Journal of Science Education and Technology (2020) 29:1–8
https://doi.org/10.1007/s10956-019-09803-w

Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education

Irene Lee[1] · Shuchi Grover[2] · Fred Martin[3] · Sarita Pillai[4] · Joyce Malyn-Smith[4]

Published online: 3 December 2019

Abstract

This article provides an introduction for the special issue of the *Journal of Science Education and Technology* focused on computational thinking (CT) from a disciplinary perspective. The special issue connects earlier research on what K-12 students can learn and be able to do using CT with the CT skills and habits of mind needed to productively participate in professional CT-integrated STEM fields. In this context, the phrase "disciplinary perspective" simultaneously holds two meanings: it refers to and aims to make connections between established K-12 STEM subject areas (science, technology, engineering, and mathematics) and newer CT-integrated disciplines such as computational sciences. The special issue presents a framework for CT integration and includes articles that illuminate what CT looks like from a disciplinary perspective, the challenges inherent in integrating CT into K-12 STEM education, and new ways of measuring CT aligned more closely with disciplinary practices. The aim of this special issue is to offer research-based and practitioner-grounded insights into recent work in CT integration and provoke new ways of thinking about CT integration from researchers, practitioners, and research-practitioner partnerships.

Keywords Computational thinking · Disciplinary perspective · Integrating computational thinking · K-12 science · technology · engineering · and mathematics education

Computational thinking (CT) is the thought process involved in formulating problems such that their solutions can be expressed as computational steps or algorithms to be carried out by a computer (Cuny et al. 2010; Aho 2012; Grover and Pea 2013; Lee 2016). As such, CT can be seen as the connective tissue (Martin 2018) that links computer science (CS) to many disciplines. In particular, CT has been used as an investigative and problem-solving method that utilizes computer science concepts, tools, and techniques in science, technology, engineering, and mathematics (STEM). New integrated fields such as computational biology, computational chemistry, computational geometry, and computational physics have emerged that capitalize on the power of computation in extending discovery and innovation within classical STEM fields.

As computationally enabled scientific innovations and technological advances are reshaping the ways we live and the type and scope of problems we can pose and solve, there has been increasing interest in engaging K-12 students in CT. It has been touted as an essential competence that should be included in every student's skill set (Grover and Pea 2018) and acknowledged as a key scientific practice in the Next Generation Science Standards (NGSS Lead States 2013). The integration of CT in elementary and secondary education has been promoted by both national computer science education (Seehorn et al. 2011) and science education associations prompting many school districts and states to make wide-sweeping changes based on a limited research base (Stanton et al. 2017).

✉ Irene Lee
ialee@mit.edu

Shuchi Grover
shuchig@cs.stanford.edu

Fred Martin
fredm@cs.uml.edu

Sarita Pillai
spillai@edc.org

Joyce Malyn-Smith
jmsmith@edc.org

1 Massachusetts Institute of Technology, Cambridge, USA
2 Looking Glass Ventures/Stanford University, Stanford, USA
3 University of Massachusetts Lowell, Lowell, USA
4 Education Development Center, Waltham, USA

The integration of CT into STEM classrooms is relatively new. It is seen as having the potential to deepen STEM learning by positioning students as young scientists and innovators through engagement in authentic STEM practices. It has been proposed that students who learn to develop computational solutions and marshall computational tools, resources, and methods will advance their understanding of subject area content, CT skills, and awareness of modern uses of computation across STEM fields. This integration strategy also aims to bypass some of the difficulties associated with offering stand-alone courses (focusing on CS and CT) and adding additional subjects to an already crowded school day while addressing new standards in science, math, and computer science education. Thus, researchers and educators alike have embarked on projects to engage students in CT as early as elementary school as a means to prepare them with the foundational knowledge, skills, and practices for future endeavors in STEM fields.

Researchers have made progress in elucidating what CT integration entails. Recent attempts to integrate CT into science classrooms fall along a continuum from the addition of "coding" activities that provide little if any support of science learning (Lye and Koh 2014; Kazimoglu et al. 2012; Maloney et al. 2008; Tarkan et al. 2010; Touretzky et al. 2013; Grover et al. 2015); to the integration of CT in the service of science content knowledge as it currently exists in science textbooks (Sengupta et al. 2013; Wilkerson and Fenwick 2016; Benakli et al. 2016; Sherin 2001; Sherin et al. 1993; diSessa 2001); and the integration of modern uses of computation aligned with the work of STEM professionals (Uzzo and Chen 2015; Wilensky et al. 2014; Orton et al. 2016). Weintrop et al. (2016) provided a well-regarded computational thinking in mathematics and science practices taxonomy describing four main categories of practices (data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices) that form a definition of CT drawn from literature, interviews with mathematicians and scientists, and instructional materials. Yet, work in the field of CT integration suggests that much more guidance from empirical research is still needed to ensure the teaching of CT concepts and practices within STEM classrooms sets the foundations for future endeavors leading up to professional CT-integrated STEM practices.

This special issue proposes to connect earlier research on what K-12 students can learn and be able to do using CT with the CT skills and habits of mind needed to productively participate in professional CT-integrated STEM fields. In this context, the phrase "disciplinary perspective" simultaneously holds two meanings: it refers to and aims to make connections between established K-12 subject areas (science, technology, engineering, and mathematics) and newer CT-integrated disciplines such as computational sciences at the core of scientific discovery and innovation in a world driven by technology.

In the first framing article, Lee and Malyn-Smith (this issue) describe two strands of research that informed the formation of a framework for integration of CT in STEM classrooms. The first strand is a set of studies on the use of CT by STEM professionals (research scientists, product engineers, and data scientists) conducted by EDC that illustrates the transition of traditional scientists and engineers to computational thinking STEM professionals. The second strand of research is an analysis of a set of K-12 CT integration activities that was collected at the NSF-funded two-part "Workshop on Developing a Framework for Computational Thinking from a Disciplinary Perspective" (Malyn-Smith et al. 2018). They provide an overview of recent work in the CT integration domain and identify gaps between CT as it is being taught in K-12 and what may be needed to prepare students for CT integration in professional STEM workplaces. Other articles share interesting and productive research themes and compelling classroom-grounded strategies that address specific aspects of CT integration in K-12 as we explore ways to help students develop competence in computational thinking.

The next two articles describe research on the affordances and challenges of building representational fluency in the early grades as a foundation to CT. While both articles focus on computing concepts and computer programming into elementary classrooms (rather than integrations into specific STEM subjects), they offer a glimpse of thought processes of young children that can be further developed into a broader spectrum of CT integration into STEM classrooms as well.

Moore et al. (this issue) explored how young children in grades K-2 engaged with and moved among multiple representations as they managed the cognitive demands of various computational tasks. The children in the study were asked to translate information encoded in different representations including concrete, symbolic, pictorial, motor, and language representations. The authors found that students translated among representations by constructing intermediary representations such as gestures and placing objects in the environment to represent current state, an early form of simulation, to manage cognitive load. They contributed to the Lesh translation model, or LTM (Lesh and Doerr 2003), by adding gesture as a representational support. This article points to junctures in student activities where concepts of looping, parallelism, and decomposition as well as programming and debugging occur naturally and could be explicitly linked to CT used by professionals. For instance, when asked to draw a representation of a scene composed of tiles, and features such as barriers and bridges, students were seen to organize the task in different ways: one child drew each tile and its associated features; another child drew the matrix of tiles first then added features. This is akin to task vs. domain decomposition in parallel computing. In another example, upon seeing a repetitive pattern of steps (e.g., forward 5 steps) in coding cards, a student transformed the sequence into a single movement of 5 steps forward in the physical space hinting at looping. The authors contribute to this special issue by positing that understanding the strategies children naturally use when translating among representations may help educators embed scaffolds to support representational transitions and manage cognitive load in CT. They also point to an area for further study—the connection between CT and spatial reasoning. The difficulty young children had with spatial orientation tasks suggested that their mental rotation ability was not yet sufficiently developed but since spatial abilities have been shown to improve with practice and support (Feng et al. 2007; Uttal et al. 2013), this need not be a barrier to CT.

Dickes, Farris, and Sengupta (this issue) investigated how computer programming was integrated in an elementary school 3rd grade classroom through agent-based modeling in the ViMAP environment. The authors demonstrated how the classroom teachers' emphasis on mathematizing and measurement positioned computing as an epistemic tool (tool for knowledge construction) for solving real-world problems in kinematics (i.e., movement). Mathematization and measurement were supported by classroom norms for defining and designing "mathematically sound" computational models of motion. "Mathemization" (Lehrer et al. 2001) involves the highlighting of mathematics as a meaning-making lens through which the natural world can be systematized and described. The classroom norm was to assess what "counts as" an acceptable mathematical solution. Two forms of representations were integrated into classroom activities in the study of animal movement through their environment. The animals' movement within the agent-based model was compared to movement patterns in the real world. Drawing students' attention to stride and measuring stride first took place in a participatory activity in which students learned to measure and compare stride length when running and walking. Next, an animal's stride in the simulated environment, ViMAP, was investigated and analyzed for face validity (by comparing it to how students measured stride in the participatory activity). The authors concluded that curricular integration of computing within K-12 STEM contexts is a complex and challenging endeavor for both teachers and students. It involves the adoption of new literacies (e.g., programming) as well as disciplinary ideas. They propose connecting STEM and computer programming by framing CT and modeling as model eliciting activities (Lesh and Doerr 2003) in the science classroom. Their contribution to this issue is elucidating how even young students can be engaged in the practice of evaluating computer models on a mathematical basis.

The next three articles highlight computer modeling and simulation as a productive approach to integrating CT within STEM classrooms. They demonstrate how computer modeling and simulation can be integrated into K-12 science classrooms to support student learning of core science concepts and increase exposure to CS while preparing students for professional STEM practices. The ability to teach computer modeling and simulation in K-12 has been made possible by the availability of age-appropriate modeling and simulation tools such as StarLogo Nova and NetLogo. Researchers such as Moursund (2009) suggested that the underlying idea in CT is formulating and developing models and simulation of phenomena and problems that one is trying to study and solve. CT, when defined as a thinking process necessary when developing computer models, is a key component of modern scientific practice (PITAC 2005). CT is used by modern scientists as they engage in making models and running experiments using computer models for the purpose of conducting fundamental research (Gilbert 1991; Schwarz and White 2005). Scientists' ability to simulate the natural or designed phenomena has generated a great expansion of scientific knowledge (Emmott and Rison 2006).

Waterman, Goldsmith, and Pasquale (this issue) contribute a practitioner report on a CT integration module (iMOD) developed for 3rd grade students. They describe developing the module and characterized CT integration within school classes using a framework that spans three levels: exist, enhance, extend. A rich account is given of a CT-integrated activity in ecosystem science as well as utterances that evidence student learning that aligns with Massachusetts Department of Elementary & Secondary Education CT concepts and the computational thinking integration elements described in the first article (Lee and Malyn-Smith, this issue). Recommendations on designing extensions to bridge between K-12 disciplinary concepts and professional practices are provided as a roadmap for practitioners.

Aksit et al. (this issue) describe a study on exploring force and motion concepts in middle grades using computational modeling. This article extends the earlier articles by Dickes et al. and Moore et al. to examine representations and representational fluency for an older age range. The authors proposed the frame of "representational competence" to describe the thinking skills that may prepare students for the professional practice of "innovating with representations." In the context of modeling activities, the authors described students constructing, modifying, and experimenting with a model of force and motion. They identified instances of students making key connections between coded mechanisms and physics concepts. For example, they stated that "In order to formulate an algorithm [e.g. to position a falling object], students needed to conceptually understand what is meant by acceleration." Further, they described how multiple representations of the concepts as conceptual (diagram); algorithmic (mathematized version); code (encoded); and simulation run (visualization) assisted students in understanding the physics concepts. While the mathematical sophistication necessary to encode the formula to position an object was higher in this study when compared to that in Dickes et al. and Moore et al., the fluency needed to shift between different representations is similar.

The next article by Hutchins et al. (this issue) presents a comprehensive approach to integrating STEM and CT using the affordances of C2STEM, a computer-based learning environment that uses the NetBlox extension of Snap*!* Block-based programming. Hutchins et al. discussed C2STEM's negotiation of the competing interests in computing and science instruction in high school physics classrooms. They investigated "When are students learning or applying concepts or practices of physics?," " When are students learning or applying CT concepts or practices?," and "When is their learning of physics and CT mutually reinforcing?" They identified three types of synergistic learning moments when the learning of physics and CT were mutually reinforcing: (1) using a simulation to test a conceptual issue; (2) debating whether the model should capture the mechanism of the phenomenon; and (3) debugging a program that is not behaving in the expected/intended way. For all three types of synergistic learning, students encountered such moments while pursuing emergent goals as they engaged with model-building challenges within the C2STEM environment and curriculum. An example of an emergent goal was debugging a model that was not behaving in the intended/expected way. They found debugging often invoked students' conceptual understanding of physics (e.g., object is not accelerating when it should) through engagement in the practices of physics (e.g., refining a model based on observational evidence), all while debugging the model code, a key computational thinking skill.

These three articles show the breadth and depth of engagement in computer modeling and simulation practices in grades 3 through 12. Approaches to integrating CT that are productive—in that they lead up to professional modeling and simulation practices, are provided. Waterman et al. demonstrated how CT can be supported through participatory simulations and data studies without interrogating a model's encoded mechanisms. Aksit et al. described the skill of shifting between representations such as diagrams, algorithms, code, and visualizations, as key to gaining an understanding of physics concepts. Their article suggests that further study may be warranted on the connection between representational fluency and the innovating with representations seen in professional CT-enabled STEM practice. Hutchins et al.'s article contributes a clear example of when the learning of content area concepts and CT are mutually reinforcing in authentic practices (i.e., age-appropriate versions of professional practices in computer modeling and simulation). The article suggests a direction for further study: examining whether these three variants of mutually reinforcing learning episodes are universally applicable in STEM.

The following article by Pierson et al. (2019, this issue) presages the future in which CT extends to encompass dialogic relationships between humans and intelligent agents. It considers motivation and identity issues that arise when co-constructing inquiry with non-human partners. Pierson framed interactions with computer models as dialogic exchanges with co-participants then identified this type of interaction as a productive practice for disciplinary engagement in science and for computational thinking (Chandrasekharan and Nersessian 2015; Dennett 1989; Latour 1993; Pickering 1995). She found that computational models have unique affordances for dialogic interaction because they are probabilistic and iteratively executable. These features of computational models provided an entry point for students to adopt stances that treat computational models as conversational peers, co-constructors of lines of inquiry, and projections of students' agency and identity. Pierson argued that students' treatment of models as conversational peers parallels scientists' interactions with non-human entities, which often involve treating tools as agentive participants in inquiry (Latour 1999; Pickering 1995). Pierson argued that taking a computational entity as an interlocutor and co-participant in investigations is an important facet of computational thinking. She suggested that since students' enactment of a "dance of agency" acts as a precursor of legitimate disciplinary ways of operating with a scientific apparatus in conducting investigations. Thus, interactions with non-human collaborators may extend the five computational thinking integration elements in Lee and Malyn-Smith (this issue) as it affords students a pathway to practices at the intersection of disciplinary engagement and computational thinking. The co-editors of this special issue posit that this type of relationship will become ubiquitous in artificial intelligence and machine learning. In these fields, humans will interact with machine intelligences in conversational, creative, and agentive ways at the Human-Technology Frontier.

The next three articles describe studies that involve assessing CT integration from various perspectives. When considering "disciplinary" as "referring to K-12 subject areas," assessment of integrated CT addresses the questions "what are students learning about X?" and "what are students learning about CT." Furthermore, researchers ask "does the integration of CT in X improve learning of X?" and "does the integration of CT in X improve learning of CS?" Answers to these questions may help to make the case for the integration of CT in K-12 subject areas. Since these topics and skills are not yet tested in standardized assessments, their value is unknown and measurements do not exist to assess student learning or teaching. But when considering "disciplinary" to mean newer CT-integrated disciplines such as computational sciences, seeking learning gains in traditional science and CS is not the point. Instead, the objective is for students to be able to formulate questions in a domain then design, develop, and use computational tools to answer those questions. Thus, there is a tension within school day integration of CT—the "integrated learning goal" (formulating questions then using computational tools and techniques to answer those questions) may be in conflict with the goals of a STEM classroom that has to prepare students for traditional assessments in a STEM domain (AP exams, state tests); and unless traditional assessment in STEM is changed, educators are unlikely to abandon the domain learning view in favor of the integrated learning view.

Bortz et al. (this issue) argued that assessing integrated domains in K-12 education is problematic and difficult. The authors note intrinsic problems such as relying on rubric-based approaches that are insensitive to small movement toward incompletely mastered skills, and interactions between domains that may invalidate the results of the assessment either because gains in one domain may be easier to measure at certain times than the gains in the other, or because interactions between the domains may cause measurement interference. Instead, they argue that differentiated scoring methods should be employed to detect learning in each domain plus the "interplay" of domains. Based on their experience implementing and assessing a high school CT-integrated chemistry module, the authors suggest using assessments that enable students to express multiple ways of knowing and doing. Open-ended questions are considered as a form of assessment that may capture students' conceptions via multiple representations and uncover alternative conceptions, but the authors note the challenges and impracticality of using this type of measurement in regular day classrooms. The authors conclude that assessing multiple facets from each domain is necessary until we have a deeper understanding of the interplay of elements in a CT-integrated learning experience.

In the next article, Arastoopour et al. (this issue) propose an assessment approach combining an automated assessment tool, embedded assessments, and a pre-post survey. The authors used this assessment approach for a ten-day high school biology unit with CT activities. They identified students with both positive and negative gains and examined how each group's CT practices developed as they engaged with the curricular unit. The automated assessment tool is based on epistemic network analysis or ENA (Shaffer et al. 2016; Shaffer et al. 2009; Shaffer and Ruis 2017). It is a novel discourse analysis tool that creates a centroid representation of discourse networks. Discourse elements were identified based on a thematic analysis that identified common discourse elements in the student data and based on whether such discourse elements related to existing practices in the CT-STEM taxonomy (Weintrop et al. 2016). To measure connections among cognitive elements, the nodes in the network represent an individual's knowledge and skills identified in discourse and the links represent the individual's associations between knowledge. The links were analytically determined when elements co-occur in the discourse. The development of the network representation was based on research that shows that co-occurrences of concepts in a given segment of discourse data are good indicators of cognitive connections (Arastoopour et al. 2016; Lund and Burgess 1996). ENA measures when and how often learners make links between domain-relevant elements during their work. Their results showed that (1) students exhibited both science and computational learning gains after engaging with a science unit with computational models and (2) the use of embedded assessments and discourse analytics tools reveals how students think differently with computational tools throughout the unit. The authors suggest that discourse learning analytics might help teachers identify a student's struggles with understanding particular mechanisms. Such automated assessments may address some of the concerns raised in Bortz et al. (this issue) about the impracticality of using open-ended assessments in typical classroom settings.

In the third article on assessment, Hadad et al. (this issue) describe informal formative assessment of CT in makerspaces. Makerspaces are arenas in which students practice as active learners working directly with materials as if they were professionals in the field. Thus, these spaces have the potential to provide the opportunity to assess students' ability to formulate questions and design solutions in a domain then develop and use CT and tools to answer those questions or implement their design. The authors found that students used multiple representations of their construction: the artifact itself and diagrams were used as "objects to think with" (Papert 1980). In one case, students led by the mentor discussed levels of abstraction in the diagram and what needed to be represented in the diagram for it to be useful as a thinking tool. Using a framework from Csizmadia (citation), the authors describe individual "moments of notice" or instances when formative assessment could guide students' understanding of CT. This played out as an incident in which a program mentor finds an opening to query students about their project, their intent, and/or their reflections on the project or process. They identified four approaches to formatively assessing students' project work that surface their CT: using materials and CT terms; drawing or sketching for understanding; debugging practice; and fluidity of roles. This article contributes to the set by describing how instructors can use formative assessment to uncover students' prior knowledge and improve their use of CT in constructionist and project-based settings.

The final paper in the special issue focuses on teacher education and professional development (PD). Teacher preparation has been noted as a critical factor in the integration of CT into K-12 subject areas (Barr and Stephenson 2011; Voogt et al. 2015; Yadav et al. 2016; Yadav et al. 2017). Teacher PD programs focusing on CT integration often include key features of effective PD known from education research literature. To be effective, PD should be grounded in teachers' needs and their work environments, and address core areas of teaching: content, curriculum, instruction, and assessment. Common features of effective teacher PD programs include opportunities to (a) gain new knowledge, (b) reflect on changes in teaching practice, and (c) increase abilities and skills.

Additionally, effective PD focuses on student learning outcomes and models learner-centered instruction such that teachers experience and reflect upon learning activities that they will ultimately lead for their students (Hassel 1999; Gaible and Burns 2005). In addition to what is known about PD generally, Yadav et al. (2016) suggested that effective PD in CT should be tied to teachers' curricular needs, explicitly describe overlaps between learning objectives in the subject area and CT, promote the development of a community of practice among teachers, and be continuous rather than episodic.

Ketelhut et al. (2019, this issue) examined teacher change following a PD experience in integrating CT into elementary school science. The authors found that teachers' views of CT developed across personal, domain-specific practice, and outcomes dimensions. Teachers communicated their own knowledge, beliefs, and attitudes about CT integration (personal domain), they designed lessons to integrate CT into instruction (domain of practice), and they described outcomes of CT-infused learning for their students (domain of consequence). The authors then synthesized across the case studies and general findings, looking for common themes that arose out of the relationships between these domains. Across the personal and practice domains, they found teachers did not feel they had adequate knowledge and skills to troubleshoot students' problems or issues in open-ended inquiry-based projects, and teachers' developing notions of CT caused them to overestimate their claims of integrating CT. Additionally, though the relevance of computation was questioned, teachers found that efforts to integrate CT into their classrooms provided a forum for teachers to focus deeply on providing "good" science instruction. In the outcome domain, teachers grew to believe that integrating computation engaged student learners of all backgrounds. Numerous difficulties were encountered during the teacher PD program aimed at supporting 1st–2nd grade teachers for integration of CT in ES classrooms. The researchers found that teachers struggled with how CT best fit within their curricula, how to champion CT in school environments unfamiliar with CT, and how to find the resources and support they needed to enact their ideas for CT integration. This article brings an important question to the forefront: how deeply do teachers need to understand and master CT in order to help their students develop CT capacity and skills?

Together, this set of articles provides a picture of the current status of CT integration in STEM subjects from a disciplinary perspective as educators and researchers explore what it takes to prepare today's students for work at the Human-Technology Frontier. We believe that this special issue's presentation of a framework for computational thinking from a disciplinary perspective connects the interesting and productive research themes and compelling classroom-grounded strategies described in this issue to the foundational skills, knowledge, and dispositions needed by computational thinking scientists and engineers. Together, these advance our understanding of STEM+CT integration in K-12 learning, and we hope that it will provoke new ways of thinking about CT integration and generate discussion among the community of scholars, STEM teachers, and other STEM professionals.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

Aho, A. (2012). Computation and computational thinking. *The Computer Journal, 55*, 832–835. https://doi.org/10.1093/comjnl/bxs074 https://ubiquity.acm.org/article.cfm?id=1922682.

Arastoopour, G., Shaffer, D., Swiecki, Z., Ruis, A. R., & Chesler, N. C. (2016). Teaching and assessing engineering design thinking with virtual internships and epistemic network analysis. *International Journal of Engineering Education, 32*(3B), 1492–1501.

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.

Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2016). Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis. *International Journal of Mathematical Education in Science and Technology, 48*(3), 393–427. https://doi.org/10.1080/0020739X.2016.1254296.

Chandrasekharan, S., & Nersessian, N. (2015). Building cognition: the construction of computational representations for scientific discovery. *Cognitive Science A Multidisciplinary Journal, 39*(8), 1727–1763.

Cuny, J., Snyder, L., and Wing, J. (2010). Computational thinking: a definition. Retrieved from www.cs.cmu.edu › ~CompThink › resources › TheLinkWing on 10-14-19.

Dennett, D. C. (1989). *The intentional stance*. Cambridge: MIT press.

diSessa, A. A. (2001). *Changing minds: computers, learning, and literacy.* Cambridge: MIT Press.

Emmott, S. & Rison, S. (2006). *Towards 2020 science*. Microsoft Research. Downloaded at http://research.microsoft.com/towards2020science/downloads/T2020S_Report.pdf>. Accessed 11/19.

Feng, J., Spence, I., & Pratt, J. (2007). *Playing an action video game reduces gender differences in spatial cognition*. Psychological Science: A Journal of the American Psychological Society / APS. https://doi.org/10.1111/j.1467-9280.2007.01990.x.

Gaible, E. and Burns, M. (2005). Using technology to train teachers [Online]. Available from infoDEV: http://www.infodev.org/en/Publication.13.html (Accessed 4/12).

Gilbert, S. W. (1991). Model building and a definition of science. *Journal of Research in Science Teaching, 28*(1), 73–79.

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. In S. Sentance, E. Barendsen, & S. Carsten (Eds.), *Computer science education: perspectives on teaching and learning in school* (pp. 19–37). London: Bloomsbury Academic.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199–237.

Hassel, E. (1999). *Professional development: learning from the best*. Oak Brook, Illinois: North Central Regional Educational Laboratory.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science, 9*, 522–531.

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, R. (2019). Teacher Change Following a Professional Development Experience in Integrating Computational Thinking into Elementary Science. *Journal of Science Education and Technology*, https://doi.org/10.1007/s10956-019-09798-4.

Latour, B. (1993). Pasteur on lactic acid yeast: a partial semiotic analysis. In *Configurations, 1.1* (pp. 129–146). Baltimore: Johns Hopkins University Press.

Latour, B. (1999). *Pandora's hope: essays on the reality of science studies*. Cambridge, Mass: Harvard University Press.

Lee, I. (2016). Reclaiming the roots of CT. *CSTA Voice-Special Issue on Computational Thinking, 12*(1), 3–5.

Lehrer, R., Schauble, L., Strom, D., & Pligge, M. (2001). Similarity of form and substance: modeling material kind. *Cognition and instruction: Twenty-five years of progress*, 39–74.

Lesh, R., & Doerr, H. M. (2003). Foundations of a models and modeling perspective on mathematics teaching, learning, and problem solving. In R. Lesh & H. M. Doerr (Eds.), *Beyond constructivism: models and modeling perspectives on mathematics problem solving, learning, and teaching* (pp. 3–34). Mahwah, NJ: Erlbaum.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers, 28*(2), 203–208.

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. https://doi.org/10.1016/j.chb.2014.09.012.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. Portland, OR, USA.

Malyn-Smith, J., Lee, I. A., Martin, F., Grover, S., Evans, M. A., & Pillai, S. (2018). Developing a framework for computational thinking from a disciplinary perspective. In Proceedings of the International Conference on Computational Thinking Education 2018. Hong Kong: The Education University of Hong Kong.

Martin, F. (2018). Rethinking computational thinking. *CSTA - The Advocate.*, (Feb. 17, 2018).

Moursund, D. (2009). Computational thinking. IAE-pedia. Available online at <http://iaepedia.org/Computational_Thinking >. Accessed August 8, 2010.

NGSS Lead States. (2013). *Next Generation Science Standards: for states, 793 by states*. Washington, DC: The National Academies Press.

Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing computational thinking into high school mathematics and science classrooms. *Proceedings of ICLS 2016* (pp. 705–712). Singapore. Retrieved from http://ccl.northwestern.edu/2016/Orton_et_al_ICLS_2016.pdf

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York, NY: Basic Books.

Pickering, A. (1995). *The mangle of practice: time, agency, and science*. Chicago: University of Chicago Press.

Pierson, A.E., Brady, C.E. & Clark, D.B. (2019). Balancing the Environment: Computational Models as Interactive Participants in a STEM Classroom. *Journal of Science Education and Technology*, https://doi.org/10.1007/s10956-019-09797-5.

President's Information Technology Advisory Committee (PITAC). (2005). Computational science: insuring America's competitiveness. Washington, DC: National Coordination Office for Information Technology Research and Development. Retrieved from https://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf

Schwarz, C. V., & White, B. Y. (2005). Metamodeling knowledge: developing students' understanding of scientific modeling. *Cognition and instruction, 23*(2), 165–205.

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Owens, B. B., Stephenson, C., & Verno, A. (2011). *CSTA K–12 computer science standards: revised 2011*. ACM, New York, NY, USA: Technical Report.

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*, 351–380. https://doi.org/10.1007/s10639-012-9240-x.

Shaffer, D. W., & Ruis, A. R. (2017). Epistemic network analysis: a worked example of theory-based learning analytics. In *Handbook of learning analytics and educational data mining*.

Shaffer, D. W., Hatfield, D., Svarovsky, G., Nash, P., Nulty, A., Bagley, E. A., et al. (2009). Epistemic network analysis: a prototype for 21st century assessment of learning. *The International Journal of Learning and Media, 1*(1), 1–21.

Shaffer, D. W., Collier, W., & Ruis, A. R. (2016). A tutorial on epistemic network analysis: analyzing the structure of connections in cognitive, social, and interaction data. *Journal of Learning Analytics, 3*(3), 9–45.

Sherin, B. L. (2001). A comparison of programming languages and algebraic notation as expressive languages for physics. *International Journal of Computers for Mathematical Learning, 6*(1), 1–61. https://doi.org/10.1023/A:1011434026437.

Sherin, B., diSessa, A. A., & Hammer, D. (1993). Dynaturtle revisited: learning physics through collaborative design of a computer model. *Interactive Learning Environments, 3*(2), 91–118.

Stanton, J., Goldsmith, L., Adrion, W. R., Dunton, S., Hendrickson, K. A., Peterfreund, A., Yongpradit, P., Zarch, R., & Zinth, J. D. (2017). *State of the states landscape report: state-level policies supporting equitable K–12 computer science education*. Waltham, MA: Education Development Center, Inc. http://www.edc.org/state-states-landscape-report-state-level-policies-supporting-equitable-k-12-computer-science.

Tarkan, S., Sazawal, V., Druin, A., Golub, E., Bonsignore, E. M., Walsh, G., & Atrash, Z. (2010). Toque: designing a cooking-based programming language for and with children. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2417–2426). ACM.

Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013, March). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 609–614). ACM.

Uttal, D. H., Miller, D. I., & Newcombe, N. S. (2013). Exploring and enhancing spatial thinking: links to achievement in science, technology, engineering, and mathematics? *Current Directions in Psychological Science*. https://doi.org/10.1177/0963721413484756.

Uzzo, S., & Chen, R. (2015). Integrating computational thinking and environmental science: design based research on using simulated ecosystems to improve students understanding of complex system behavior. Retrieved from http://www.nsf.gov/awardsearch/showAward?AWD_ID=1543144

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715–728.

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.

Wilensky, U., Brady, C., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM, 57*(8), 17–21.

Wilkerson, M., & Fenwick, M. (2016). The practice of using mathematics and computational thinking. In C. V. Schwarz, C. Passmore, & B. J. Reiser (Eds.), *Helping students make sense of the world using next generation science and engineering practices*. Arlington, VA: National Science Teachers' Association Press.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends, 60*(6), 565–568.

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In M. Mulder (Ed.), *Competence-based vocational and professional education* (pp. 1051–1067). Cham, Switzerland: Springer.