A Design-Based Approach to a Classroom-Centered OELE

Nicole Hutchins(✉), Gautam Biswas, Miklos Maroti, Akos Ledezci,
and Brian Broll

Vanderbilt University, Nashville, TN 37212, USA
nicole.m.hutchins@vanderbilt.edu

Abstract. The sustainable, synergistic integration of computational thinking (CT) and STEM learning environments into K12 classrooms requires consideration of learner-centered and classroom-centered design. In other words, not only do we have to take into account the learning goals and capabilities of students, but also the technological capabilities of the classroom environment and the combined impact of the teacher and technology on the classroom dynamics, curriculum, and progress. This paper discusses the design and development of an open ended learning environment aimed at high school physics curriculum taught within a CT-based framework. We conclude with preliminary results from a semester-long implementation study in a high school physics classroom.

Keywords: Computational thinking · STEM · Blended learning environments

1 Introduction

The sustainable integration of innovative, open ended learning environments (OELEs) in K12 classrooms requires the aggregation of learner-centered [8] and classroom-centered design approaches. This approach takes into consideration the prior knowledge and capabilities of the student and scaffolds their individual learning processes. It also takes into account the logistics and environment of the classroom, and how to achieve a cohesive balance between the role of the teacher and the technology. OELEs provide students meaningful learning opportunities by adopting a pedagogy that enables students to acquire domain information, construct, test, and revise solutions to authentic, problem solving tasks [3, 4, 7, 9]. However, in previous work, OELE frameworks have primarily focused on learner-centered design, with little consideration of the role of the teacher in helping to orchestrate classroom dynamics, progression through the curriculum, and student engagement to support learning. While technological advancements have introduced adaptive tools in K-12 classrooms that provide personalized learning opportunities, assimilation as part of a standard classroom environment has remained elusive.

This paper outlines the design and development of a classroom-centered OELE aimed at integrating computational modeling into a high school physics classroom. To do so, we will describe the classroom and student-learning components relevant to our architecture, provide an overview of our system design, and conclude with results,

C. Penstein Rosé et al. (Eds.): AIED 2018, LNAI 10948, pp. 155–159, 2018.
https://doi.org/10.1007/978-3-319-93846-2_28

analysis, and necessary system modifications derived from lessons learned in our semester-long classroom study.

2 C2STEM

The Collaborative, Computational STEM (C2STEM) environment utilizes a novel paradigm that combines visual programming [6] with domain specific modeling languages (DSMLs) [2] to promote learning of physics and CT concepts and practices.

2.1 Classroom Structure

Like a typical classroom curriculum, C2STEM tasks are made up of easily identifiable Instructional, Model Building, Assessment, and Challenge tasks. *Instructional tasks* help students focus on learning and applying primary physics concepts, often one at a time, to prepare students for building computational models. This helps address students' lack of knowledge in physics and programming. Instructional tasks build on previous content to make it easier for students to learn in small chunks.

Following instructional tasks, students work on *model building tasks* that require them to combine the information gained from *instructional tasks* along with CT practices to build a correct computational model of a Physics phenomena. A third component, *formative assessments* or *assessment tasks*, assess student learning in Physics and CT with multiple choice and short answer questions and small model building exercises. Finally, *challenge tasks* require students to solve more difficult problems that test their abilities to put together concepts and practices emphasized in the module. For instance, in the one-dimensional kinematics motion challenge, students are required to program a medical delivery truck that completes a straight path trip whilst adhering to various speed limits on the path and then safely stops at a STOP sign, before continuing on to its final destination.

2.2 Modeling

Modeling tasks are broken down into conceptual and computational modeling tasks. These two connected activities “support modeling at different levels of abstraction and operationalize the important CT concepts of abstraction and decomposition” [1].

Conceptual Modeling. Conceptual modeling allows for systematic planning of the objects and their associated properties and behaviors needed to build a correct computational model for the assigned problem. This model building activity is completed in the “Plan” tool of the C2STEM environment. When conceptual model elements are selected, their relevant block(s) appear in the “Build” component of the learning environment. Similarly, if objects or behaviors are removed from the conceptual model, the associated blocks are removed in the Build component. Students are allowed to move between the two model-building representations as they build their simulations.

Computational. Computational modeling is implemented by embedding the SNAP! programming environment in C2STEM. As mentioned in our primary research objective, we utilize a physics-based DSML. A DSML is “a programming language or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain” [10]. It allows students to express solutions in the physics domain, and it provides students with precise, self-documenting code [5] supporting the application of relevant CT practices and constructs.

Our computational modeling environment also has a graphing module and a table generator that generates a spreadsheet of data values for selected variables. These tools help students interpret their simulation behaviors and debug their models.

2.3 Architecture

The C2STEM system uses a modular architecture that allows for seamless integration of its components for classroom instruction. A simple infrastructure was essential to accomplish sustainability in the classroom, and to minimize common logistical issues regarding installation and software updates on a large number of school-owned computers. The system is web-based and runs off a cloud server thereby allowing student access at school and at home. Figure 1 illustrates the overall architecture of our environment, including the ability to handle multiple clients simultaneously. This architecture provides researchers with the ability to grade student work in real-time through a logging functionality not described in this paper, as well as collaboration opportunities (e.g. students sharing computational workspaces).

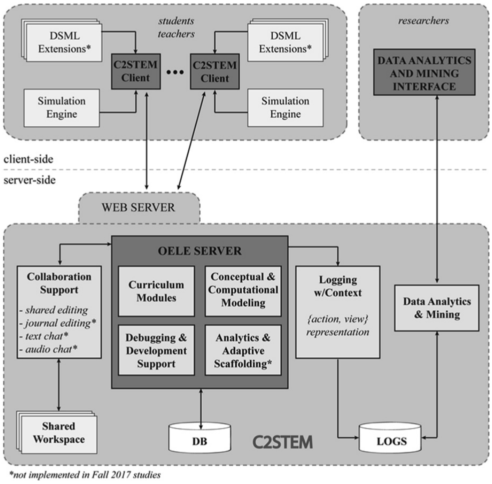

Fig. 1. System architecture

While our architecture supports our goal of a sustainable learning environment, the components of our system allow for scalability and adaptability by the teacher, instructional scaffolding, and quick grading of student assignments. Users have continuous access to all instructional tasks (Sect. 2.1) and resources, supporting the open-ended nature required of OELEs [5]. All tasks are composed of HTML components (task descriptions, multiple choice and short answer questions, etc.), the conceptual modeling component, and the computational modeling component.

3 Classroom Implementation

Following a preliminary usability study, our team conducted a semester-long study in a Physics high school classroom in Nashville, Tennessee. The study included 174 students taking an Honors Physics course; 84 students participated in the experimental group and used our C2STEM environment, and 90 students were in the control group – they did not use the system. Students completed four physics modules: three in Kinematics: 1D motion (with acceleration), 2D motion with constant velocity, and 2D motion with gravitational forces, and an introductory unit on 1D Force.

Our experimental group utilized the C2STEM environment an average of three out of four classroom sessions a week, with non-system days dedicated to content lectures by the physics teacher (delivered to both control and experimental groups). Participants completed pre- and post- tests in Kinematics, Mechanics, and CT.

For our preliminary results for the pre-posttests we randomly selected 30 experimental group participants. In the Kinematics pre-posttests, the experimental group's average score on the pre-test was a 22.4(5.4) and increased to an average score of 34.2 (3.4) in the post-test. For CT, the average score on the pre-test was a 22.4(4.7) on the pre-test, improving to 30.7(4.2) on the post-test. These preliminary results demonstrate that the experimental group was able significantly improve in their learning of physics content and received the additional benefit of significantly increasing their CT knowledge in the process.

4 Conclusion and Future Work

Our research goal for this phase of our design-based research approach was to develop and implement a classroom-centered OELE focused on the synergistic learning of CT and STEM. Results showed significant learning gains in Physics and CT. The Physics instructor combined classroom lectures and lab work with the use of C2STEM, and used the assessments on the system to improve his instruction and to help students learn their Physics concepts better. The pre-post test results indicate that the additional effort also led to better learning. Furthermore, an added bonus of using the system is that it taught students both Physics and CT concepts and practices. However, it is clear that to build this system up to its full learner-centered and classroom-centered potential, a number of additional tools will have to be developed for adaptive scaffolding of student learning, and also to aid the teacher in assessing students and keeping track of students' progress.

Acknowledgments. We thank Luke Conlin, Kevin McElhaney, Shuchi Grover, Satabdi Basu, Kristen P. Blair, Doris Chin, and our other collaborators from Vanderbilt University and Stanford University for their numerous contributions. This research is supported by NSF grant #1640199.

References

1. Basu, S., Biswas, G., Kinnebrew, J.S.: Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. User Model. User-Adap. Inter. **27**(1), 5–53 (2017)
2. Basu, S., Biswas, G., Kinnebrew, J.S.: Using multiple representations to simultaneously learn computational thinking and middle school science. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 3705–3711. MIT Press, Boston (2016)
3. Biswas, G., Segedy, J.R., Bunchongchit, K.: From design to implementation to practice - a learning by teaching system: Betty's brain. Int. J. Artif. Intell. Educ. **26**(1), 350–364 (2016)
4. Hannafin, M.J., Hill, J.R., Land, S.M., Lee, E.: Student-centered, open learning environments: research, theory, and practice. In: Spector, J., Merrill, M., Elen, J., Bishop, M. (eds.) Handbook of Research on Educational Communications and Technology. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-3185-5_51
5. Hasan, A., Biswas, G.: Domain specific modeling language design to support synergistic learning of STEM and computational thinking. In: Proceedings of the International Conference on Computational Thinking Education (2017)
6. Kelleher, C., Pausch, R.: Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. ACM Comput. Surv. **37**(2), 83–137 (2005)
7. Land, S.: Cognitive requirements for learning with open-ended learning environments. Educ. Tech. Res. Dev. **48**(3), 61–78 (2000)
8. Quintana, C., Krajcik, J., Soloway, E.: Exploring a structured definition for learner-centered design. In: 4th International Conference of the Learning Sciences, Erlbaum, Mahwah, NJ, pp. 264–265 (2000)
9. Sengupta, P., Kinnebrew, J.S., Basu, S., Biswas, G., Clark, D.: Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. Educ. Inf. Technol. **18**(2), 351–380 (2013)
10. Van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: an annotated bibliography. SIGPLAN Not. **35**(6), 26–36 (2000)