A technology-mediated co-design approach for integrating Computational Thinking in a science classroom

Sugat Dabholkar, Michael Horn, and Uri Wilensky
sugat@u.northwestern.edu, michael-horn@northwestern.edu, uri@northwestern.edu
Northwestern University

Purpose

While most researchers and educators agree that there is a need to integrate computational thinking (CT) in high school curricula, they differ in their conceptualization of CT, ways to achieve integration, and the rationale for doing so (Wing, 2006; Weintrop et al., 2016; Grover, 2018). In line with the ongoing efforts for curricular reforms to engage students in authentic disciplinary practices (e.g., Goldman, Ko, Greenleaf & Brown, 2018), Wilensky, Horn and colleagues have argued for CT integration in science and mathematics classrooms for the following reasons: (a) for students to learn authentic contemporary disciplinary practices, (b) prelogical effectiveness of thoughtfully integrated computational tools, and (c) to reach the widest possible audience, especially women and minorities who are underrepresented in computational fields (Wilensky, Brady & Horn, 2014; Weintrop et al., 2016).

One challenge for such integration is to develop pedagogically effective tools and curricula that teachers can adopt in their classrooms (Windschitl, Thompson, Braaten, & Stroupe, 2012). It requires both development of appropriate technological tools and novel methodological approaches to design curricula (e.g., Jeong & Hmelo-Silver, 2016). Inviting teachers to be design partners for co-designing such units has been an effective approach to increase their ownership and engagement in the appropriate pedagogical practices (Kyza & Georgiou, 2014).

In this paper, we investigate how our technology-based design approach mediated such co-design efforts. We present a case study of a researcher-practitioner design partnership for three years as it matured from using pre-designed CT-integrated curricula to co-designing new CT tools and curricula. We discuss changes in teacher's involvement in the co-design process and how the underlying technology platform that was used to design the computational tools supported those changes. We also discuss how the participation of the teacher in the co-design process influenced her shifts in her classroom teaching practice. We argue for the effectiveness of our technology-mediated design approach for co-designing CT-integrated science curricula.

Theoretical Frameworks

CT-STEM practices. This work is framed with the operational definition of CT-STEM (Computational Thinking in STEM) as a set of practices (Weintrop et al. 2016). This taxonomy categorizes CT-STEM practices in terms of four major strands: *data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices* (Table 1). This project focused on integrating these CT-STEM practices into high school science curricula. We developed new computational models and activities that are rooted in the disciplinary contexts for students to learn disciplinary ideas. We used two technology platforms to create computational models for students - (1) NetLogo (Wilensky, 1999; 2001), an agent-based modeling software, and (2) NetTango (Horn & Wilensky, 2011; 2012), a block-based programming interface to NetLogo which uses semantically meaningful blocks tuned to the content domain (Wilkerson-Jerde & Wilensky, 2010; Wilkerson-Jerde et al, 2015) (Figure 1).

Emergent Systems Microworlds (ESM). The design framework for the computational models is called Emergent Systems Microworlds (ESMs) (Dabholkar & Wilensky, 2019). ESMs are agent-based computational models that are designed as constructionist microworlds (Wilensky, 2001; Papert, 1980). Microworlds are learning environments that allow students access to external representations, often computational, to explore, investigate, and learn about disciplinary ideas (Noss & Hoyles, 2017). The ESM design framework is based on *restructuration theory* (Wilensky & Papert, 2010). ESMs enable learners to access representational forms that reformulate the disciplinary knowledge (Dabholkar, Anton & Wilensky, 2018; Dabholkar & Wilensky, 2020). Wilensky & Papert have coined the term *restructurations* to describe such knowledge reformulations.

Wilensky and Papert define structuration as the encoding of the knowledge in a domain as a function of the representational infrastructure used to express the knowledge. A change from one structuration of a domain to another resulting from the change in representational infrastructure is restructuration. In the design of ESMs, the agent-based models are the source of the restructurations. The use of agent-based models provides a powerful entry point into understanding an emergent phenomenon (Wilensky & Reisman, 2006). The agent-based restructurations reduce cognitive and perceptual limitations by allowing students to reason about emergent patterns at the system level by observing behaviors of agents (Goldstone & Wilensky, 2008).

In this paper, we argue that the ESM-based co-design approach allows a teacher to design pedagogically effective representations and devise appropriate pedagogical strategies to support student learning of CT practices. To understand the effectiveness of this technology-mediated co-design approach we investigate the following research question:

How does restructuration through ESM facilitate the co-design process for CT-integration into science units?

Methods & Data Sources

We present a longitudinal case study of a co-design partnership of three years between a researcher (the first author) and a teacher, Tracy (a pseudonym), a participant teacher in the CT-STEM project, which involved teacher professional development during summers and classroom implementations during the school years (Peel et al., 2020). These models and other computational tools were integrated into three published and publicly available open-source units (Dabholkar, Woods, Bain, Hall, Horn, & Wilensky, 2018; Dabholkar, Granito, Horn, & Wilensky, 2018; Granito, Dabholkar, Horn, & Wilensky, 2019). The units also had questions to scaffold students' learning of disciplinary ideas as well as CT practices. Tracy taught these curricular units in her biology classrooms at Greenville High School (pseudonym). See Table 2 for school demographics.

In year one, Tracy was given a CT integrated biology unit that was previously designed by the research team. In year two, Tracy provided direction to the researcher as they designed a new CT integrated biology unit. She chose the biology content, provided her lesson plans, and gave feedback as lessons were designed by the researcher. In year three, Tracy worked side-by-side with the researcher to co-design a new CT integrated unit during a Computational Thinking Summer Institute (CTSI). Each year Tracy taught the curricular units eight to ten class periods of 45-50 minutes.

Through this multi-year process, we collected data to characterize Tracy's experiences. The data sources include interviews with the teacher, weekly reflections, session recordings, a post-workshop interview. Additionally, we analyzed the unit designs and classroom video of implementations to support the analysis. We used a case-study method (Yin, 2004) to analyze how the ESM-based design approach supported and increased Tracy's involvement and changed her pedagogical approach while teaching in the classroom. We coded the interview data to identify changes in Tracy's involvement in co-design, classroom implementation, and views of student outcomes. For qualitative analysis of the video data, we used the activity-logging approach to identify episodes that illustrated Tracy's pedagogical practices while using computational tools. The claims and codes were triangulated where possible with multiple data sources, such as the intermediate and final co-design artifacts.

Findings

In this section, we present our qualitative analysis of how the agent-based restructurated representations increased Tracy's involvement in the co-design process across three years. We also discuss how the technology-mediated pedagogical discussions during the co-design process resulted in changes in Tracy's pedagogical practices to support students' learning of CT practices.

Restructurated representations. Agent-based restructurations allow learners to have visual and cognitive access to agent-level representations as well as system-level patterns. For example, learners can visualize behaviors of individual moose and wolves, as well as the population changes in the NetLogo model of prey-predator interactions in the *Year 1 Curriculum*. Since this unit used a pre-designed model, Tracy did not have any role in designing the model nor did she design the pedagogical activities using the model to support student learning. While teaching this unit, Tracy mainly encouraged students to explore and learn from the agent-based models, but she did not provide any specific suggestions to guide their explorations. Her focus was to use computational models to teach the concepts (Table 3 a). Tracy's pedagogical practices to support students' engagement in the embedded CT activities changed over the next two years.

In the *second year*, Tracy was more involved in figuring out how to make curricular activities computational (Table 3 b). During the very short co-design time of 4-5 hours in total, Tracy not only engaged deeply with the computational representations, she quickly appreciated their pedagogical effectiveness. In the class, she took a much more active role in guiding students' CT activities regarding modeling and simulation, and data practices. In her post-implementation interview, Tracy said that because the students were learning actively (by engaging in these practices), they had a "much deeper understanding" of the content (Year 2 post-implementation interview).

In the third year, Tracy played an active role in both designing the agent-based models and the design of pedagogical activities. Over the course of two weeks, Tracy and a project researcher discussed the designs of ESM computational models, as the researcher coded the models. During that time, the ESM design platforms, NetLogo and NetTango, allowed Tracy to view the models as they were being built to give feedback for making it more pedagogically effective and imagine pedagogical activities using the model. The researcher could quickly incorporate her design suggestions. They co-designed pedagogical activities and often discussed how to support students' learning using computational tools, which potentially impacted Tracy's teaching practices. A vignette in the next section demonstrates how Tracy used a pedagogical strategy based on her own experiences in the co-design process.

Technology mediated pedagogical strategies. In the *Year 3 curriculum*, in one of the lessons students used coding blocks to build an agent-based model of animal behavior (Figure 1 c). The agent-based coding blocks in NetTango allow an easy visualization of changes in agent behaviors based on the changes in their properties. In this vignette, the teacher and students are talking about two computational agents, *chambers*, that are used for an experimental setup (Figure 1 c). The dialogue between Tracy and a student transcribed from a video (Figure 2) illustrates how Tracy used a prior discussion between Sanjay (Pseudonym, for the researcher) and her to encourage a student to debug her model without giving a direct answer to her computational problem.

Student: *"In my chamber…. I am changing the sizes, but it won't get bigger"*
Tracy: *"This is how Sanjay told me. He never told me answers either. He made me figure it out. When you are taking a math class and when you are making a graph, what does x mean?"*
Student: *"I mean, it's like….." (makes a horizontal movement with hands)* (see figure 2)
Tracy: *"So what does Y mean?"*
Student: *Students "It's that". (Makes a vertical gesture)*
Tracy: *"So when you are changing the x and the y, are you changing the size of the chamber?"*

After Tracy asked this question, the student gestured positively indicating that she figured out the solution to her coding problem.

This vignette illustrates how ESM computational models allow students to quickly try out, reason, and figure out solutions to computational problems while engaged in model construction. Encouraging students to figure out the computation themselves by asking relevant questions is an important pedagogical strategy for helping students bridge the science and the CT. This is one of many learning strategies that came up during the co-design discussions to support the learning of CT practices in ESM-based units. The other strategies include asking students to 1) design robust experimental procedures to account for randomness, 2) modify a computational model to account for new factors, and 3) use automated experimentation and data collection strategies for creating and analyzing big datasets. The teacher used all these strategies and accompanying pedagogical moves that came up during the co-design discussions.

Scholarly Significance

The research community still has a lot to learn about how to teach CT in science contexts, and even less is known about how to support teachers in designing and implementing this integration (Sands, Yadav, & Good, 2018). In this paper, we described a novel co-design approach for creating CT-integrated science units. We discussed how the co-design partnership was mediated by emergent systems microworlds (ESM), the technological design framework used for creating computational models for learning CT practices. Visual and cognitive access to agent-level representations in an ESM allowed deeper and meaningful participation of the teacher in the co-design process. This work demonstrates the effectiveness of our ESM-mediated co-design approach to increase teacher ownership and engagement in crafting effective pedagogical practices.

Acknowledgements

This work was made possible through generous support from the National Science Foundation (grants CNS-1138461, CNS-1441041, DRL-1020101, DRL-1640201 and DRL-1842374) and the Spencer Foundation (Award #201600069). Any opinions, findings, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding organizations.

Tables

Table 1. The CT-STEM project's Taxonomy of CT-STEM Practices

Data Practices	**Modeling and Simulation Practices**	**Computational Problem Solving Practices**	**Systems Thinking practices**
Collecting Data	Using Computational Models to Understand a Concept	Preparing problems for computational solutions	Investigating a complex system as a whole
Creating Data	Using Computational Models to Find and Test Solutions	Programming	Understanding the relationships within a system
Manipulating Data	Assessing Computational Models	Choosing effective computational tools	Thinking in levels
Analyzing Data	Designing Computational Models	Assessing different approaches	Communicating information about a system
Visualizing Data	Constructing Computational Models	Developing modular solutions	Defining a system and measuring complexity
		Creating computational abstractions	
		Troubleshooting and debugging	

Table 2. Demographics of the School

School	**Race Demographics**	**Free/Reduced Price Lunch**	**English Language Learners**	**Individualized Education Plans**
Greenville	44% White, 30% Black, 18% Hispanic, 5% Asian, 0.4% Native American	39% free or reduced-price lunch	4.2% ELL	12% IEP’s

Table 3. Excerpts from Tracy’s Post-interview

Year 1	**Year 2**	**Year 3**
(a)“...using the models to show what happens. I think it's computational thinking… then to see what happens if you change parameters. And it's instant that you get the information like right away, predicting what would happen in an actual ecosystem. So using computer models to also teach the concepts.”	*(b) “... we met at least a couple of hours at least once a week...We started from the activities from [last] year working on the ones that were already made… he already asked me, so what do you do? What are the activities that you already do? … and then [he] talked about, well how can we turn these into the computational?”*	*(c)“I learned because behind the code… I didn't understand what made the agents work the way they work. I wouldn't even know what an agent was… You have to tell program to… they're not moving naturally. And you have to tell them to do that. You have to tell them what the preferences and how did you figure that out? Well, you actually did the research, right? So we knew this is a model of like real behavior.”*

Figures

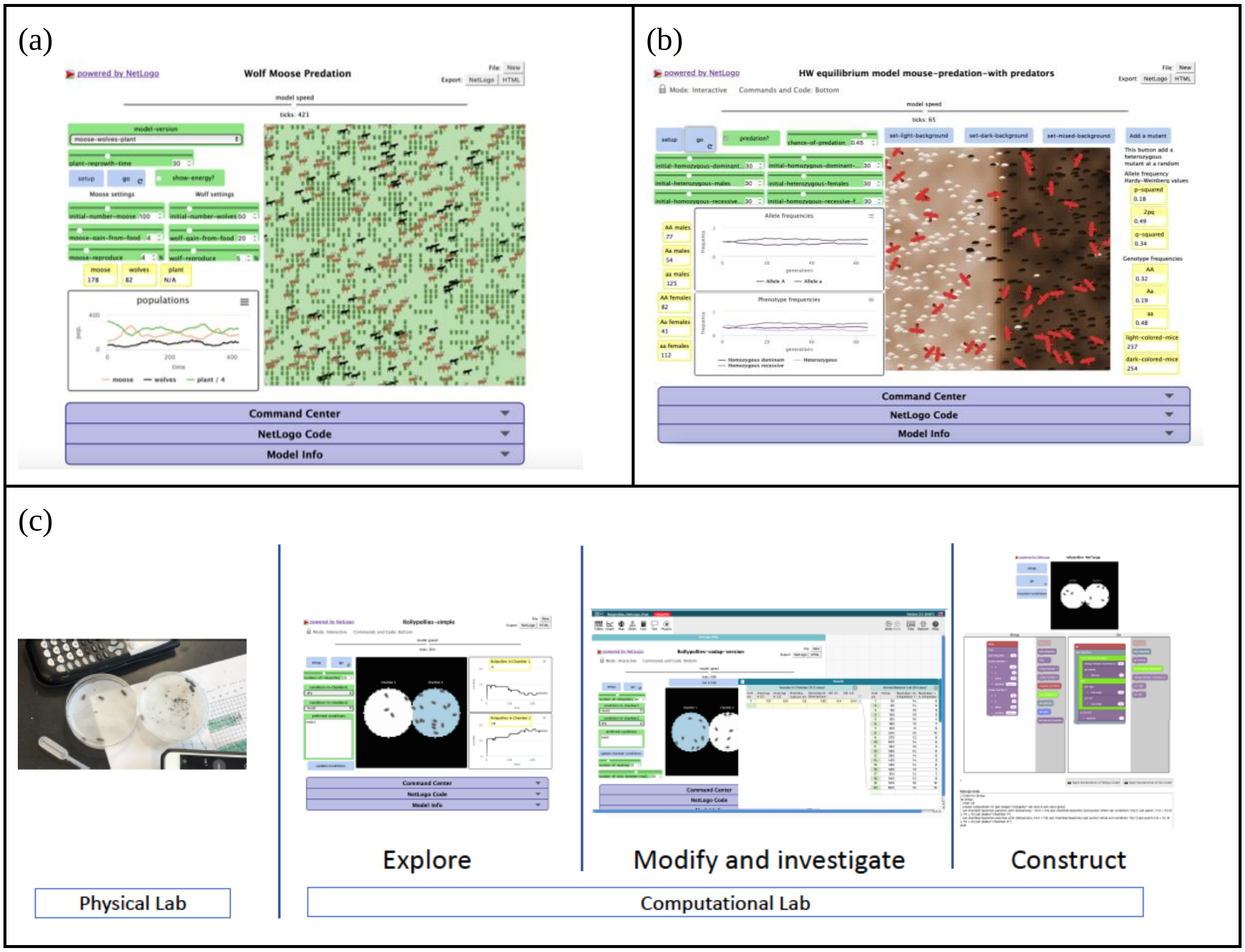

Figure 1: (a) A NetLogo model about prey-predator interactions in a pre-designed unit in 2017; (b) A NetLogo model for learning about Hardy-Weinberg equilibrium and natural selection in 2018; (c) The Curricular flow of the co-designed unit about animal behavior in 2019.

Figure 2: Tracy discussing a debugging strategy with a student. The white arrow shows the horizontal motion gesture made by the student while answering Tracy's question.

References

Dabholkar, S., Anton, G. & Wilensky, U. (2018). GenEvo - An emergent systems microworld for model-based scientific inquiry in the context of genetics and evolution. Proceedings of the International Conference for the Learning Sciences (ICLS) 2018, London, UK.

Dabholkar, S., Woods, P., Bain C., Hall, K., Horn M., Wilensky U. (2018). Evolution of populations. [curricular unit]. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. https://ct-stem.northwestern.edu/curriculum/preview/681/.

Dabholkar, S. & Wilensky, U. (2019). Designing ESM-mediated collaborative activity systems for science learning. Proceedings of International Conference of Computer Supported Collaborative Learning (CSCL) 2019, Lyon, France.

Dabholkar, S., Granito T., Horn & Wilensky (2019). Evolution of Populations to Speciation (Advanced). Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. https://ct- stem.northwestern.edu/curriculum/preview/961/.

Dabholkar, S. & Wilensky, U. (2020). Designing computational models as emergent systems microworlds to support learning of scientific inquiry. Proceedings of EpiSTEME8 International Conference to review research in Science, Technology and Mathematics Education, Mumbai, India.

Goldman, S. R., Ko, M. L., Greenleaf, C. & Brown, W., (2018). Domain-specificity in the practices of explanation, modeling, and argument in the sciences. In F. Fischer, K. Engelmann, J. Osborne, C. Chinn (Eds.) Interplay of Domain- Specific and Domain-General Aspects of Scientific Reasoning and Argumentation Skills. (pp. 121-141). NY, NY: Taylor Francis.

Granito, T., Dabholkar, S., Horn M., Wilensky U. (2020). Experimental Design and Computational Thinking. [curricular unit]. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. https://ct-stem.northwestern.edu/curriculum/preview/1523/.

Grover, S. (2019, February). Thinking about Computational Thinking: Lessons from Education Research. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (pp. 1283-1283). ACM.

Horn, M. S., & Wilensky, U. (2012). NetTango: A mash-up of NetLogo and Tern. AERA 2012.

Hmelo-Silver, C. E., Liu, L., Gray, S., & Jordan, R. (2015). Using representational tools to learn about complex systems: A tale of two classrooms. Journal of Research in Science Teaching, 52(1), 6-35.

Jeong, H. & Hmelo-Silver, C. E. (2016). Seven affordances of CSCL Technology: How can technology support collaborative learning. Educational Psychologis, 51, 247-265.

Kyza, E. A. & Georgiou, Y. (2014). Developing in-service science teachers' ownership of the PROFILES pedagogical framework through a technology-supported participatory design approach to professional development. Science Education International, 25(2), 55-77.

Noss, R., & Hoyles, C. (2017). Constructionism and microworlds. In Technology Enhanced Learning (pp. 29-35). Springer, Cham.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In *Computational thinking in the STEM disciplines* (pp. 151-164). Springer, Cham.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. Education and Information Technologies, 18(2), 351-380.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.

Windschitl, M., Thompson, J., Braaten, M., & Stroupe, D. (2012). Proposing a core set of instructional practices and tools for teachers of science. Science education, 96(5), 878-903.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of Knowledge Disciplines through new representational forms. In J. Clayson & I. Kalas (Eds.), Proceedings of the Constructionism 2010 Conference. Paris, France, Aug 10-14. p. 97.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. Cognition and instruction, 24(2), 171-209.

Wilensky, U. (2001). Modeling nature's emergent patterns with multi-agent languages. In Proceedings of EuroLogo (pp. 1-6).

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering Computational Literacy in Science Classrooms. Commun. ACM, 57(8), 24–28.

Wilkerson-Jerde, M., & Wilensky, U. (2010). Restructuring Change, Interpreting Changes: The DeltaTick Modeling and Analysis Toolkit. Paper presented at the Constructionism 2010 Conference, Paris.

Wilkerson-Jerde, M., Wagh, A. & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the DeltaTick project. Science Education, 99(3), 465-499.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.