Seeds of (r)Evolution: Constructionist Co-Design with High School Science Teachers

Jacob Kelter[1,2,3], *jacobkelter@u.northwestern.edu*
Amanda Peel[1,3], *amanda.peel@northwestern.edu*
Connor Bain[1,2,3], *connorbain@u.northwestern.edu*
Gabriella Anton[1,3], *gabriella.anton@u.northwestern.edu*
Sugat Dabholkar[1,3], *sugat@u.northwestern.edu*
Ümit Aslan[1,3], *umitaslan@u.northwestern.edu*
Michael S. Horn[1,2,3], *michael-horn@northwestern.edu*
Uri Wilensky[1,2,3], *uri@northwestern.edu*
Learning Sciences[1], Computer Science[2] and Center for Connected Learning[3], Northwestern University, Evanston, IL, USA

Abstract

In the decades since Papert published *Mindstorms* (1980), computation has transformed nearly every branch of scientific practice. Accordingly, there is increasing recognition that computation and computational thinking (CT) must be a core part of STEM education in a broad range of subjects. Previous work has demonstrated the efficacy of incorporating computation into STEM courses and introduced a taxonomy of CT practices in STEM. However, this work rarely involved teachers as more than implementers of units designed by researchers.

In *The Children's Machine*, Papert asked "What can be done to mobilize the potential force for change inherent in the position of teachers?" (Papert, 1994, pg. 79). We argue that involving teachers as co-design partners supports them to be cultural change agents in education. We report here on the first phase of a research project in which we worked with STEM educators to co-design curricular science units that incorporate computational thinking and practices. Eight high school teachers and one university professor joined nine members of our research team for a month-long Computational Thinking Summer Institute (CTSI). The co-design process was a constructionist design and learning experience for both the teachers and researchers. We focus here on understanding the co-design process and its implications for teachers by asking: (1) How did teachers shift in their attitudes and confidence regarding CT? (2) What different co-design styles emerged and did any tensions arise?

Generally, we found that teachers gained confidence and skills in CT and computational tools over the course of the summer. Only one teacher reported a decrease in confidence in one aspect of CT (computational modeling), but this seemed to result from gaining a broader and more nuanced understanding of this rich area.

A range of co-design styles emerged over the summer. Some teachers chose to focus on designing the curriculum and advising on the computational tools to be used in it, while leaving the construction of those tools to their co-designers. Other teachers actively participated in constructing models and computational tools themselves. The pluralism of co-design styles allowed teachers of various comfort levels with computation to meaningfully contribute to a computationally enhanced constructionist curriculum. However, it also led to a tension for some teachers between working to finish their curriculum versus gaining experience with computational tools. In the time crunch to complete their unit during CTSI, some teachers chose to save time by working on the curriculum while their co-design partners (researchers) created the supporting computational tools. These teachers still grew in their computational sophistication, but they could not devote as much time as they wanted to their own computational learning.

Keywords

Co-design, teachers, high school, science, STEM, modeling, computational thinking

Introduction

In *Mindstorms* (1980), Papert foresaw the increasing role of computation in STEM and STEM education and began to describe how STEM content can be reformulated using computational representations. Wilensky & Papert (2010) named this kind of transformation a restructuration of knowledge. In the decades since *Mindstorms'* publication, computation has transformed nearly every branch of scientific practice. Accordingly, there is increasing recognition reflected in standards documents and reports that computation and computational thinking (CT) must be a core part of STEM education in a broad range of subjects. The approach of integrating CT into STEM courses has broad benefits including: (1) aligning science education with authentic scientific practices (2) introducing students to powerful computational ideas and tools that can be used to understand a broad range of scientific concepts and (3) increasing participation in computing by incorporating it into courses that every student takes (Wilensky et al., 2014).

Previous work developed a taxonomy of CT practices for STEM education (CT-STEM) consisting of four broad categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices (Weintrop et al., 2016). This taxonomy can serve as a framework to help design and analyze STEM curricula that incorporate CT. Several successful examples of such curricula have been designed and implemented in a range of scientific subjects including chemistry, physics, biology and materials science (Blikstein & Wilensky, 2009; Dabholkar et al., 2018; Levy & Wilensky, 2009; Sengupta & Wilensky, 2009). These units were largely designed by researchers and then taught by the researchers alone or with partner teachers. While this approach had some benefits, it did not emphasize teacher agency and growth or sustainability of adoption.

This paper presents preliminary findings from the Computational Thinking Summer Institute (CTSI) in which STEM teachers joined us to co-design "CT-ified" curricular units. In contrast to past CT-STEM projects in which teachers primarily implemented pre-designed curricula, in CTSI they co-created the curricula from the beginning. In *The Children's Machine*, Papert asked "What can be done to mobilize the potential force for change inherent in the position of teachers?" (Papert, 1994, pg. 79). The co-design process itself is a constructionist learning experience for teachers which can increase their CT skills and confidence. Additionally, when teachers co-design curricula, they are more likely to take ownership of the change, a core element for sustaining adoption (Coburn, 2003). Such opportunities for teachers are seeds for cultural evolution in education.

Co-design with teachers as a dual approach to curriculum development and professional development was rewarding for both teachers and the research team. Most teachers gained CT skills and were happy with the curricula they developed. However, the process was not without its tensions. Recent literature has identified tensions inherent in teaching teachers and supporting constructionism in the classroom (Brennan, 2015; Hickmott & Prieto-Rodriguez, 2018). We discuss a tension that arose for some teachers between designing their curricula and devoting time to develop their own computational thinking and comfort with computational tools.

Research Questions

In the rest of the paper we describe CTSI, present case studies of four teachers' co-design processes and present a preliminary analysis to answer the following questions:

1. How did teachers shift in their attitudes and confidence regarding computation?
2. What different co-design styles emerged and did any tensions arise?

The Computational Thinking Summer Institute

Participants

Eight science teachers from urban and suburban public high schools in the midwestern United States and one university professor representing physics, chemistry, biology, statistics, earth science and materials science participated in CTSI along with nine members of our research team (six graduate students, two post-docs and a curriculum developer). Some of the teachers have worked with members of our team in the past two years to implement a CT-STEM unit, but none of them had previously participated in full-fledged co-design. The teachers' prior experience and comfort with computational tools ranged from total novice to fairly experienced.

Setting and Timeline

The institute lasted for four weeks during the summer, five hours per day. Three days a week the teachers joined the research team on our campus for workshops and co-design. The other two days the teachers were expected to work on their units from home. The in-person days consisted of two hours of work in the morning, one hour for group lunch, and two hours of work in the afternoon.

Workshops and Computational Tools

The first week of the institute was devoted to a "crash course" in computational thinking in STEM. Both teachers and researchers participated as students in a CT-STEM lesson taught by a researcher and then analyzed it from a pedagogical perspective. They also participated in lessons aimed at showcasing the various computational tools available, including "unplugged" computational activities, NetLogo (Wilensky, 1999), NetTango (Horn & Wilensky, 2011), and CODAP (Finzer, 2016).

NetLogo is an agent-based modeling environment widely used in both scientific practice and education. NetTango is a blocks-based interface to NetLogo that can be integrated with a web-based version of NetLogo, allowing for an easier entry into creating agent-based models. This was the first time our group used NetTango in a professional development setting with teachers. NetLogo and NetTango were emphasized, because agent-based representations are a natural fit for many scientific phenomena, making them easier to understand and study (Wilensky & Papert, 2010). CODAP is a data analysis and visualization tool. It has been integrated with NetLogo Web to facilitate easy data collection from computational models. Together, these tools can be used to support the full range of practices in the CT-STEM taxonomy cited in the introduction.

After the first week, teachers could (and did) attend optional workshops to improve their skills with various computational tools. They also attended two workshops by the Principal Investigators of the project on the relationships between CT, science, and education.

Constructionist Co-design

The co-design aspect of the institute had two goals: for teachers (1) to have a constructionist learning experience to enhance their own learning of CT practices and constructionism, and (2) to incorporate constructionist design and pedagogical principles in the curricular units that they designed.

After the first week, the majority of the institute consisted of co-design with built-in time for feedback and group reflection. Each teacher was paired with a member of the research team as a co-designer. The teams each decided how they wanted to work together.

At the beginning of each week, small groups of three to five teachers and researchers gave one another feedback. Every teacher received feedback from one other teacher and at least one researcher. Throughout the feedback and discussion sessions, the researchers foregrounded constructionist approaches for designing the embedded computational models and tools as microworlds (Papert, 1980) and sometimes more specifically as Emergent Systems Microworlds (Dabholkar & Wilensky, 2019). At the end of each week all of the participants came together to reflect on their experiences, sharing both accomplishments and challenges.

Case Studies

Methodology

The following three cases involve four different teachers representing a range of experiences and roles throughout CTSI. To construct these cases, we combined our own observations with responses from teacher surveys conducted at the end of each week of CTSI and interviews conducted at the end of the institute.

Physics: David and Elizabeth

David and Elizabeth are physics teachers. David has eighteen years of experience teaching and Elizabeth has seven. David has been mentoring Elizabeth since she began working at David's high school two years ago. Since then, they have closely worked to align their teaching practices, leveraging the Modeling Physics Instruction curriculum (Hestenes, 1997). Neither David nor Elizabeth have prior experience participating with this research team. However, David has participated in other university education programs focusing on learning and teaching computational thinking in the sciences. In post-interviews, both David and Elizabeth described their teaching as using a pedagogy that foregrounds students as "creators and askers and formulators of knowledge" (Elizabeth, Post-interview) and themselves as scaffolders of materials and experiences to support student growth.

During the co-design institute, David and Elizabeth partnered to redesign two units on motion maps and electrostatics. Throughout CTSI, the pair focused substantial time on designing and programming NetLogo models and positioned their co-designers as providers of just-in-time help to support programming goals. Their experience in the summer program can be summarized as follows. After the first week of scaffolded experiences introducing computational tools, David and Elizabeth both expressed concern about whether they would be able to integrate agent-based approaches of CT into physics. However, during the first feedback and co-design sessions, they brought twelve ideas for integrating CT to deepen student understanding of charge behavior and interactions, 2D motion, and electric potential. They first modified an existing NetLogo model on electrostatics to better fit the needs of their instruction style and curriculum, then designed a NetLogo model on charge interactions (which a co-designer implemented), and finally modified a model to create a 2-D motion map. They leveraged three researchers as co-designers who supported the creation of NetLogo Models and CODAP activities.

This focus on programming appeared to be a result of David's prior interest in and exposure to programming in other languages, including Logo when he was a child. In surveys and interviews, David expressed increasing confidence and enjoyment in the programming focused design, while Elizabeth expressed uncertainty and decreasing confidence. Based on Likert responses to pre-/post-survey and post-interviews, both David and Elizabeth expressed increased comfort in computational data practices. On the pre-survey, both expressed comfort with computational modeling practices as well, reflecting their integration of many PhET (University of Colorado) models and other simulations throughout their curricula in the past. The post-survey, however, captures a divergence in the pair's comfort with computational modeling. Elizabeth expressed a decrease in her comfort with computational modeling practices, disagreeing with statements on comfort defining computational modeling, finding resources, and helping colleagues. Additionally, she selected neutral on statements about adapting lessons, creating new lessons, and identifying students' practices in computational modeling. Conversely, David expressed increased comfort in computational modeling practices, strongly agreeing that he could answer student questions and find resources about computational modeling. We hypothesize that Elizabeth entered the summer institute with a conception of computational modeling practices more aligned with the use of simulations, like PhET models, that she had often integrated in her curricula and then was exposed to a more nuanced conception and set of associated practices which disrupted her confidence.

More generally, this divergence in confidence appears to be influenced by David’s and Elizabeth’s differing backgrounds in programming and how this impacted their needs and perceptions of the co-design experience. Elizabeth initially expressed interest and enjoyment from coding activities but then expressed a shift in confidence as the pair transitioned into modifying and programming models: “you guys gradually exposed us to different things we could do and changing the colors of the turtles. I thought it was really fun and made me feel relatively confident at the time. Then the next week came” (Elizabeth, Post-interview). She expressed feeling uncomfortable with NetLogo code and that she either needed greater exposure to programming in NetLogo or for the researcher co-design partners to develop models. Conversely, David expressed that he “had some experience with coding and Netlogo.... So [he] sort of took on the role of trying to start modifying existing NetLogo models” (David, Post-interview). He expressed excitement throughout the weeks as he tinkered with, debugged, and developed functioning models with the support of his co-designers. For David, this process appears to have allowed him to push his own programming abilities while largely leveraging his co-design team and teaching partner for thinking about content or providing just-in-time help. For Elizabeth, this process appears to have highlighted the practices she had yet to learn, resulting in lower comfort in modeling despite leaving the program with personally identified skills in reading code and programming. However, despite decreases in comfort on the post-survey, Elizabeth was very positive about the experience in the post-interview, mentioning that she felt confident reading code in NetLogo models, helping her students, and teaching the unit overall but that she would need either more NetLogo programming education or support from researchers to make units with new models.

Chemistry: Clara

Clara has taught honors and AP chemistry for ten years. She worked with the research team two years prior to participating in CTSI, teaching CT-integrated chemistry units designed by the research team. Clara viewed her role in the classroom as being a facilitator of student-driven learning.

> I am a guide on helping them figure out what questions to ask to be able to answer the bigger question that I've posed for them at the beginning of any given unit. And then I'm there to sort of help them put the puzzle pieces together of how to answer that big question… But I also am there to help them to work with each other and to understand that I am not the only one in the room with any knowledge. (Clara, Post-Interview)

During the institute, Clara co-designed a new unit with one member of the research team focused on heat and energy transfer during chemical reactions. Clara’s co-design process was curriculum-centric: she focused on designing the curriculum while providing her co-designer with the overall vision for the computational aspects of the unit. For example, Clara wanted a NetLogo model of the decomposition of potassium iodide in water to show the energy transfer taking place in the reaction. She described the phenomenon, what she wanted the model to do, and how she wanted it to look. Her co-designer then created the model and altered it based on her feedback. Clara’s final unit used two NetLogo models and computational data manipulation.

Based on a Likert response pre-/post-survey and post-interviews, Clara identified an increase in her comfort with CT practices and their integration after CTSI, specifically with identifying, defining, and teaching computational data practices. She also reported feeling more confident in her ability to modify curricula to include computational data practices:

> I've been very excited that I'm integrating some CODAP this year, which I ...didn't use at all prior to this summer… I'm excited because I already see other possible places in my year that I can use this. (Clara, Post-Interview)

Similarly, after CTSI Clara felt more confident in her abilities to identify computational modeling practices and to adapt curricula or create new lessons to include computational modeling.

Clara and her co-design partner included several opportunities for students to engage in constructionist learning. In one lesson, students construct statistical models using the computational data tool CODAP. Students also use a NetLogo model to run experiments &

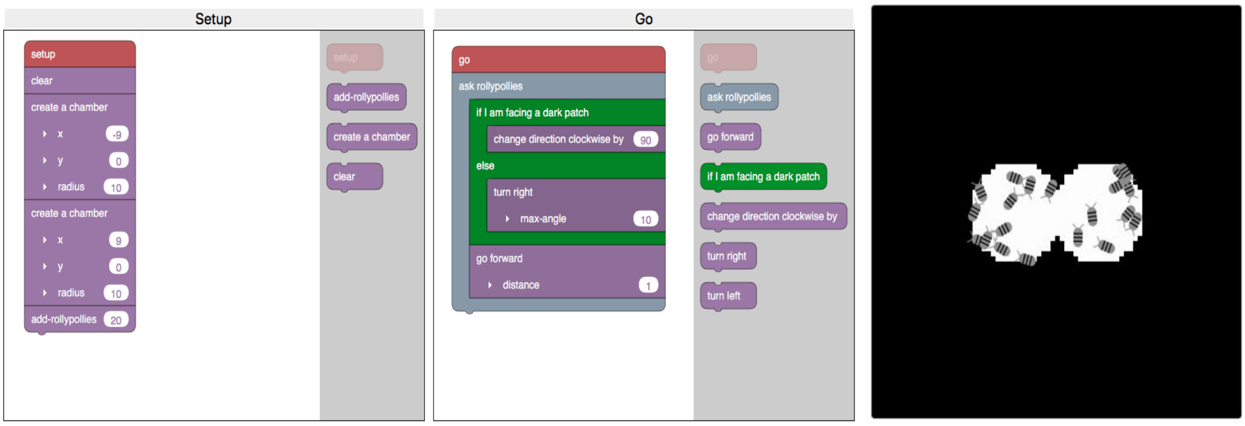
develop hypotheses on the nature of chemical reactions. This is in direct contrast to how Clara taught this lesson in the past when students wrote and balanced equations on a worksheet. Lastly, Clara and her co-designer included activities in which students discuss the underlying programming logic of the models to investigate how the models were constructed.

Biology: Tracy

Tracy has been a high school biology teacher for 30 years, teaching all levels of biology from basic to advanced and AP level classes. Prior to the institute, Tracy had worked with the CT-STEM team for two years in which her role and involvement in the design of the curricula expanded. Initially, she saw the implementation of curricula as a "research project" in which she did not want to "mess up" (Tracy, field notes). In the second year, Tracy identified natural selection as a content area that would benefit from computational integration and worked with a researcher to create a computationally enhanced version of the curriculum. During the implementation of this unit, Tracy took a more active role in the classroom, discussing a natural selection phenomenon using NetLogo models projected at the front of the classroom and connecting students' use of CODAP for data analysis with another phenomenon that she routinely taught.

Tracy entered CTSI with these two years of experience and, according to our pre-survey, moderately high comfort with integrating computational modeling and data practices into her classroom. During the institute, she chose to redesign a curricular unit on experimental design. Tracy saw integrating computational thinking into this curriculum as an opportunity to accomplish two goals: one, overcome obstacles that students encounter when performing physical experiments by integrating computational modeling practices, and two, make the student experience more authentic to actual scientific practice by integrating computational data practices.

The unit had students design experiments to find the preferred habitat conditions of the *pill bug* (a species of woodlouse colloquially also called roly-poly). Tracy decided that students would start out with a regular physical experiment using a simple environment of two connected chambers, one damp and one dry. The students would then place pill bugs inside of the environment and observe the change in population of the two chambers over time. After the physical experiment, Tracy then wanted students to digitally explore, modify and recreate the animal behavior experiment by using a NetLogo model and then creating a computational model using NetTango (Figure 1).



```
setup
clear
create a chamber
  x -9
  y 0
  radius 10
create a chamber
  x 9
  y 0
  radius 10
add-rollypollies 20
```

```
go
ask rollypollies
  if I am facing a dark patch
    change direction clockwise by 90
  else
    turn right
      max-angle 10
  go forward
    distance 1
```

Figure 1: A screenshot from the model of pill bugs that Tracy built with the help of her codesign team. Students use the NetTango blocks (left) to program the behavior of the pill bugs (right).

Incorporating programming was a large change for Tracy who avoided this aspect of computational thinking throughout her two previous years of involvement in the project. In fact, she noted this as one of her largest advancements:

> [before] I would not attempt to code anything all by myself. I would go in now [and program myself]. And [before] like if the kids had a question [about programming], I would direct it

> to the CT-STEM team. Now I can do something over there [at the student’s computer] and see if they can figure it out. I still need to work on that, but it's farther than I thought I would get. (Tracy, Post-interview)

In fact, while she saw her role in the classroom as a "facilitator" rather than as an "instructor," she mentioned feeling very intimidated by students' ability to quickly uptake new computational tools and surpass her own skills: "before they knew what my computer skills were like, so they just made fun of me" (Tracy, Post-interview). Rather than seeing programming as a realm of new opportunities for students to engage in the science content, she saw it as a blackbox in which she would not be able to independently support her students. The institute was the first time that she was able to engage with programming tools, specifically NetLogo and NetTango, as a learner herself. After having the experience of constructing a model from scratch, she felt able to support students in that process, even without being an expert programmer. While her co-design partner still supported her greatly in the development of the new model, Tracy took an active role in this process, designing features, contributing code, and actively choosing which programming blocks should be made available to students.

While gaining programming experience during the institute was key, Tracy also talked about how designing these models had informed her ideas of what it means to use a computational model:

> I just have to make sure that they (the students) understand that we coded the preference in there. We might be wrong ... and there are other things that might affect the behavior of the roly-poly that we're not aware of. So, you know, remember this is just a model. So, let's talk about the pros and cons of using the models. Just like there were in using the actual physical experiment. (Tracy, Post-interview)

Through co-designing the model, Tracy gained a deep understanding of how each design choice affects the outcome of the model. The model was no longer just a curricular tool to be *used* to understand the behavior of pill bugs; it was an artifact to understand what it *means to create* a computational model. She acknowledged that students could gain a deeper understanding of the phenomenon at play by building these models themselves.

At first, Tracy had major concerns about "deadlines" in her classroom—that by integrating computational activities, she would be losing time needed for other topics in the Advanced Placement Biology class. By the end of CTSI, she saw these computational open-ended design and investigation activities as *integrated* rather than *additional*: “I'm going to just really try to incorporate it by removing the direct lecture and incorporating the content. And it shouldn't take any longer. Like my evolution unit did not take any longer because I did [a CT-STEM unit]…and they learned just as much” (Tracy, Post-interview).

It can be difficult for even experienced teachers to change their pedagogical styles to integrate constructionist CT activities. However, Tracy's learning and co-design experiences during CTSI demonstrate her emerging understanding of how integrating CT in a constructionist way might fundamentally change her students’ experiences and science learning.

Discussion and Conclusion

Regarding our first research question of how teachers shifted their attitudes and confidence regarding computation, most teachers gained considerable confidence and skills in computational tools and various aspects of CT in STEM over the course of CTSI. The one exception to this trend was Elizabeth’s reported decreased confidence in computational modeling practices specifically. However, based on her responses in the post-CTSI interview, we interpret this as reflecting her more expansive notion of computational modeling at the end of the summer compared to the beginning. So, even this result can be seen as a positive development. Moreover, teachers expressed more nuanced understanding and appreciation of the ways that CT and computational tools can support students’ experiences and deepen science learning.

In answer to our second research question regarding co-design styles, a number of different styles emerged, which can be placed on a spectrum of teacher roles. At one end of the spectrum, teachers such as Clara focused almost exclusively on curriculum design, leaving the construction of computational models and tools to their co-designers. On the other end of the spectrum are teachers who focused equally on curriculum design and constructing computational models/tools. Tracy and David were close to this end of the spectrum, as they were actively involved with co-constructing computational models. Ideally, teachers become as comfortable with the computational tools as they are with curriculum design.

The different co-design styles emerged naturally in the co-design teams. While this was largely positive, one tension should be noted. On the positive side, allowing for different co-design styles meant that all teachers were supported to engage meaningfully with creating computationally enhanced curricula, regardless of their progress towards becoming computationally sophisticated educators. Those with more CT experience were able to co-construct new computational models and tools, while those with less could focus on designing their curriculum and advising on the pedagogical role of the computational models which their co-designers implemented. However, this affordance, combined with the time pressure to create a whole curricular unit during CTSI, created a tension for some teachers between developing their own computational skills and working to finish their units. One teacher remarked at an end-of-week reflection “I kind of wish we just had time to learn NetLogo.” For teachers with less computational experience, it was natural to have their co-designer primarily create the computational tools for their lessons, but this may have inadvertently limited their opportunities to increase their own CT skills. Based on the surveys and interviews, all of the teachers still increased in their confidence with at least some elements of CT, but for some teachers this progress may have been curtailed by the tradeoff between curriculum development versus computational tool building and attendant learning. In future years, we may offer additional skills-based workshops outside of the four-week CTSI as one way to ease this tension.

In the school year after CTSI, the teachers implemented their units with support from the research team. Future work will analyze these implementations to further understand the role of co-design in shaping teacher practice and student outcomes in CT integrated science curricula. Building from the lessons and successes of this inaugural CTSI, the research team will host a second institute this summer for both new and returning teachers.

All in all, CTSI was a rewarding experience for both the teachers and the research team. Engaging teachers in constructionist co-design helps them grow as educators and helps us grow as researchers, especially in our sensitivity to the needs and tensions faced by teachers. Most importantly, co-designing with teachers empowers them to be agents for constructionist cultural evolution.

Acknowledgements

We thank Sally Wu, Golnaz Arastoopour Irgens, and Hillary Swanson for their role in helping design and execute CTSI and all of our participating teachers for co-designing with us and implementing the units. 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 authors and do not necessarily reflect the views of the funding organizations.

References

Blikstein, P., & Wilensky, U. (2009). An atom is known by the company it keeps: A constructionist learning environment for materials science using agent-based modeling. *International Journal of Computers for Mathematical Learning*, *14*(2), 81–119. https://doi.org/10.1007/s10758-009-9148-8

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., Anton, G., & Wilensky, U. (2018). *Developing Mathetic Content Knowledge using an Emergent Systems Microworld*. Proceedings of Constructionism 2018, Vilnius, Lithuania.

Finzer, W. (2016). *Common online data analysis platform (CODAP)*. Retrieved from concord.org/codap

Hestenes, D. (1997). Modeling methodology for physics teachers. *AIP Conference Proceedings*, *399*(1), 935–958. https://doi.org/10.1063/1.53196

Horn, M., & Wilensky, U. (2011). NetTango [Computer Software]. Evanston, IL: Center for Connected Learning and Computer Based Modeling, Northwestern University. Retrieved from http://tidal.sesp.northwestern.edu/nettango/

Levy, S. T., & Wilensky, U. (2009). Students' Learning with the Connected Chemistry (CC1) Curriculum: Navigating the Complexities of the Particulate World. *Journal of Science Education and Technology*, *18*(3), 243–254. https://doi.org/10.1007/s10956-009-9145-7

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* (2nd ed). New York, NY: Basic Books.

Papert, S. (1994). *The Children's Machine: Rethinking School In The Age Of The Computer* (Reprint edition). New York, NY: Basic Books.

Sengupta, P., & Wilensky, U. (2009). Learning Electricity with NIELS: Thinking with Electrons and Thinking in Levels. *International Journal of Computers for Mathematical Learning*, *14*(1), 21–50. https://doi.org/10.1007/s10758-009-9144-z

University of Colorado. (n.d.). PhET Interactive Simulations. Retrieved October 24, 2019, from PhET website: https://phet.colorado.edu/

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. https://doi.org/10.1007/s10956-015-9581-5

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

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM*, *57*(8), 24–28. https://doi.org/10.1145/2633031

Wilensky, U., & Papert, S. (2010). *Reformulating Knowledge Disciplines through New Representational Forms 1*. Presented at Constructionism, Paris.