# Integrating STEM & Computing in PK-12: Operationalizing Computational Thinking for STEM Learning & Assessment

# Integrating STEM & Computing in PK-12: Operationalizing Computational Thinking for STEM Learning & Assessment


Shuchi Grover (organizer, chair), Looking Glass Ventures/Stanford University, shuchig@cs.stanford.edu
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu
Amanda C. Dickes, Gulf of Maine Research Institute, adickes@gmri.org
Amy V. Farris, The Pennsylvania State University, Farris, amy@psu.edu
Pratim Sengupta, University of Calgary, Pratim.sengupta@ucalgary.ca
Beth A. Covitt, University of Montana, Beth.Covitt@mso.umt.edu
Kristin L. Gunckel, University of Arizona, kgunckel@email.arizona.edu
Alan Berkowitz, Cary Institute of Ecosystem Studies, berkowitza@caryinstitute.org
John C. Moore, Colorado State University, John.Moore@colostate.edu
Golnaz Arastoopour Irgens, Clemson University, garasto@clemson.edu
Michael Horn, Northwestern University, michael-horn@northwestern.edu
Uri Wilensky, Northwestern University, uri@northwestern.edu
Shari Metcalf, Harvard Graduate School of Education, shari_metcalf@gse.harvard.edu
Soobin Jeon, Harvard Graduate School of Education, soobinjeon@gse.harvard.edu
Christopher Dede, Harvard Graduate School of Education, chris_dede@gse.harvard.edu
Gilly Puttick, TERC, gilly_puttick@terc.edu
Debra Bernstein, TERC, debra_bernstein@terc.edu
Kristen Wendell, Tufts University, Center for Engineering Education and Outreach, kristen.wendell@tufts.edu
Ethan Danahy, Tufts University, Center for Engineering Education and Outreach, ethan.danahy@tufts.edu
Michael Cassidy, TERC, michael_cassidy@terc.edu
Fay Shaw, Tufts University, Center for Engineering Education and Outreach, Fay.Shaw@tufts.edu
Daniel Damelin, Concord Consortium, ddamelin@concord.org
Steve Roderick, Concord Consortium, sroderick@concord.org
Lynn Stephens, Concord Consortium, lstephens@concord.org
Namsoo Shin, Michigan State University, namsoo@msu.edu
Irene Lee, Massachusetts Institute of Technology, ialee@mit.edu
Emma Anderson, Massachusetts Institute of Technology, eanderso@mit.edu
Ximena Dominguez, Digital Promise Global, xdominguez@digitalpromise.org
Phil Vahey, SRI International, philip.vahey@sri.com
Aman Yadav, Michigan State University, ayadav@msu.edu
Katie Rich, Michigan State University, richkat3@msu.edu
Christina Schwarz, Michigan State University, cschwarz@msu.edu
Rachel Larimore, Michigan State University, larimor5@msu.edu
Paulo Blikstein (discussant), Teachers College/Columbia University, paulob@tc.columbia.edu



**Abstract:** There is broad belief that preparing all students in preK-12 for a future in STEM involves integrating computing and CT tools and practices. Through creating and examining rich learning environments that integrate "STEM+C", researchers are defining what CT means in STEM disciplinary settings. This interactive session brings together a diverse spectrum of leading STEM researchers to share how they operationalize CT, what integrated CT and STEM learning looks like in their curriculum, and how this learning is measured. It will serve as a rich opportunity for discussion to help advance the state of the field of STEM+C integration.


## Motivation & Objectives

Few argue with the need for integrating computing and computational thinking (CT) as a tool to drive innovation in STEM. The learning sciences community also acknowledges that K-12 STEM learning must become more authentic in the 21st century through the integration of coding and CT. Efforts for "STEM+C" learning in the US received a fillip with CT listed as a disciplinary practice in the Next Generation Science Standards (NGSS; NGSS Lead States, 2013) and modeling emphasized in the NGSS and Common Core Mathematics Standards as a means to critically interrogate phenomena and understand simplifying assumptions. Although past efforts provide exemplars for the productive integration of math and science with computing (diSessa, 2001; Papert, 1980), developing integrated STEM+C curricula and measuring such learning is seen as challenging, in part because the broader community does not have a unified definition of CT (Grover & Pea, 2013). There is thus a need to better understand how to achieve productive integration and learning of STEM and CT, how to best involve STEM teachers, and how to assess learning in such integrated contexts.

The current landscape of STEM & computing/CT education affords ideal opportunities to convene leading researchers in the field to critically discuss current approaches for integrating STEM & CT. This symposium brings together researchers with a diverse set of approaches tackling this challenge head-on, from a variety of perspectives and pedagogical strategies at all levels of PK-12. In particular, symposium presenters will provide curricular details, examples, and insights into *1) how they operationalize CT, what CT definitions and frameworks guide their work, and how the integration of disciplinary STEM ideas with CT is engendered in their research and curricular approaches; and 2) the methods and measures they use to evaluate changes in students' STEM & CT learning.* Themes include: computational modeling in science and math (Grover et al.; Dickes, Farris & Sengupta; Metcalf et al.); co-design with teachers to modify STEM curricula to integrate CT (Irgens et al., Yadav et al., Dominguez et al) and designing teacher PD (Lee et al.); CT and systems thinking to understand complex phenomena (Covitt et al., Damelin et al.); and design activities that integrate CT & STEM (Puttick et al.). The symposium serves to showcase similarities in CT operationalization and assessment, curricular approaches (such as modeling), and methods for design and implementation (e.g., co-design with teachers) while also highlighting the diversity of perspectives that comprise a growing landscape of PK-12 STEM+C integration.

## Session Format

To promote active and productive discussion, the symposium will be conducted as an interactive poster session. Following brief teaser introductions on each project, attendees will be invited to view presenters' posters. This will provide attendees ample opportunities to examine and discuss curricular and methodological decisions made by the presenters, and how they may be adapted for attendees' own designs in a way that traditional talk do not allow. The symposium will close with an open discussion, in which the discussant will engage presenters and attendees in discussion around the main themes and any areas of interest that emerge during the session.

## Designing for Synergistic Learning of Science and CT in C2STEM

*Shuchi Grover and Gautam Biswas*

Introducing computational modeling into STEM classrooms provides opportunities for learning of STEM domain as well as CT concepts and practices. Programming a computational model of a physical phenomenon involves identifying appropriate underlying mathematical or computational rules that govern the behavior of relevant entities, making comparisons of the generated representations and explanations with observations of the target phenomenon, making sense of data, and iteratively debugging and refining models to generate accurate explanations of the scientific phenomenon to be modelled. These activities map to CT concepts and practices identified by Grover & Pea (2013) like abstraction, decomposition, testing, debugging, and refining of computational models, as well as data and modeling practices as outlined in Weintrop et al. (2016).

Our research focuses on designing and developing Collaborative, Computational STEM (C2STEM) a learning environment that aims to promote authentic STEM learning as outlined in the NGSS (2013). Students develop, refine, and interrogate computational models of real-world phenomena using a domain specific modeling language (DSML) in a block-based programming environment (Snap!/NetsBlox) to promote synergistic learning of science and CT. C2STEM curricular modules, designed with inputs from STEM classroom teachers on the project team, comprise instructional/inquiry tasks, modeling building and simulation modules, as well as embedded formative and summative assessments to support and measure learning of science and CT concepts in middle and high school. A preparation for future learning assessment measures students' abilities to generalize and apply CT concepts and practices across problem solving tasks and domains.

The summative assessment (adapted from prior studies) consists of multiple-choice (MC) and short response items to measure students' conceptual knowledge in physics and CT (separately). The designed embedded formative "check-ins" include modeling and MC items to measure students' integrated proficiencies in science and CT. The C2STEM environment has been empirically studied in middle and high school classrooms. For high school, we developed, tested, and refined curriculum and assessments for study of Kinematics in physics, and for middle school, we developed and examined a marine biology unit where students studied the effect of environmental changes on coral health. Findings across classroom studies show that students working with C2STEM developed a better understanding of concepts and practices of physics and CT than students who learned through a traditional curriculum. C2STEM also helped positively impact students' understanding of the purpose of modeling and how scientists reason using computational models.

## Heterogeneity and Practice: Programming as Expressive Media for K12 STEM

*Amanda C. Dickes*, Amy V. Farris*, Pratim Sengupta* (*Equal contribution)*

Dominant arguments for teaching programming include preparing a skilled workforce, investing in learners' economic mobility, and broadening entree to the STEM pipeline. While important, these drivers often result in K12 curricula and initiatives that place a greater emphasis on learning the isolated technicalities of computing, rather than on providing teachers and students opportunities to reflect in and on their own computing practices (Lye & Koh, 2014; Grover & Pea, 2013). Our response counters technical rationality (Schön, 1987) by illustrating how coding and CT can become a literacy for heterogeneous forms of experience, broadening and deepening what it means to code in the K12 STEM classroom and expanding learner and practitioner agency in the production of STEM knowledge (Sengupta, Dickes & Farris, in press).

Our poster will present results from a two-year, design-based microgenetic study in which elementary students used ViMAP, an agent-based programming and modeling platform (Sengupta et al., 2015), to complement existing science and math curricula through lessons *co-designed* and *taught* by the classroom teacher. Our analysis traces this integration and demonstrates how the heterogeneous nature of computing provided opportunities for the teacher to use computer programming as a context for her students to model natural phenomena. In particular, we will show how the classroom teacher framed programming as a modeling activity, and how working with code as a medium of "doing" math and science shaped the development of representational fluencies and modeling epistemologies, which in turn shaped her pedagogical work. We will also demonstrate how the students developed distinct and varied conceptions about what "counts" as a model among their code, the agents' enactments of that code, their material investigations, and their verbal explanations of what their programs mean. For both the teacher and the students, computational abstractions were deeply connected to measurements of the physical world, and mathematical, physical, and embodied forms of representations.

Across the academic year, the teacher's and the students' computing work helped to position them as the epistemic authority to interpret the phenomena they were modeling, the data collected from those phenomena, and the scientific and computational models themselves. Our work makes significant contributions to the praxis of CT and modeling in the K12 STEM classroom by demonstrating that heterogeneous forms for sense-making create contexts in which teachers' and students' representational and epistemic work is transformative in terms of the development of their epistemologies about the relationships among reality, scientific representation, and programming as ways of making sense of and explaining the world.

## Moving from Literal to Principle-Based Computational Reasoning: A Learning Progression for Integrating Computational Thinking with Earth and Environmental Sciences Instruction

*Beth A. Covitt, Kristin L. Gunckel, Alan Berkowitz, and John C. Moore*

To support development of a public prepared to participate in discussions that draw on Earth and Environmental Systems (EES) computational models, and consistent with NGSS (2013), we argue that CT necessary for public participation should be better integrated into K-12 science. While a growing body of work has defined frameworks for K-12 CT (e.g., Weintrop et al., 2016), this task has not been undertaken in EES. Furthermore, little research has examined students' ways of understanding computational models (e.g., Wilensky & Reisman, 2006). Our project aims to (1) integrate ideas from CT, systems thinking, and EES sciences to articulate a learning progression (LP) framework for CT in K-12 EES sciences, (2) develop and test instructional approaches for integration, and (3) provide evidence of student learning as a result of instruction designed with reference to the LP.

We will present our (1) LP framework with validity evidence from item response theory analyses for increasingly sophisticated ways students make sense of EES computational modeling, (2) instructional approach that utilizes connected experiences with multiple models moving from concrete toward abstract, and (3) evidence of student learning based on pre/post assessments from 1,276 high school students in two states. Our framework articulates three progress variables (defining the system, sensemaking with system data / representations, and explaining and predicting events with imperfect data/models), and three reasoning levels (lower: literal, middle: procedural, and upper: principle/model-based). Principle/model-based reasoning that integrates knowledge and practice in EES and CT is necessary for explaining and predicting events and processes in complex systems.

Several implications for designing effective instruction that integrates CT and environmental sciences have emerged. First, becoming more sophisticated in CT is not merely a matter of becoming incrementally better at interpreting computational system models. Rather, it includes shifting how one views what a system and/or computational system model is and what it is useful for. Also, as students move from engaging in literal reasoning about systems and system models toward reasoning that invokes scientific principles to explain and predict events and processes in systems, they become better positioned to think computationally (e.g., concerning boundaries, discretization, and parameterization) about those systems. We have found that an instructional approach that engages students in multiple connected experiences with different types of models of the same system (e.g., physical, conceptual, computational), and that moves from concrete to abstract experiences over time, can support students in developing increased sophistication and capacity for integrated systems and computational reasoning.

## CT-ifying STEM Education: Co-designing with teachers to integrate computational thinking into high-school math and science curricula

*Golnaz Arastoopour Irgens, Michael Horn, and Uri Wilensky*

Integrating CT and STEM (CT-STEM) helps students approach scientific inquiry in authentic and computationally sophisticated ways. In terms of assessing CT-STEM, few studies have empirically tested assessments or used contemporary learning sciences methods to measure learning (Grover & Pea, 2013). In various iterations of the CT-STEM project, we have produced a taxonomy of practices based on interviews with computational scientists (Weintrop et al., 2016), developed learning objectives and curricula that engaged a wide range of students, including those who are underrepresented (Swanson et al., 2019), and collaborated with teachers to "CT-ify" curricula and developed assessments. In our current project, teachers are co-designers in modifying existing science curricula to include computational tools and practices. The objectives are to help teachers develop an understanding of CT-STEM practices, discover why teachers think CT is powerful for science learning, and empower teachers to feel confident in teaching CT-STEM.

To assess learners' CT-STEM practices, we developed a pre-posttest consisting of an agent-based NetLogo (http://ccl.northwestern.edu/netlogo/) model seven questions related to the model that were aligned with CT-STEM learning objectives, and a rubric for each question. To characterize learners' developing practices, we coded student responses to reflection questions embedded in the curriculum using automated qualitative coding methods and visualized their learning using Epistemic Network Analysis (ENA) (Shaffer, 2017). Our analyses suggest that a combination of automated coding algorithms, ENA, and pre-post assessments, revealed differences among students' CT-STEM practices. For example, in a biology CT-STEM unit implementation with 41 students (Arastoopour Irgens et al., 2019a), students showed an increase from pre to post in (1) exploring a model and explaining how interactions produce system behaviors, (2) identifying simplifications, and (3) understanding a model's range of applications. Moreover, students with high scores were more likely to justify agent actions in the model and link justifications to biological phenomena. In contrast, students with low scores were more likely to discuss agents' actions but less likely to justify and link agents, actions, and data. In another example, in a chemistry CT-STEM unit with 384 students, ENA networks and a clustering analysis revealed that students identified relationships among micro-level particles and macro-level phenomena, such as pressure and temperature change, in various ways (Arastoopour Irgens et al., 2019b). Current work on this project involves (1) expanding the CT-STEM taxonomy to include developing algorithms, data mining/visualization, and designing/building computational models, (2) refining pre-post assessments to align with additional practices, and (3) continue co-designing with teachers to "CT-ify" curricula that integrate CT and STEM learning.

## Computational thinking and modeling for elementary science education via immersive virtual worlds

*Shari Metcalf, Soobin Jeon, Amanda Dickes, and Christopher Dede*

The NGSS (2013) identifies CT and scientific modeling as essential STEM education practices. Computational modeling is a useful subset of CT well suited for elementary science education. Modeling is an authentic science practice through which knowledge is constructed, and engaging students in computational modeling provides opportunities to make scientific concepts more accessible and to enhance student understanding of phenomena (Dwyer et al., 2013; Sengupta et al., 2013).

Immersive virtual environments provide highly engaging and situated experiences (Dede, 2009) for science learning, and visual block-based programming interfaces offer new opportunities to make coding accessible in elementary school science. The EcoMOD research project blends two modalities, a 3D virtual forest ecosystem and a 2D visual block-based programming tool, in order to support computational modeling and ecosystem science learning. During the 14-day EcoMOD curriculum, 3rd grade students explore an immersive virtual forest ecosystem, collect data, observe change over time, and program an agent-based computational model of a beaver building a dam. As students run their model, they observe emergent outcomes as their programmed agent impacts other elements of the ecosystem.

The curriculum was implemented with 7 teachers and approximately 150 students; data analysis is currently in progress. Our computational modeling assessments used qualitative methods to examine student artifacts: computational models constructed over three class periods, and hand-drawn concept map representations of the ecosystem three points during the curriculum. Analysis of computational models centered on student use of the agent-based modeling language, focusing on growth in coding skills, including understanding of sequencing, loops, and conditionals. We developed a rubric to assess student progress and identify common errors, both semantic and syntactic (Lane & VanLehn, 2005). Early analysis of students' computational modeling activities found that, in a sample of 35 pairs of students, all students achieved some steps towards programming the beaver building a dam, though only 31% were able to construct a complete model on their own. Analysis of student concept maps centered on the type and number of connections identified, as well as student reasoning used to make claims. Initial findings indicate that, over the course of the curriculum, the concept maps were significantly more detailed and included more complex pathways and connections between factors in the ecosystem over time. In group concept mapping activities, students cited evidence from both the immersive 3D world and the 2D computational models. We believe our strategies for assessment of student models can generalize to a range of learning experiences.

## Computational Thinking Practices in an Interdisciplinary Middle School Curriculum

*Gilly Puttick, Debra Bernstein, Kristen Wendell, Ethan Danahy, Michael Cassidy, and Fay Shaw*

Most learning in middle schools is segregated by discipline, yet recent trends in science and engineering education suggest this approach is outdated. The Designing Biomimetic Robots project is designed to support students to think across disciplines. Students participating in this 4-week curriculum first study the natural world to learn how animals accomplish different tasks, then engineer a robot that is inspired by what they learned.

Students deploy a set of practices that span computing, engineering, and biology. In addressing the robotics design challenge, students create a conceptual model of an animal structure-function (S/F) relationship by decomposing the animal system and abstracting relevant principles to create their model (Clement & Rea-Ramirez, 2008), then use the model as a basis for conceptualizing their robot design, and building a functional robot. Students engaged in problem scoping, iterating on their designs, and algorithmic thinking and debugging while programming (Grover & Pea, 2013), all elements of computational thinking (CT). To guide analysis of student work, we defined each of the CT practices, then operationalized each practice with respect to the two disciplines (see Table). Our poster will demonstrate how application of these codes to student work helps us understand students' sensemaking in the interdisciplinary environment.

Design work is a good vehicle for promoting disciplinary learning (Kolodner et al., 2003), and robotics design allows to practice CT skills (Sullivan & Heffernan, 2016). We found that the design task encouraged students to develop models of S/F relationships. When students explained their robot design choices, they argued for the relationships between animal S/F and robot S/F, thus understanding S/F relationships in a deeper way than a design task in the disciplines alone.

| **Computational Thinking definition** | **Operationalized in: Biology** | **Operationalized in: Engineering** |
|---|---|---|
| Decomposition: Breaking a sequence into steps/breaking a large problem down into several smaller problems | Identifying the structures that help an animal dig | Identifying multiple parts needed to enable a robot to perform in certain way |
| Abstraction: Identifying and representing the most important features in a model or design sketch | Conceptualizing and labeling the relevant movements/ functions of digging structures | Matching understanding of digging movements to an engineering mechanism, creating a design sketch |
| Algorithmic Thinking: Creating a series of ordered steps to carry out a task | | Defining the sequence of actions that each component of the robot will take. |
| Iteration: Refining a sequence of operations to achieve a result successively closer to a desired outcome | | Using results from a test to re-design a particular robot component |

## Situating Computational Thinking in the Context of Systems Modeling Using an Approach to Expand Equitable Access

*Daniel Damelin, Steve Roderick, Lynn Stephens, and Namsoo Shin*

Today's most challenging problems, from climate change, to understanding social networks, are best tackled by drawing on disciplinary core ideas across multiple STEM fields and practices. Two types of thinking are critical in addressing these types of issues, systems thinking (ST) and computational thinking (CT). Computational models and the underlying ST with its emphasis on system structure and behavior are invaluable in understanding complex phenomena and developing solutions. Wing, Cuny, and Snyder's (2010) definition of CT broadens CT to include all people working on solutions, not just programmers.

Given the broad application of CT and ST across a spectrum of phenomena, it is critical that we develop a strategy for engaging *all* students in developing their ability to use CT and ST as a lens for understanding the world around them. Two things are necessary to facilitate this change: (1) a framework that describes how CT and ST can be contextualized in a common scientific practice like system modeling, which can be integrated throughout STEM classrooms, and (2) development of a tool that makes student building of computationally solvable system models accessible.

To develop the framework, we reviewed a wide range of perspectives on CT. Two main branches of thought emerged, those suggesting a generic approach such as Grover and Pea (2018), and those that situate CT in STEM (Weintrop et al., 2016). Systems thinking definitions vary as well (Sweeney & Sterman, 2000; Stave & Hopper, 2007). Using the following criteria, we reviewed multiple papers defining CT and ST to generate a set of practices we incorporated into the framework: (a) Which aspects of CT or ST are common across multiple definitions? (b) Are these aspects generalizable across applications of CT or ST in solving a variety of problems? (c) Is there enough specificity to a particular aspect of CT or ST such that it can uniquely be applied to that domain? and (d) Can the aspect be operationalized in a way that it can be measured for research in STEM education? By applying these criteria, we developed a framework that makes explicit how elements of CT and ST are utilized in various phases of system modeling. For example, one phase of system modeling in our framework is *Define the Boundaries of the System*, which brings together *Defining a system* from ST and *Decomposing problems* and *Representing data through abstractions* from CT.

Creating the tool to facilitate student engagement in system modeling is ongoing and involves designing a pathway for students to use a visual interface for creating a runnable system model without the need for writing complex equations or traditional coding. Together with the framework we have a way forward for engaging students in developing critical ST and CT skills, developing curricula, and designing assessments.

## Designing Teacher Professional Development to Support CT Integration in Middle School Science

*Irene Lee and Emma Anderson*

The implementation of CT integration curricula that address new standards and initiatives is mediated by availability of curricula, accessibility to associated technology, and teacher preparation. Teacher preparation in particular has been noted as a critical limiting factor in the integration of CT into K-12 subject areas (Barr & Stephenson, 2011; Yadav, Hong, and Stephenson, 2016). To prepare science teachers for the integration of CT in science classrooms, Teachers with GUTS designed and studied a yearlong teacher professional development (PD) program aimed at increasing teachers content knowledge in and practice of CT, and providing a systematic model of CT integration in science through computer modeling and simulation. To align CT with teachers' science learning objectives for students, CT was operationalized as a thinking process necessary when developing computer models of scientific phenomena and using those models as experimental testbeds to conduct simulation experiments. The program's research focuses on teacher learning and subsequent enactment of a CT-rich modeling and simulation curriculum in middle school science classrooms. A design-based research approach was used to refine the PD, curriculum modules, and online PD network. We found that additional implementation supports such as teacher guides, online practice sessions, and webinars were needed to augment face-to-face PD offerings and retain connections to teachers throughout the year.

Scenario- and artifact-based interviews, and pre-post tests were used to measure teachers' knowledge, skills, attitudes, and application of CT. Observation and interviews were used to study teachers' enactment of the CT-rich Project GUTS curriculum. CT knowledge and skills were assessed in the pre-post test as independent conceptual knowledge in five domains: modeling and simulation; complex adaptive systems, agent based modeling, computer science constructs, and program tracing and decoding. We conducted a mixed method study of teachers' development of CT and how their CT understanding relates to their enactments of a CT-rich curriculum. Findings across studies show that participants' growth in CT/CS was correlated with their attendance hours and implementation hours though their specific enactments of curriculum (categorized as simulation-centric, coding-centric, abstraction-centric) were correlated with their disciplinary instructional goals more so than their growth in CT/CS. These findings suggest that broadening teachers' conception of CT to the full range of practices in our operationalization is needed before they attempt to make the match between CT practices and instructional goals.

## Enriching mathematics and science with computational thinking: Co-designing preschool activities with educators and parents

*Ximena Dominguez, Shuchi Grover, and Phil Vahey*

Although research on CT in early learning is limited, current CT definitions (Grover & Pea, 2013) include skills and practices that align with school readiness goals related to mathematics (Clements & Sarama, 2007) and science (NGSS, 2013). In order to understand how elements of CT align with the abilities and interests of young children and explore how CT can be integrated with early STEM learning in a mutually supportive manner, we developed learning blueprints to guide the development of integrated activities and partnered with teachers and families to co-design and pilot test resulting activities in classrooms and homes.

Early co-design meetings helped identify target content and initiate the design of the learning blueprint. As part of the initial stages of co-design, our team identified the following CT skills as productive starting points for our work: (1) problem decomposition; (2) algorithmic thinking; and (3) abstraction; and (4) testing and debugging. The blueprint listed developmentally appropriate learning goals for each CT skill. The team held co-design meetings involving discussions and interactive activities where teachers, coaches, parents, and researchers brainstormed and generated hands-on and digital activity sketches. Resulting activities were later pilot-tested in two classrooms and three homes. Data from the pilots were evaluated to inform next iterations to the activities. When linking CT with mathematics we found that activities designed to promote algorithmic thinking provided natural opportunities to promote visual spatial thinking, whereas activities designed to promote problem decomposition often provided opportunities to promote understanding of shapes. Counting, cardinality, and comparison of quantities were easily integrated across most CT activities. When linking CT with science we found that activities designed to promote abstraction naturally provided opportunities for children to engage in observation, description and sorting, whereas activities designed to promote problem decomposition naturally provided opportunities for children to engage in prediction and experimentation.

We are currently conducting a field study to examine the promise of the designed activities in 5 preschool classrooms and the homes of 10 families. In addition to classroom and home observations to examine implementation, we are conducting teacher and parent interview parents to gather their feedback, and one on one assessments with a subsample of children using a standardized early math measure (Weiland et al., 2012) and a series of play based learning tasks developed by our research team to examine the program's promise in improving children's learning of CT, mathematics, and science.

## Leveraging computational thinking to teach elementary mathematics and science

*Aman Yadav, Katie Rich, Christina Schwarz, and Rachel Larimore*

Recent educational reforms in K-12 education, such as the NGSS and Common Core State Standards, either explicitly or implicitly call out the need for students to engage in computational thinking (CT). While there is work starting to emerge on teacher training at the high school level, there has been less attention paid to how to bring CT to elementary classrooms. As elementary schools face additional pressures of accountability due to standardized tests in language arts and mathematics, core subjects like science are being pushed out (Marx & Harris, 2006) and there is even less time during the school year to add new initiatives like CT.

One approach to bring CT to elementary classrooms is to work within the constraints of K-12 systems and integrate it within core subject areas, such as mathematics, science, and language arts. In our work with elementary teachers, we used in-depth interviews to study how teachers conceptualize CT within the context of their mathematics and science instruction (Rich, Yadav, & Schwarz, 2019). We found teachers primarily focused on the problem-solving aspects of CT. However, teachers also saw connections between some CT practices and their teaching practices, with stronger connections to their mathematics instruction than science.

In this presentation, we will describe how we have productively used four CT practices—abstraction, decomposition, patterns, and debugging—with elementary teachers as a part of our NSF-funded project--CT4EDU. In particular, we will present how our unplugged approach to CT (i.e., without the use of computers or technology) shifted how our partner teachers conceptualized CT and how those conceptualizations influenced their mathematics and science instruction. Drawing from a number of data sources (e.g., focus groups, classroom video, and interviews), we will discuss how CT can be conceptualized in a way that removes barriers teachers see to integrating CT and allows them to more frequently embed CT in their curricula. We will also discuss how our focus on a few core CT practices can serve as an on-ramp for elementary teachers to bring computationally rich environments into their classrooms.

# References

Arastoopour Irgens, G., … Wilensky, U. (2019a). Modeling and Measuring Students' Computational Thinking Practices in Science. *Journal of Science Education and Technology.*

Arastoopour Irgens, G., Chandra, S., Dabholkar, S., Horn, M., & Wilensky, U. (2019b). *Classifying Emergent Student Learning in a High School Computational Chemistry Unit.* Paper presented at AERA 2019.

Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54.

Clement, J.J., & Rea-Ramirez, M.A. (Eds.). (2008). *Model based learning and instruction in science.* Springer.

Clements, D. H., & Sarama, J. (2007). Effects of a preschool mathematics curriculum: Summative research on the Building Blocks project. *Journal for research in Mathematics Education*, 136-163.
Cuny, J., Snyder, L., and Wing, J. (2010). Computational Thinking: A Definition. (in press)
Dede, C. (2009). Immersive interfaces for engagement and learning. Science, 323(5910), 66-69.
DiSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.
Dwyer, H., Boe, B., Hill, C., Franklin, D., & Harlow, D. (2013). Computational thinking for physics : Programming models of physics phenomenon in elementary school. *Physics. Ed. Research*, 133–136.
Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43.
Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, 19.
Kolodner, J. L., Camp, P.J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice. *Journal of the Learning Sciences*, *12*(4), 495-547.
Lane, H. C., & VanLehn, K. (2005). Intention-based scoring: an approach to measuring success at solving the composition problem. In *ACM SIGCSE Bulletin* (Vol. 37, No. 1, pp. 373-377). ACM.
Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, *41*, 51-61.
Marx, R., & Harris, C. (2006). No child left behind and science education: Opportunities, challenges, and risks. *The Elementary School Journal, 106*(5), 467-478.
Metcalf, S. J., Kamarainen, A. M., Torres, E., Grotzer, T. A., & Dede, C. (2018). EcoMUVE: A case study on the affordances of MUVEs in ecosystem science education. In Y. Qian (Ed.), *Integrating Multi-User Virtual Environments in Modern Classrooms* (pp. 1-25). Hershey, PA: IGI Global.
NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. Washington, D.C.: NAP.
Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, 27(2), 165-205
Schön, D. A. (1987). Educating the reflective practitioner.
Sengupta, P., Dickes, A., & Farris, A.V. (In press). *Voicing code in STEM: A dialogical imagination*. MIT Press.
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.
Sengupta, P., Dickes, A., Farris, A. V., Karan, A., Martin, D., & Wright, M. (2015). Programming in K-12 science classrooms. *Communications of the ACM*, *58*(11), 33-35.
Shaffer, D. W. (2017). *Quantitative Ethnography*. Madison,WI: Cathcart Press.
Stave, K. A., & Hopper, M. (2007). What Constitutes Systems Thinking? A Proposed Taxonomy. In *25th International Conference of the System Dynamics Society*. Boston, MA.
Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, *48*(2), 105-128.
Swanson H., Anton G., Bain C., Horn M., Wilensky U. (2019) Introducing and Assessing Computational Thinking in the Secondary Science Classroom. In: Kong SC., Abelson H. (eds). *Computational Thinking Education*. Springer, Singapore
Sweeney, L. B., & Sterman, J. D. (2000). Bathtub dynamics: initial results of a systems thinking inventory. *System Dynamics Review*, 16(4), 249–286.
Weiland, C., Wolfe, C. B., Hurwitz, M. D., Clements, D. H., Sarama, J. H., & Yoshikawa, H. (2012). Early mathematics assessment: Validation of the short form of a prekindergarten and kindergarten mathematics measure. *Educational Psychology*, *32*(3), 311-333.
Weintrop, D., ..., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology, 25*(1), 127–147.
Weintrop, D., ..., & Wilensky, U. (2014). Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms. In D. Reidsma, I. Choi, & R. Bargar (Eds.), *Proceedings of Intelligent Technologies for Interactive Entertainment,* Chicago, IL, USA (pp. 22–25).
Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering Computational Literacy in Science Classrooms. *Communications of the ACM, 57*(8), 24–28.
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. *Cog. & Instruction, 24*(2), 171-209.
Yadav, A., Hong, H., and Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st Century problem solving in K-12 classrooms. *Tech Trends. 60*(6), 565-568.