INTEGRATING COMPUTATIONAL THINKING IN HIGH SCHOOL BIOLOGY AND CHEMISTRY CLASSES

Robert Gotwals[1], Daniel Heck[2], Joan Pasley[2]

[1]North Carolina School of Science and Mathematics (USA)
[2]Horizon Research, Inc. (USA)

Abstract

Computational thinking is identified as one of the "essential skills for 21st-Century students." [1] Studies of CT in school programs are being funded by many organizations, including the United States National Science Foundation. In this paper, we describe "lessons learned" over the first two years of a research program (PREDICTS: Principles and Resources for Educators to Infuse Computational Thinking in the Sciences) with the goal of developing knowledge of how to integrate CT into introductory high school biology and chemistry classes for all students. Using curricular modules developed by program staff, two in biology and two in chemistry, teachers piloting the program engaged students in CT with computational evidence from authentic tools in order to develop understanding of science concepts. Each module, representing about a week of instruction, addresses science ideas in the prescribed course of study for high school programs. Project researchers have collected survey data on teachers': (1) beliefs about effective science teaching; (2) beliefs about their effectiveness as a science teacher and their students' ability to learn science, and; (3) content preparedness. In addition, we observed module implementation, collected and analyzed student artifacts, and interviewed teachers at the conclusion of module implementation. Preliminary results indicated some challenges (access to technology, varying levels of experience among students) and cause for optimism (student and teacher engagement in CT and the computational tools used in the modules). Continuing research efforts are described in this paper, along with descriptions of the curricular modules and the use of observations and "CT check-ins" to assess student engagement in, application of, and learning of CT.

Keywords: Computational Thinking, K–12 Education, Computer Education.

1 INTRODUCTION

Over the past 10 years or so, the focus on computational thinking (CT) and its importance in the STEM fields has increased significantly in the STEM and STEM education literature. Peter Denning and Matti Tedre, both professors of computer science (Naval Postgraduate School and the University of Eastern Finland, respectively), are among the more regular contributors to this discussion, including in the pages of the *Communications of the ACM*. In their text *Computational Thinking* [2], they describe the long history of CT, making the (bold) statement that "computational thinking evolved from ancient origins over 4,500 years ago to its present, highly developed, professional state," locating the beginnings of CT in the Mesopotamian civilizations. More realistically, they describe the work of mathematicians such as Euclid (300 BCE), the Greeks (400–350 BCE), Newton and Leibnz (1680), and Gauss (mid-1800s). These mathematicians laid the foundation for many of the core mathematical techniques that now can be implemented on a large scale with high-performance computers, capable of trillions of calculations per second.

The current emphasis on CT can mark the early 2000s as the start of many conversations, especially in education circles. The term "fluency with information technology" (and a textbook by that name) was adopted by a group of high school teachers, primarily teachers of computer science. The focus of CT was, and to a large extent still is, in the technologies, techniques, and tools of computer science. However, Denning and Tedre [2 pp. 16–17] offer this observation:

"A turning point came in 2006 when Jeanette Wing, then starting as an assistant director at the US National Science Foundation (NSF), reformulated the quest from fluency to CT. She proposed that CT was a style of thinking that everyone needed to learn in the computing age. At the NSF she mobilized significant resources to train teachers, upgrade the Advanced Placement test, design new 'CS [computer science] principles' first-courses for colleges, define CT for the K–12 education sector, and issue curriculum recommendations for K–12 schools. This 'CS for all' movement has achieved much greater penetration of computing into K–12 schools than any of its predecessors."

This movement has continued and expanded in the US education community by the prominence of CT in the Next Generation Science Standards [1] and by a significant number of journal articles, blogs, discussion forums, YouTube podcasts, and other media that extol the importance of CT, especially in K–12 education. Although there is consensus on the need for developing students' CT skills, how to do so in K–12 classrooms is not universally agreed upon nor has it been systemically operationalized. The project described in this paper (PREDICTS: Principles and Resources for Educators to Infuse Computational Thinking in the Sciences) has been funded under the NSF's STEM+C program, one of many CT-focused research programs. These programs, and the research being conducted in them, represent a substantial effort by the NSF to understand the role of CT in the K–12 classroom, especially in STEM subjects.

Fig. 1 [3] shows a relatively modern timeline of CT, with Wing's work squarely in the middle, but with beginnings in the 1970s and 1980s in Don Knuth's revolutionary work in computer science and Seymour Papert's development of tools such as the LOGO programming environment for young students. Groups like the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) have embraced CT concepts in a significant manner.

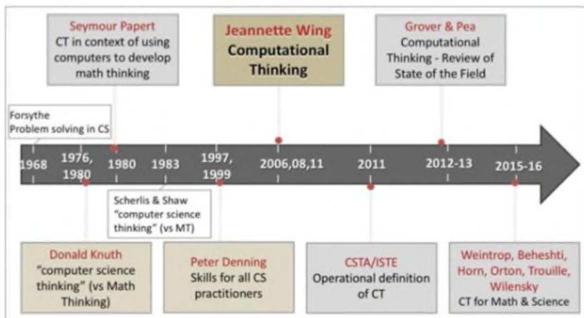

Fig. 1. Computational Thinking Timeline

One of the more widely-cited taxonomies of CT is that of Weintrop and colleagues [4]. Uri Wilensky, well-known for his work with the NetLogo agent-based programming environment, is also one of the authors of this work. Their taxonomy describes four "practices" (see Fig. 2.) which served as a foundation for the work described in this paper, but we found difficulty fitting our work into these practices, and consequently moved to other CT frameworks as our efforts evolved.

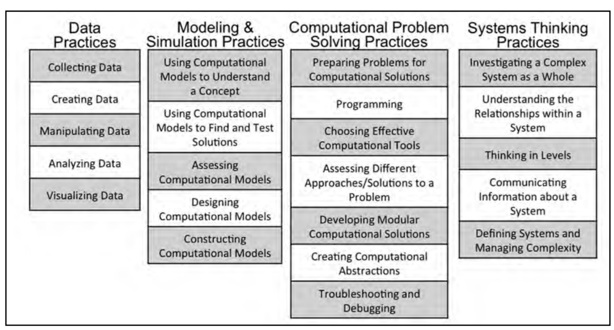

Fig. 2. Computational Thinking Taxonomy

Google, CSTA, ISTE, Brigham Young University [5], and others have developed a different taxonomy, shown in Fig. 3, which has served as a basis for the current work described in this paper. Four of the five essential skills—decomposition, pattern recognition, algorithm design, and evaluation—are the CT skills that are the focus of our development and research. Although the skill of abstraction is incorporated implicitly in our work, we elected not to explicitly address this skill. When operationalized in science instruction, we found it difficult to distinguish abstraction from other skills in this taxonomy.

Components of Computational Thinking		
Skills	Attitudes	Approaches
• ***Decomposition***: Breaking down data, processes, or problems into smaller, manageable parts. • ***Pattern Recognition:*** Observing patterns, trends, and regularities in data. • ***Abstraction:*** Creating a visual model or simulation of a problem that incorporates only the most important details. • ***Algorithm Design:*** Developing the step by step instructions for solving this and similar problems. • ***Evaluation:*** Ensuring that your solution is a good one.	• ***Confident:*** Believing in one's own ability to solve problems. • ***Communicative:*** Willing and able to communicate effectively with others. • ***Flexible:*** Able to deal with change and open-ended problems.	• ***Tinkering:*** Experimenting and playing. • ***Creating:*** Designing and making. • ***Debugging:*** Finding and fixing errors. • ***Persevering:*** Keeping going. • ***Collaborating:*** Working together.

Fig. 3. BYU Computational Thinking Taxonomy

In our review of the literature, both when writing our proposal to the NSF's STEM+C program and as a part of our ongoing efforts, we have identified two common threads. First, we continually find disclaimers to the fact that the field really does not have a good definition of what CT is, especially in terms of the classroom environment. It's probably self-evident that if you can't define it, you can't teach it, and if you can't teach it, you can't measure it. Given the already heavy burdens imposed on most high school teachers in terms of high-stakes, end-of-course testing, adding a new "must have" expectation for teachers is challenging. Second, most of the literature about CT is really focused on computer science. Our question was: where is the place for computational science? This is the interdisciplinary field of study that incorporates computer science, applied mathematics, and science as a way to understand interesting scientific problems, such as the behavior of electrons in atoms and molecules and the understanding of genomics data. Based on the Principal Investigator's exposure to a local school's large, specialized program (11 courses) in the computational sciences, the research question of how to best integrate CT (in a computational science environment) in introductory high school biology and chemistry classes for all students was developed, proposed, and funded. This paper describes early results from our work.

2 METHODOLOGY

In this section, we describe the various components of the research project funded by the NSF, including the curricular module development, in-school piloting, and early results.

Currently, science lab experiences in most high schools are typically limited to observational and experimental approaches, despite the increasingly prevalent use of computational methods in science research and application. The goals of PREDICTS are to generate knowledge, and related resources, about: (1) how to create instructional experiences for all high school students that engage them in CT in the learning of science; (2) what teachers require to effectively provide these experiences as a regular part of their science program; and (3) how to measure student engagement in and learning of CT in the context of science.

In order to reach the goals of the project, PREDICTS is conducting a three-year investigation that involves two primary strands of work. The first strand included developing, piloting, and studying the implementation of four curricular modules that use computational approaches to learning biology and chemistry. Feedback from four veteran pilot teachers informed revisions to the modules that are currently being field tested by 16 teachers (eight biology, eight chemistry) across seven school districts in North Carolina. There are two research questions for this strand of work:

RQ1. How can CT be operationalized in high school courses?

a. What are the design features of curricular modules that feasibly and effectively infuse CT into high school science courses?
b. What is an appropriate balance in designing experiences that use computational tools to support learning science content authentically through computing, but also structure the use of the tools to make them accessible to students and feasible to implement?

RQ2. What do high school teachers need to understand and believe to promote CT in their science courses and to teach science using CT?

a. What professional learning experiences develop needed understandings and beliefs?

b. What support do classroom teachers need in order to effectively and efficiently integrate CT into high school science programs?

The second strand of work includes developing and piloting instruments to assess both engagement in CT and evidence of CT in student work. Two types of instruments have been developed. First, the project staff adapted a classroom observation protocol developed in a previous project to measure instructional opportunity for students to learn specific science ideas. The adapted protocol includes both CT learning goals and science content learning goals addressed through CT. Specifically, the instrument focuses on observable evidence of instructional opportunity for students to engage with and learn the CT concepts in the service of learning the targeted science ideas. The rating categories include:(1) appropriateness of science content addressed through CT, (2) motivation for and elicitation of ideas for using computing to answer scientific questions, (3) engagement with computational science and CT tools to generate and analyze evidence; and (4) using evidence to make claims and sense making about computational evidence and CT to answer specific questions.

Second, the project staff developed brief assessment tasks, known internally as “CT check-ins”. Each CT check-in is built into the sequence of the curricular modules where students are engaging in CT as a part of doing science or explaining science concepts using computational evidence. The administration of the tasks in each module is timed to capture assessable ideas at opportune moments. Two CT-check-ins were developed for each module. The research question for this strand of work is:

RQ3. How can CT integrated into science instruction be rigorously measured in terms of students’:

a. Opportunities to engage in and learn CT?

b. Learning outcomes in CT in the sciences?

The development and research for both strands of PREDICTS are ongoing. The remainder of this paper will focus on describing the development and piloting of instructional materials intended to engage students in CT in the service of learning science concepts. In addition, the findings from data collected during the pilot test of the materials are presented.

2.1 Curricular module development

Four curricular modules were developed for this work, two in biology and two in chemistry. Each module was designed to address topics already included in the North Carolina Standard Course of Study for these introductory courses. We wanted to be careful that the materials did not require or request that the teachers present information that was additional to what they were expected to teach or spend more time on the topics than they would typically.

The modules were based on instructional materials that have been used for many years at a STEM focused high school in the USA (The North Carolina School of Science and Mathematics) in a computational science program of 11 courses, including computational biology and computational chemistry (as well as physics, medicine, nanotechnology, scientific programming, digital humanities, and two research courses). Module developers had the following design constraints: the modules had to represent about 5 days of instruction (assuming a schedule of hour-long class periods); the modules had to address course standards to replace a teacher’s existing instruction (not requiring additional instructional time); and the activities could assume student mathematics knowledge was no higher than first-year high school algebra. In addition, the modules were designed to embody elements of effective science instruction based on learning theory [6].

Each curricular module was designed to have at least one “central computational activity” in which students engage with computational evidence for a science idea. These activities use an authentic tool that is used by the science research community to either access existing big data sets or generate data using models. For example, in one of the biology modules students searched a database maintained by the National Center for Biotechnology Information and used a tool called BLAST to locate the mutation that results in Sickle Cell Anemia. In chemistry, students generate a model of chemical bonding and use computational evidence from the model to make sense of the properties of different types of materials based on the nature of bonds.

Activities that lead up to the central computational activities enable students to engage effectively with the computational evidence that they will encounter in those activities. For example, these “enabling” activities help students understand what the data in the tools represent and/or what is happening within the computational process. Activities that follow the central computational activity are intended to help students reinforce or apply what they have learned.

The four curricular modules we have developed are: in biology – (1) Cracking the Code: Using Computation to Understand Genetic Disorders; (2) Traits to Phylogenetic Trees: Using Computation to Understand Relatedness Among Organisms; and in chemistry – (3) Orbit Jumping: Using Computation to Understand Atomic Emissions; and (4) The Bonding Triangle: Using Computation to Understand Chemical Bonding.

After initial drafts of the four modules were completed, each was reviewed by at least one member of the project's external advisory board for alignment to the targeted learning standards, appropriateness of the content and approach of the computational investigations, and potential to promote CT. (The advisory board consists of computational scientists from local universities, science educators, and computationally-savvy high school educators.) The project staff and members of the advisory board also identified teacher needs for effective teaching of the modules, including both science and CT ideas and practices, and learning-theory aligned pedagogy. All feedback informed the final revisions to the modules and the substance of professional development provided for teachers.

2.2 Piloting of curricular modules

Four experienced high school science teachers (two biology, two chemistry) working in four different North Carolina public schools were identified to pilot the PREDICTS modules during the 2018-2019 school year and provide detailed feedback on the modules and their implementation. In July 2018, the pilot teachers attended a two-day orientation to familiarize them with the modules prior to the pilot test and to get their initial feedback.

As a part of the orientation, pilot teachers had the opportunity to engage in CT and module activities as learners. In addition, teachers posed questions they might have when teaching the modules and discussed logistics issues. During the orientation, teachers also provided suggestions for the modules, such as questions that might be added for students, which were incorporated in revisions made prior to pilot test implementation.

Required research approvals were obtained from the participating schools, and pilot teachers implemented the modules in both honors- and standard-level classes. Teachers were observed and videorecorded as they implemented the modules. Researchers completed a daily log of instruction and conducted a daily debriefing interview with each teacher to get immediate user feedback on teaching with the modules.

3 RESULTS

Data collected during the pilot test were analyzed to identify needed revisions to the modules that were evidence-based. The key findings from these analyses are listed below along with implications for revisions and the resulting modifications to the modules.

Finding 1: There were varied technology issues depending on the types of computers students used and the web access provided (e.g. access to Java, blocked websites).

Because the modules will be used by teachers in schools with varying technology, these findings indicated that, if the modules were to be broadly accessible, teachers would need guidance on work-arounds when they encounter similar technology issues. As a result, the modules now incorporate suggestions for dealing with technology issues (e.g., suggesting where teachers can do a demonstration when students don't have access to the appropriate technology).

Finding 2: Within a school, the implementation of the modules was quite different comparing the standard- and honors-level classes. For example, more teacher guidance occurred in the standard-level classes and students in these classes appeared to struggle with the mathematics compared to students in the honors-level classes (even though the computational methods used incorporated mathematics that was no higher first year high school algebra). Also, the pace was slower in standard-level compared to honors-level classes.

The intent of the modules is to provide students in all introductory high school science classes with opportunities to engage with computational evidence and CT to develop an understanding of science. Findings from the pilot test indicated that the module activities needed restructuring and additional teacher guidance in order to provide access to students in different level courses. As of the result, project staff reviewed the module activities to identify complex tasks, broke these down into smaller steps, and revised questions posed in the student materials to emphasize each steps more clearly. Additional guidance was also added for teachers on how to modify selected tasks, with integrity to the intended goals, if a class was having difficulty.

Finding 3: The modules are intended as replacement units; that is, the module should replace the teaching of the same content in the same amount of time it would typically take to teach the content using non-CT activities. Pilot test data indicated that the modules replaced the content appropriately but took longer to implement than the activities they replaced.

Because the time required to teach the modules is a key factor in the feasibility of their use, the project staff worked to streamline them using two strategies. First, activities intended to teach prerequisite content were removed and directions were made more explicit about what students needed to know prior to starting each module. Second, enabling or extension activities deemed not essential for engaging students in CT or learning the intended science content were shortened or removed.

Finding 4: As expected, implementation varied substantially across pilot test classrooms.

The pilot test provided opportunities to observe different teachers implementing the activities of the curricular modules. Differences in implementation (e.g., how the teachers managed activities, questions posed to students) were carefully observed and documented. Successful strategies that engaged students in CT and advanced the learning of key science content ideas were identified and incorporated into the teacher guides.

Finding 5: Although there was evidence that students were engaged in CT, there was minimal focus on helping students attend to the types of CT they were doing in the activities and how those types of thinking were contributing to their understanding of the science ideas the modules target.

During the pilot test, teachers were made aware of the computational approaches students were using but were given autonomy to explicitly discuss these approaches during instruction when they were most comfortable doing so. Observational evidence indicated that students were engaging with computational tools and using CT skills during instruction, namely, examining data for patterns, experiencing decomposition of problems, applying algorithms to generate data, and evaluating solutions. Unfortunately, there was insufficient explicit attention to the CT skills incorporated into these modules, perhaps because the modules took longer to teach than anticipated. Because a primary purpose of the modules is to help students understand how they are learning about the world using computational data (which is different from experimental science), this message needs to be made explicit or it may not be addressed.

As a result of these findings, revisions were made to student materials and teacher guides to incorporate explicit attention to the four selected CT skills shown in Fig. 3. The project staff identified activities in each module where it is very evident that students are engaging in specific CT skills, added icons to alert teachers to the CT skills in these activities, incorporated opportunities for students to reflect on what they are doing, and labeled these CT skills with a “student-friendly” definition. In some instances (generally the second time in the module where students engage with a particular CT skill), the advantages of CT and one or more specific CT skills are highlighted for students, both in the context of the problem at hand and more broadly.

Fig. 5 shows an example of a student activity sheet (from the Cracking the Code bilology module) that asks students to reflect on their thinking in the context of the problem at hand. A sample student response to the activity is shown in red. This example shows how a student might describe the CT skill of decomposition in a genetics scenario.

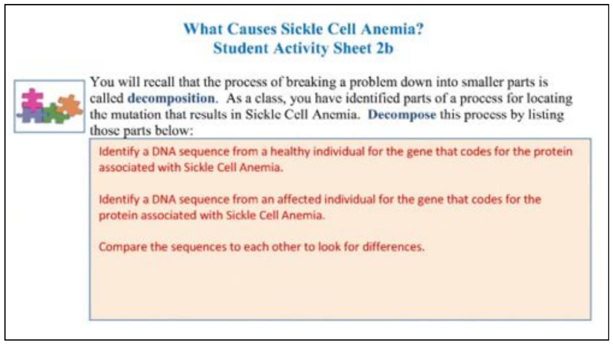

Fig. 5. Sample Student Activity Sheet with Sample Response

4 CONCLUSIONS

Based on the preliminary work we have completed for PREDICTS, it is clear that there are challenges to the integration of CT concepts and practices in the high school science classroom, but our evidence also suggests that there are opportunities for and benefits to doing so. Evidence from this study indicates that students in different levels of science classes can engage in CT as part of learning science but that those experiences need to be carefully designed so that they engage students with appropriate computational evidence and explicit attention to CT. It should be noted that this project did not consider what teachers might need to develop computationally-based science teaching materials on their own, so further work will be needed in this area. We anticipate that the “lessons learned” from the field test taking place during the 2019–20 school year will result in a much better understanding of integration of CT for diverse groups of science educators and their students in a broader range of school settings.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Grant No. 1741831, as a part of the STEM+C funding program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Next Generation Science Standards Lead States. *Next Generation Science Standards: For States, by States.* Washington, DC: The National Academies Press, 2013.

[2] P. J. Denning and M. Tedre. *Computational Thinking.* Cambridge, MA: The MIT Press, 2019.

[3] S. Grover. *A Tale of Two CTs* (and a Revised Timeline for Computational Thinking). *Communications of the ACM*, https://cacm.org/bloggs/blog-cacm/232488-a-tale-of-two-cts-and-a-revised-timeline-for-computational-thinking/. 2018.

[4] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky. “Defining computational thinking for mathematics and science class.” *Journal of Science Education and Technology*, vol. 25, no. 1, pp. 127-147. DOI: 10.1007/s10956-015-9581-5. 2016.

[5] E. Hunsaker. *Understanding Computational Thinking* (2nd. ed.). Salt Lake City, UT: Brigham Young University, 2018. http://scholarsarchive.byu.edu.

[6] E. Banilower, K. Cohen, J. Pasley, and I. Weiss. *Effective science instruction: What does research tell us?* (2nd ed.). Portsmouth, NH: RMC Research Corporation, 2010.