CS Principles* (The College Board, 2014, p. 2) which is a curricular framework describing “the content, practices, thinking, and skills central to the discipline of computer science.”
- It is informed by student-centered pedagogies, with emphasis on problem-based learning (PBL) and project-based learning.
- It is divided into modules, with each module focusing on a particular area and guided by a small number of problems or projects (e.g., in the *Artificial Intelligence* module, students programmed video game components in order to learn about the different ways to employ artificial intelligence strategies).
- It is structured so that students work in small groups to create tangible artifacts that address each problem.
- It is taught in a blended fashion, where teachers facilitate regularly scheduled classroom sessions in a face-to-face 1:1 computing environment and where all learning materials are located in an online learning environment that facilitates the use of digital resources, assignment submission, and management of the course at geographically dispersed schools.
- It is assessed via end-of-module artifacts and examinations that address module problems or projects.

The aim of the course is to improve the teaching of CS, to engage underrepresented populations in CS, and to increase the number of CS students and teachers at the high school level. The course used an open textbook, made use of open educational resources and media developed by course developers, and is available under a Creative Commons license. The course was developed by a team of learning designers, education researchers, and CS experts, and it was refined by the same team with feedback from K-12 teachers, students, and external evaluators. Now in its third year of offering, the course has been offered

Table 1. Demographic Comparison of Underrepresented Groups.[2]

	Thriving in Our Digital World (%)	AP computer science A	
		Texas (%)	United States (%)
Female	30.3	23.5	19.2
Black	6.7	3.6	3.9
Hispanic	32.9	19.2	8.3

at 23 high schools and to more than 500 students across the state of Texas. These students represent a diverse population not typically seen in college preparatory CS courses. For instance, 32.1% and 21.2% of students enrolled qualified for free or reduced lunches in 2012 and 2013, respectively. Table 1 illustrates how TODW enrollments of underrepresented groups compared with those of Advanced Placement CS.

Instructional Model and the Foundations of the Course

A backwards design methodology (Wiggins & McTighe, 2006) was used to outline the mindsets and competencies that the team wished the students to gain by the end of the course, and then worked backwards to design and develop draft versions of activities, assessments, course goals, course themes, course structure, and learning objectives. Early meetings focused on the development of module topics and learning objectives from the *CS Principles*, brainstorming of authentic problems and projects for each module, organization of course content within learning modules, and the selection of a learning management system to host course content and interactions.

The instructional model used was based on Krajcik and Blumenfeld's (2006) PBL model. The model worked as follows: Each module and overarching problem was launched with an anchor video that provided context and engaged students with the content (Kumar, 2010). This aspect of the project was heavily influenced by the work of the Cognition and Technology Group at Vanderbilt (1990, 1992), which developed and examined the use of anchor videos to enhance science instruction. Students then read and discussed the problem description as well as the rubric for evaluation. Next, students formed groups, completed group contracts, and began completing "Know, Want to Know, Learned" charts that helped identify preconceptions and knowledge gaps. From there on, students collaborated in inquiry learning activities. Teachers presented content, facilitated classroom discussions, lead tutorials, and provided students with feedback. Groups were explicitly provided with opportunities to work on their projects during "work days" and to evaluate and reflect upon their and their peers' work (e.g., rubric checks). At the conclusion of each module, student

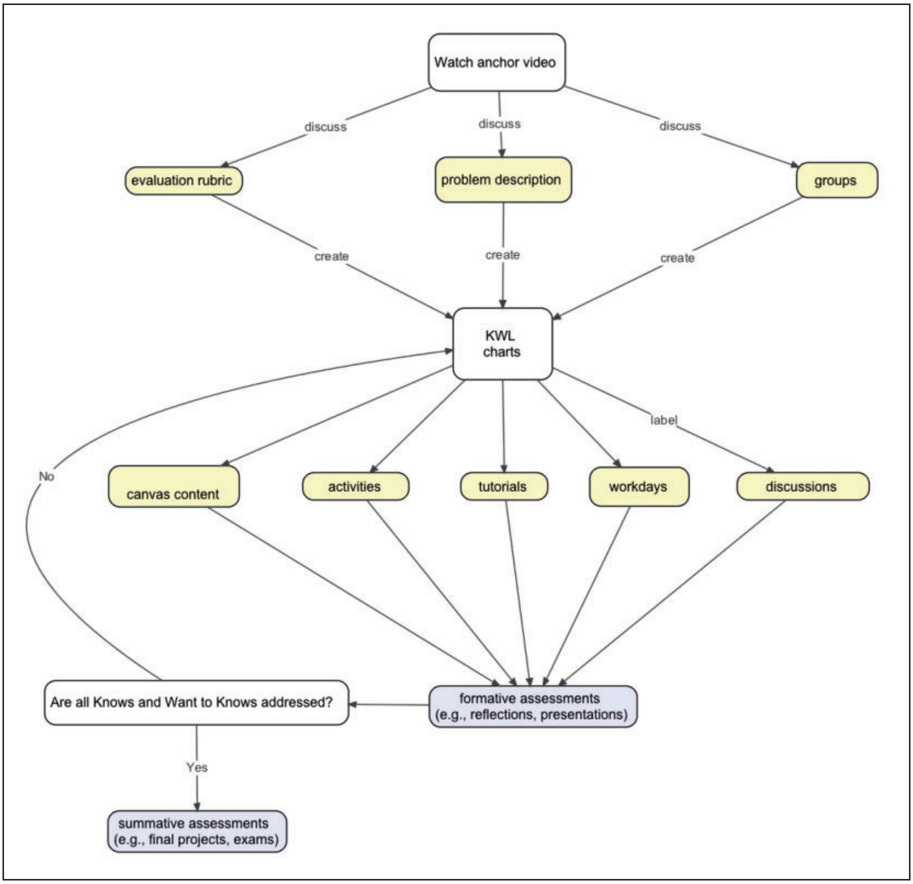

Figure 1. A graphical overview of the instructional model used in *Thriving in Our Digital World.*

learning was informally assessed via presentations and reflections and formally assessed via final projects and summative examinations. Thus, all modules included anchor videos, Know, Want to Know, Learned charts, peer feedback activities for final projects, and collaboration evaluations. Group presentations and reflections, however, changed from module to module. Figure 1 provides a graphical overview of this model.

Since problems and projects were going to be central in TODW, one major aspect of the project entailed consulting the relevant literature and devising characteristics for the problems and projects presented to learners. It was decided that each problem should

- be open-ended and inquiry-based,

- represent authentic tasks performed by professional computer scientists,
- require the authentic application of CS knowledge and skills,
- address higher order thinking skills,
- necessitate student collaboration,
- utilize PBL-specific scaffolds, and
- integrate multiple learning technologies.

Projects and problems that would challenge but appeal to high school students were sought. Table 2 outlines the course modules at the time of writing. The modules are organized in a logical order of increasing complexity. Individual learning objectives are in subtopics within each module. For example, the learning objective, "students will be able to extract structured information from unstructured data," is organized into the subtopic *Extraction* within the *Big Data* module.

Each module, with the exception of *Programming*, was centered on a large problem or project in which students had to work in small groups to devise a solution. In their groups, students engaged in a number of learning activities specified in the Canvas learning management system. As can be seen from Figure 2 which shows the front page of one of the course modules, the students were guided through content in the form of PowerPoint lectures, reading material, and videos as well as assignments and homework, most of which were completed within the Canvas site. The supporting activities were intended to develop students' knowledge pertaining to CS (cf. Anderson, 1982).

Iterative Design

Although we used backwards design to match outcomes to activities, we used an iterative rapid prototyping approach to design and develop curricular materials due to (a) the large scope of the course curriculum, (b) the small size of our team, (c) a limited time frame, (d) limited available learning materials, and (e) the paucity of research on CS education (Guzdial, 2011). A rapid prototyping approach is flexible and allows ample opportunities for feedback (Tripp & Bichelmeyer, 1990). Phases of design, implementation, evaluation, and revision informed and overlapped each other, allowing us to continuously adapt our design to address unforeseen challenges.

Early in this process, we conducted pilot implementations with two teachers and formative evaluations with six teachers. In the software development field, the terms *alpha* and *beta* are used to refer to the earliest forms of developed products. These product designations signify that the products are still imperfect. *Alpha* designates a very rough, untested, and buggy product, and *beta* designates an improved and mostly functional product, in which bugs are still expected, though far less in number. Our pilot implementation consisted of an alpha product: Although we completed a basic framework for the entire course

Table 2. TODW Modules, Descriptions, and Tasks.

Module	Description	Projects, problems, and tasks	Examples of computer science concepts addressed
Impact	A crime has been committed at Martindale High School, and an innocent student has been wrongly accused. A senior was a victim of identity theft, and his online identity was been used to cyberbully a first-year student.	Practice digital forensics in order to exonerate a student falsely accused of cyberbullying.	Students will able to compare and contrast modern technologies with those before the Digital Revolution.
Programming	Through programming, artists create music and visual art, scientists create models of possible worlds, engineers build new products, medical researchers design and test possible cures, and businesses create jobs and wealth.	Develop a unique program of your own design using fundamental commands and the Scratch programming language.	Student will be able to simulate decision-making through the use of condition-controlled loop blocks.
Representation	A startup video game company is making their first game, a two-dimensional side-scroller, and they want a new controller for the game.	Program the role of each button press of a virtual controller and map them to specific binary representations.	Students will be able to convert between decimal and binary representation of numbers.
Digital manipulation	All digital media consists of bits. Computer scientists manipulate bits to achieve a wide variety of outcomes. Image editors can	Create a digital image filter that can complete postprocessing effects algorithmically.	Students will be able to explain the differences between lossy and lossless compression algorithms.

(continued)

Table 2. Continued

Module	Description	Projects, problems, and tasks	Examples of computer science concepts addressed
	transform many characteristics of images with ease. Productivity software can perform complex mathematical functions automatically. Video game environments can be rendered dynamically.		
Big Data	Big data is all around us, and every day, big data analysis leads to innovative ideas, applications, and knowledge. Many of these innovations and discoveries are shared via TED talk presentations intended to "change the world."	Complete extensive research on a topic/theme of your choice, conduct in-depth data analysis on big data sets, and present the results to a live audience.	Students will be able to model the creation of structured data from unstructured datasets.
Artificial intelligence: Turing test	Artificially intelligent chatterbots mimic human conversations in startling ways, but can they actually pass for human?	Design and administer a Turing Test that can distinguish between an artificially intelligent chatterbot and a human.	Students will be able to analyze a "chatterbot" for pattern recognition and manipulation.
Artificial intelligence: Maze game	In a single player video game, your opponent is the computer. The computer relies on artificial intelligence to try and beat players in a variety of ways. Different types of artificial intelligence lead to different types of games.	Program opponents for a video game that use a variety of intelligent and unintelligent behaviors	Students will be able to model artificial intelligence "planning" as a form of utility maximization.

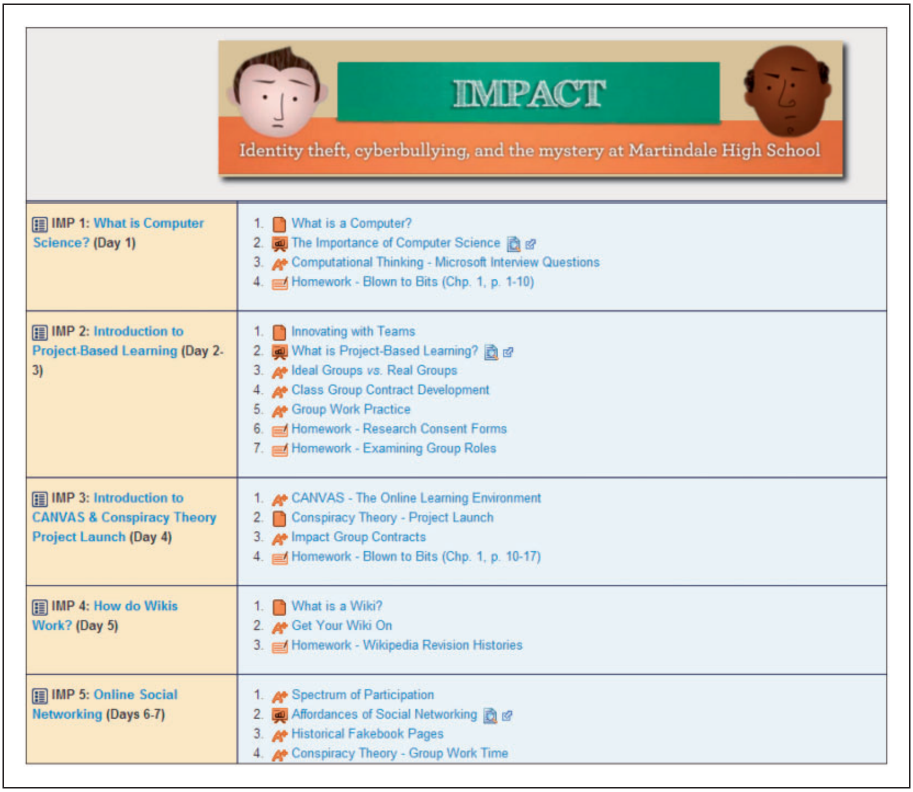

Figure 2. Part of sequence of impact module as present on Canvas.

prior to the pilot implementation, only two complete modules were distributed to teachers. Our formative evaluations consisted of a beta product. Our iterative design and development process allowed us to observe how these early modules were implemented and enabled us to refine our instructional approaches.

The methodology used to generate design principles and instructional approaches was DBR. DBR is a methodology intended to enhance educational interventions through "iterative analysis, design, development, and implementation, based on collaboration among researchers and practitioners in real-world settings" (Wang & Hannafin, 2005, p. 6). Instructional designers have used DBR in order to enhance learning in real-world contexts (Sandoval, 2004). The data informing the generation of the principles described below include analysis of a broad array of ethnographic observations, formative curriculum evaluation, teacher interviews, student focus groups, and researcher reflections on teachers' integration efforts.

In the paragraphs that follow, we describe the design decisions we made as we transitioned from the alpha (Year 1) to the beta (Year 2) to the current (Year 3) version of the course. Based on these revisions, teacher and student reception of modules drastically improved: In Year 1, about 8% of the students enrolled earned college credit, and in Year 2, about 55% of the students enrolled earned college credit. We estimate that between 75% and 80% of students will earn college credit in Year 3. Each school awarded its own course grades for internal purposes by evaluating student projects. To receive college credit, students took an examination that was aimed at evaluating the learning objectives of the course described earlier. The examination was scored by independent third parties. While it is difficult to attribute a particular portion of the stark increase in success in the college credit examination to the design decisions and revisions described later, based on evidence that we have collected, we can say definitively that the revisions described below contributed to student success either directly or indirectly.

Design Decisions and Evolution

In this section, we describe revisions to TODW that improved the quality of the project as a whole and alleviated a number of contextual challenges of classroom implementation. For each revision, we state a design principle that we followed in future iterations of the course. While we made wide-ranging revisions to our instructional products over the years (e.g., materials were revised to increase the logical flow of instruction, reduce redundancy, and increase rigor while maintaining relevancy), here we report only those design decisions and principles that we believe are novel for learning designers to consider—and especially novel to individuals designing for scale and for CS curricula.

Build, Evaluate, Improve, and Repeat

One of the guiding principles of this project is inherent in the rapid prototyping model that we used to guide our design. Specifically, we sought to *rapidly build designs and media, evaluate them with real audiences, improve them, and repeat the process*. For instance, the original videos created to launch each problem were developed using off-the-shelf software. Such software enabled us to focus on the storyline of the problem posed to the students without devoting extraneous time and resources to the media. We chose to focus on the storyline so as to bring to life the problem that we wanted learners to engage with. For instance, the first iteration of the *Impact* problem mentioned earlier was launched with a video using GoAnimate, a free tool to create animated videos (Figure 3).

This video is representative of the media and off-the-shelf software used in the first iteration of the course to launch all of the problems and projects that were

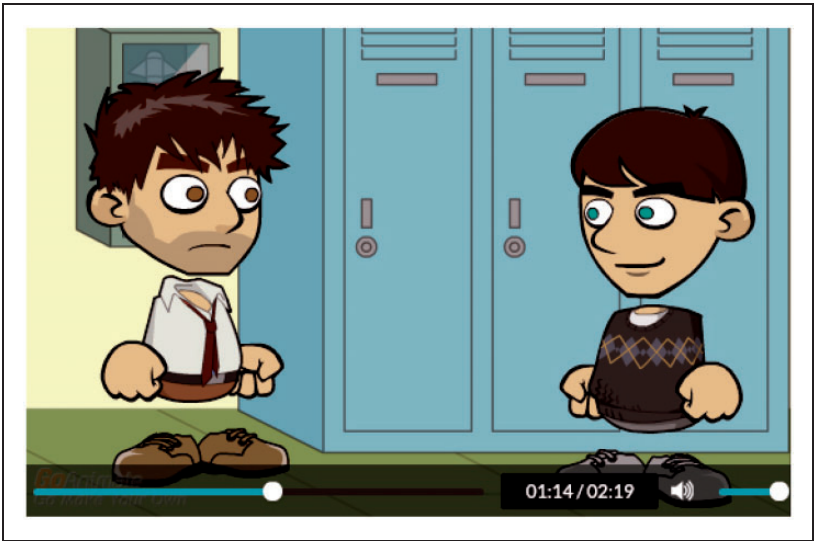

Figure 3. A screenshot of one of the GoAnimate videos used in the first iteration of *Thriving in Our Digital World.*

provided to students. We noted two deficiencies with these videos: First, the students had to watch the whole video to understand what they had to do. Second, we felt limited by the standardized media and lack of control. We believed that we could improve the quality of these videos by making the digital characters more likable and animated, directing learner attention as needed, and reducing extraneous information embedded in the media. To address these issues, we created introductory paragraphs to provide background information to students prior to watching the video and worked with a professional video development group to redesign our videos. To illustrate, the following is the introductory text and a screenshot (Figure 4) of the revised video[1] we used in the second iteration of the *Impact* module:

> A crime has been committed at Martindale High School, and an innocent man has been wrongly accused. A senior named Leandro was a victim of identity theft, and his online identity has been abused to cyberbully a freshman named Chris. As a result, Leandro has been expelled from school. As Leandro's friend, you must help convince the principal that Leandro is innocent! In your group examine, analyze, and organize all of the digital evidence that can exonerate Leandro. You're trying to convince the Martindale High School principal that it is highly unlikely that Leandro committed the acts accused of him, so build as strong case to that end as you can. Your job is to create a compelling presentation in order to exonerate Leandro.

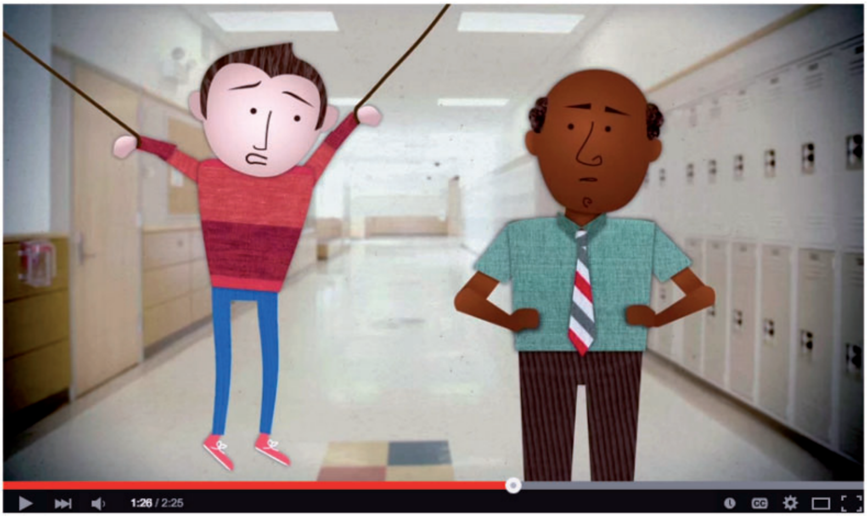

Figure 4. A screenshot of the revised video.

Design for Replication and Flexibility to Address Scale and Problems of Variation

An explicit goal of the project was to design a course that could be scaled and replicated across multiple schools. Flexibility can allow teachers to adapt the curriculum to meet context-specific challenges and needs. Scaling this project required us to solve problems of variation (Dede, Honan, & Peters, 2005), which refers to differences between implementation sites and local issues that determine whether an innovation that is introduced will be successful or unsuccessful. Such variation is inherent in the K-12 ecosystem. For instance, pilot schools each maintained different bell schedules that required adaptations to instructional activities to fit into predetermined time limits. Thus, before we could scale our course, we had to design for flexibility, which meant providing multiple homework assignments, offering resources in a variety of media, developing alternate activities to accommodate more knowledgeable students, and providing opportunities for remediation. Thus, the design principle that we developed to guide future activities was the following: *Seek flexible designs in order to address scale and variation*. To illustrate how we designed for replication and flexibility, we describe below changes in our design documents shared with teachers.

Sharing instructional designs with teachers is a valuable practice because it enables designers to communicate instructions and intentions to teachers. Without instructional designs (e.g., lesson descriptions, assessment instruments, etc.), teachers may miss critical instructional opportunities, especially considering the complexities inherent in blended and inquiry learning environments. Designs are only valuable if they influence teaching and learning, so teachers must consider them both informative and accessible if they are going to use them. Thus, we made instructional designs easier to locate on the online environment, provided thorough instructions, drew cleared connections between learning objectives and activities, and, to accommodate both slow-moving and fast-moving schools, provided more flexible access to curricular materials in the online environment (e.g., on-demand access on an individual basis).

Design a PBL-Based Curriculum but Not a PBL-Only Curriculum

Our early efforts to design PBL modules for TODW placed as many learning objectives into the overarching module project as possible. As teachers and students expressed frustration at learning objectives that had little authenticity, we observed that frustrations and dissatisfaction arose more frequently in association with objectives that were included in module projects that were tangentially related to the main project. For instance, in an early module, students were asked to develop image and audio advertisements for a company of their choosing in order to learn about digital representation and manipulation. To cover learning objectives pertinent to digital representation and manipulation, we expanded the project to include a student–employer interview during which students were asked to respond to a variety of prompts. Although the interview task itself was authentic, the questions that were included were not authentic in the context of a job interview (e.g., what digital information can and cannot be compressed?). The interview portion of the project seemed forced and became confusing. The revised project instead asked students to program a virtual video game controller, mapping the physical button pushes to binary representations that trigger a digital character's actions. Learning objectives that needed to be addressed in this module but that did not fit in this project were addressed using other approaches.

The PBL models used in the pilot curriculum was more prominent and rigid than in revised curricula that utilized more inquiry-based and teacher-centered pedagogies. Originally, teachers found the model drastically different from their own practices and cumbersome in its requirements. The rigidity of the model was off-putting to teachers, as they were being asked to trust the model and its potential outcomes, when their classroom experience was telling them otherwise. With their feedback, we decreased the demands of the model and sought to include pedagogical approaches that aligned with a socioconstructive ethos of learning but were not PBL-specific. Thus, we transitioned to a PBL-based curriculum as opposed to a PBL-only curriculum. In this way, we improved the flexibility of our approach, freed our designs from encompassing an endless array of learning outcomes, and supported teachers in making better sense and finding greater value in PBL approaches to the course. Overarching projects focused on core learning objectives, including those addressing higher order thinking skills (e.g., analysis and evaluation) that were deemed to be critical to course goals. Foundational knowledge and lower level skills were addressed with direct instruction and other student-centered alternatives.

This change was pragmatic and in accordance with our perspective that learning design is inherently a design discipline. Rather than blindly adhering to philosophical beliefs irrespective of outcomes, we hoped that this change would lead to more effective and engaging practices. While we discussed whether this change meant that the pedagogical practices were ineffective, we realized that this was a normal part of the process of introducing new practices into complex systems. Teachers enter the classroom with prior experiences, beliefs, and practices, often supported with various measures of success. Acknowledging and valuing teachers' skills and knowledge in the administration of a curricular model may promote buy in and lead to higher quality implementations.

Thus, we arrived at a third design principle: *Relying exclusively on one instructional approach to teach all of the content and skills necessary may be ineffective; instead, identify learning objectives to address via an overarching project/problem and address the rest via other approaches*. By relying solely on PBL, we were discounting other pedagogical approaches that were often more appropriate to the subject matter—practices that often tended to make learning more effective and efficient.

Use Both Student-Centered and Teacher-Centered Pedagogies

Within inquiry-based curricula, there are instances in which teacher-centered approaches to instruction, such as demonstration and lecture, are practical and effective (Bransford & Schwartz, 1998). We used inquiry-learning approaches to target higher order thinking skills like analysis, evaluation, and synthesis, and teacher-centered approaches for lower order thinking skills like knowledge and comprehension. However, during our pilot implementation, we discovered that some of the PBL activities that addressed declarative knowledge consumed an overwhelming amount of classroom time. For instance, to apply binary counting to a real-world problem (i.e., too few candles for numerals), the *Representation* module included an assignment that asked students to write an algorithm for lighting a birthday cake using only binary numbers. The task included a project launch, narrative, anticipatory set, graphic organizers, discussion, peer feedback, and evaluation, and teachers and students spend an inordinate amount of time examining "basic" concepts in overly complex ways. The difficulty of the module's objectives did not seem to warrant the time allotted for all of these activities. As a result, we adopted the following guiding principle: *While inquiry-based activities may address all of our objectives,* *address lower-order thinking and practical skills using demonstration, direct instruction, and lecture, as these approaches may be more effective and efficient.*

As a result, the course makes selective use of a diverse range of pedagogical approaches. Complementing inquiry-based curricula with teacher-centered approaches appears to be especially pragmatic for classroom implementations, as we strive to provide valuable opportunities for students to experiment, discover, and construct knowledge in meaningful ways. By using teacher-centered approaches for lower order thinking and practical skills, we allow students and teachers to spend more time on exploration and experimentation.

Provide Teachers With Experiences With Content and Pedagogies and Establish Common Expectations

All individuals teaching this course for the first time were veteran CS teachers who were relatively unfamiliar with this course's curriculum and pedagogy. For instance, some lacked prior experience and knowledge of with social networking sites and wikis, major components of the *Innovations* module. Other modules, such as *Big Data, Artificial Intelligence*, and *Security*, were daunting to teachers who were unfamiliar with these concepts. Individuals who did not participate in a summer professional development program offered prior to the teaching of the course encountered even more challenges. Additionally, all teachers felt anxious about using new pedagogies in their classroom. As the school year began, some were not comfortable with open-ended discussions, some were concerned about cooperative learning and the idea of dividing work among group members, and most were worried about classroom management. In summary, teachers were expected to teach new content and skills and to do so using pedagogies that challenged and made them uncomfortable.

All this resulted in increased demands on teacher preparation time. Teachers were expected to implement an unfamiliar curriculum and practice unfamiliar pedagogies while using an unfamiliar learning management system that housed all curricula. Teachers spent exorbitant amounts of time reviewing lesson plans and content pages, familiarizing themselves and experimenting with the learning management system, and customizing the experience for their classroom. With the exception of one site, each teacher was the lone CS teacher at their school and the only teacher of this course. Although two teachers consistently interacted with course designers and researchers, others did not, and one individual mentioned not wanting to "bother" the designers and researchers.

In the absence of adequate support at their school and preparation to use inquiry-based pedagogies in a blended learning format, the impacts of well-designed curricula and activities may be minimal. Thus, the final principle that we devised to guide future implementation was the following: *Substantial support should be provided to teachers to learn the content and pedagogies and overcome the obstacles they are facing in their day-to-day practice*. Teacher learning experiences should be accompanied with feedback, support, and adequate time for practice. In our context, a face-to-face summer institute to train teachers accompanied with on-demand support throughout the year worked well. Learning new content and how to teach it in a new manner within an unfamiliar online environment is onerous. In enacting this principle in Year 2 of the project, teachers noted that experiences practicing the new pedagogies while teaching the new content were valuable. For designers and developers, providing teacher support may be time-consuming difficult, but it is critical for the successful adoption, adaptation, and implementation of a project. Consideration for how users will be prepared to enact curricula should not be overlooked.

Conclusion

In this article, we have described TODW, its foundations and its instructional model. Next, we explained the design decisions that we made between multiple iterations of the course. These were as follows:

- Design iteratively and rapidly, rapidly build designs and media, evaluate them with real audiences, improve them, and then repeat the process.
- To address scale and variation, challenge designs specifically with respect to flexibility.
- Be willing to deviate from the overarching pedagogical approach for some of the content and skills that need to be addressed, as relying exclusively on one instructional approach to teach all of the content and skills necessary may be ineffective.
- While inquiry-based activities may address all of our objectives, address lower order thinking and practical skills using demonstration, direct instruction, and lecture, as these approaches may be more effective and efficient.
- Provide substantial support to teachers to learn the content and pedagogies and overcome the obstacles they are facing in their day-to-day practice.

As we scale this intervention to more and more high schools, we find ourselves encountering fewer and fewer challenges. We anticipate that the curricular materials will need to be updated to maintain relevance, and as more data become available on the effectiveness of our implementations, we expect to further refine our materials and approaches. We recommend that teams involved in the design of learning materials and technology-enhanced approaches for CS consider how these principles might support their work. As socioconstructive approaches to education are gaining prominence in CS, these findings may be worthwhile to consider.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This article was provided by the National Science Foundation under awards CNS-1138506 and DRL-1441009 and by the State of Texas under the OnRamps Program, coordinated by The University of Texas at Austin. Veletsianos also acknowledges funding received by the Canada Research Chairs program.

Notes

1. The video can be seen at https://www.youtube.com/watch?v=5g__WGXJdc
2. Data for Advanced Placement participation are drawn from Ericson (2014). The demographics of the corresponding populations within the State of Texas are roughly 11.32% Black and 36.71% Hispanic.

References

Almstrum, V., Hazzan, O., Guzdial, M., & Petre, M. (2005). Challenges to computer science education research. *Proceedings From SIGCSE '05: The 36th SIGCSE Technical Symposium on Computer Science Education*. St. Louis, MO: Association for Computing Machinery.

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological review*, *89*, 369–406.

Astrachan, O., Osborne, R. B., Lee, I., Beth, B., & Gray, J. (2014, March). Diverse learners, diverse courses, diverse projects: Learning from challenges in new directions. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 177–178). New York, NY: ACM.

Baron, G. -L., Drot-Delange, B., Grandbastien, M., & Tort, F. (2014). Computer science education in French secondary schools: Historical and didactical perspectives. *Transactions on Computing Education*, *14*(2), 11, 1–11, 27.

Bell, T., Andreae, P., & Robins, A. (2014). A case study of the introduction of computer science in NZ schools. *Transactions on Computing Education*, *14*(2), 1–10.

Bransford, J., & Schwartz, D. (1998). A time for telling. *Cognition and Instruction*, *16*(4), 475–522.

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *Transactions on Computing Education*, *14*(2), 1–9.

Cognition and Technology Group at Vanderbilt. (1990). Anchored instruction and its relationship to situated cognition. *Educational Researcher*, *19*(6), 2–10.

Cognition and Technology Group at Vanderbilt. (1992). The Jasper experiment: An exploration of issues in learning and instructional design. *Educational Technology, Research and Development*, *40*(1), 65–80.

Dede, C., Honan, J. P., & Peters, L. (2005). *Scaling up success: Lessons learned from technology-based educational improvement*. San Francisco, CA: Jossey-Bass.

Ericson, B. (2014). *Detailed data on pass rates, race, and gender for 2013*. Retrieved from http://home.cc.gatech.edu/ice-gt/556

Exploring Computer Science. (2015). *Exploring computer science: Our mission*. Retrieved from http://www.exploringcs.org/about/mission

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

Gilbert, J. E. (2006). Making a case for BPC [broadening participation in computing]. *Computer*, *39*(3), 83–86.

Girl Scouts Research Institute. (2012). *Generation STEM: What girls say about science, technology, engineering, and math*. New York, NY: Girl Scouts of the USA.

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

Krajcik, J. S., & Blumenfeld, P. C. (2006). Project-based learning. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 317–330). Cambridge, England: Cambridge University Press.

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

Reeves, T. (2006). Design research from the technology perspective. In J. V. Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Eds.), *Educational design research* (pp. 86–109). London, England: Routledge.

Reeves, T. (2011). Can educational research be both rigorous and relevant? The pendulum swings: Rigor and relevance. *Educational Designer*, *1*(4), 1–24.

Ryoo, J., Margolis, J., Lee, C., Sandoval, C., & Goode, J. (2013). Democratizing computer science knowledge: Transforming the face of computer science through public high school education. *Learning, Media & Technology*, *38*(2), 161–181.

Sandoval, W. (2004). Developing learning theory by refining conjectures embodied in educational designs. *Educational Psychologist*, *39*(4), 213–223.

Simard, C., Stephenson, C., & Kosaraju, D. (2010). *Addressing core equity issues in K-12 computer science education: Identifying barriers and sharing strategies*. Paulo Alto, CA: Anita Borg Institute for Women and Technology.

The College Board. (2010). *AP computer science: Principles – learning objectives and evidence statements*. New York, NY: The College Board.

The College Board. (2014). *CS principles framework*. Retrieved from http://www.csprinciples.org/home/about-the-project/docs/CSPrinciples-Framework-March-2014-DRAFT.pdf?attredirects=0&d=1

Tripp, S., & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design strategy. *Educational Technology Research and Development*, *38*(1), 31–44.

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, *53*(4), 5–23.

Wiggins, G., & McTighe, J. (2006). *Understanding by design*. Upper Saddle River, NJ: Pearson Education.

Author Biographies

Dr. George Veletsianos is a Canada research chair of Innovative Learning and Technology and an associate professor at the Royal Roads University. His research focuses on the practices and experiences of learners, educators, and scholars in emerging digital environments such as online social networks.

Bradley Beth is interested in the intersection of computer science and education. His research interests focus on computer science pedagogical content knowledge, intelligent tutoring systems, and computer-supported collaborative learning.

Dr. Calvin Lin is a professor of computer science. His research interests focus on compilers, parallel computing, and microarchitecture.

Gregory S. Russell is a senior curriculum developer for Oracle, Inc. specializing in the development of self-paced, interactive web-based trainings. Prior to joining Oracle, he worked as an instructional designer, researcher, and trainer for project engage at the University of Texas at Austin and as a middle school teacher in downtown Los Angeles.