

COMMENTARY

Open Access



The computational thinking for science (CT-S) framework: operationalizing CT-S for K–12 science education researchers and educators

Timothy Hurt^{*} , Eric Greenwald , Sara Allan , Matthew A. Cannady , Ari Krakowski , Lauren Brodsky, Melissa A. Collins , Ryan Montgomery  and Rena Dorph 

Abstract

Contemporary science is a field that is becoming increasingly computational. Today's scientists not only leverage computational tools to conduct their investigations, they often must contribute to the design of the computational tools for their specific research. From a science education perspective, for students to learn authentic science practices, students must learn to use the tools of the trade. This necessity in science education has shaped recent K–12 science standards including the Next Generation Science Standards, which explicitly mention the use of computational tools and simulations. These standards, in particular, have gone further and mandated that *computational thinking* be taught and leveraged as a practice of science. While computational thinking is not a new term, its inclusion in K–12 science standards has led to confusion about what the term means in the context of science learning and to questions about how to differentiate computational thinking from other commonly taught cognitive skills in science like problem-solving, mathematical reasoning, and critical thinking. In this paper, we propose a definition of *computational thinking for science* (CT-S) and a framework for its operationalization in K–12 science education. We situate our definition and framework in Activity Theory, from the learning sciences, in order to position computational thinking as an input to and outcome of science learning that is mediated by computational tools.

Keywords Computational thinking, Computational thinking for science, Activity Theory, K–12 science education

Introduction

Computation has become critical to an ever-broadening list of disciplines, particularly within STEM (science, technology, engineering, and mathematics) fields (Kaczmarczyk & Dopplick, 2014). Computational tools have long been employed in science to conduct research with greater precision, accuracy, and efficiency than would otherwise be possible. These tools have also enabled new modes of investigation, analysis, and explanation (Grover & Pea, 2013; Wing, 2010). Further, advances in

computation have led to fundamental shifts in how scientific research is conducted, with computational tools expanding the epistemological problem space of scientific inquiry, enabling scientists to investigate “grand challenges” in science (Denning, 2017; Foster, 2006; Wilson, 1989). As new technologies are developed, new applications of those technologies lead to new questions for investigation (Grover & Pea, 2013).

The importance of computational tools in science is not limited to adult practitioners—such tools are becoming increasingly common in science classrooms. The use of computational tools has been shown to support the learning of science content (Aksit & Wiebe, 2020; Dicks et al., 2016; Hutchins et al., 2020; Malone et al., 2018; Peel et al., 2019; Sengupta et al., 2013) and to help students understand modern scientific practices (Foster,

*Correspondence:

Timothy Hurt
thurt@berkeley.edu
Lawrence Hall of Science, University of California, Berkeley, USA



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

2006; Malyn-Smith et al., 2018; Weintrop et al., 2016; Wiese & Linn, 2021). Increasingly, contemporary scientific practices also include the design and evaluation of computational tools in the service of achieving science goals. To use, design, and evaluate these tools effectively for science, students must have some understanding of computation, as well as how computation can be leveraged to support a science goal (Grover & Pea, 2013). To wit, the field of computational science emerged from this increasing need for the power of computation to advance science—scientists in that nascent field used the term *computational thinking* to refer to the thought processes involved in using, designing, and evaluating computational tools (Denning, 2017). As computational science becomes increasingly descriptive of the practice of modern science in general, it follows that computational thinking for science is critical for science educators, learning designers, and assessment designers to attend to. To do this requires that these stakeholders understand the types of science learning experiences in which computational thinking is likely to occur (i.e., *where* that thinking is likely to occur) and *how* those experiences can engage students in such thinking.

Our paper addresses this need by presenting the *computational thinking for science* (CT-S) framework. For researchers, this framework can be used to (a) identify science learning experiences that are likely to elicit computational thinking so that it can be investigated during those experiences and (b) inform the development of assessments aimed at revealing CT-S so that it can be measured. For designers of science learning experiences and K–12 science educators, the CT-S framework can support the development of learning experiences at the intersection of computational thinking and science.

Efforts at defining and operationalizing computational thinking

The CT-S framework and definition presented here were informed by our review of two interrelated bodies of literature. First, we discuss prior efforts to define computational thinking and the cognitive processes that typify it. We then describe efforts to articulate the types of practices that engage students in computational thinking and work that has been done to integrate those practices into K–12 subjects. The framework and definition presented here build on and extend these prior efforts by articulating the cognitive processes that are likely to occur in science classrooms that are characteristic of CT-S.

Defining computational thinking

The term *computational thinking* was introduced in 1980 by Papert in a discussion of the potential impacts of computers on the way people think and learn. He suggested

that interactions with technology may actually contribute to the development of new types of mental processes (Papert, 1980). His perspective informed the field's current understanding of computational thinking as a cognitive process—not an application of knowledge or of a technique (Grover & Pea, 2013; Li et al., 2020; Selby & Woollard, 2013). Accordingly, a viable definition of computational thinking must center on the concept of cognitive processes. A widely employed definition that meets this criterion was proposed by Cuny, Snyder, and Wing in 2010: “Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (p. 1). Importantly for our work, which focuses on middle school science students who may not know how to program, Wing (2010) emphasizes that engaging in computational thinking does not inherently require programming knowledge. In fact, it does not require that computers be used at all—there are many computational thinking learning experiences that employ *unplugged*, or non-digital, pedagogical approaches (Bell et al., 2012; Folk et al., 2015; Huang & Looi, 2021; Lealdino Filho & Mercat, 2018; Peel et al., 2022). These unplugged approaches engage students in the design and use of computational tools—like algorithms—using paper, or even their own bodies, to learn about computation in an off-line manner that still aligns with the Cuny–Snyder–Wing definition.

In cases where students are engaged in learning experiences within computer science, the Cuny–Snyder–Wing definition, and variations of it (e.g., definitions proposed in Aho, 2011 and Nardelli, 2019)—which center on problem-solving with an information-processing agent—can be operationalized due to their precision (Krugel & Hubwieser, 2018; Lowe & Brophy, 2017). But when computational thinking is considered in a science classroom context, scholars, such as Denning (2017), have argued that these definitions lack sufficient clarity to position educators to teach or assess computational thinking in their specific context (see also Guzdial, 2015; Kite & Park, 2020). In recent years, efforts to further synthesize the literature on computational thinking have led to diverse results, and there remains a lack of consensus in the field about how to define and distinguish computational thinking from other cognitive processes like logical reasoning and critical thinking (Brennan & Resnick, 2012; Grover & Pea, 2013; Rutstein et al., 2014; Ulrich Hoppe & Werneburg, 2019).

Interacting with computers and other computational tools, even those that are unplugged, can engender critical thinking about the tool itself, and these cognitive processes are distinct from thinking about the problem or

goal for which the tool is being employed (Ackermann, 1996; Jonassen, 2000; Jonassen & Rohrer-Murphy, 1999; Kaptelinin & Nardi, 2006; Kuutti, 1995). These types of cognitive processes, emanating from and focused on interactions with computational tools, extend beyond the cognitive processes implied by definitions of computational thinking that center on formulating instructions *for delivery* to an information-processing agent. Whereas “delivering instructions” involves a transfer of information from the thinker to the tool—“interactions with a computational tool” elicit a bidirectional exchange of information between the thinker and the tool (Csizmadia et al., 2019; Nardi, 1996; Solvie & Kloek, 2007). The cognitive processes involved in interactions with computational tools are an important aspect of computational thinking as originally conceptualized by Papert and are crucial for CT-S. Namely, to prepare students for the increasingly computational nature of science, it is essential to develop students’ abilities to think about the functionality and positionality of computational tools within an activity—particularly in science learning contexts (Ah-Nam & Osman, 2017; Grover & Pea, 2013; Sengupta et al., 2013; Shaffer & Clinton, 2006; Weintrop et al., 2016).

The CT-S framework and definition presented here were designed to account for the bidirectional exchange of information between the thinker and the tool, and the different types of computational tool interactions that can elicit computational thinking in science learning contexts. We ground our work in Activity Theory, which we describe in the “[Theoretical Framework](#)” section, as an analytic heuristic that is particularly well suited to examining the bidirectional and mediating relationships across the learner, computational tool, and science goals to which the tool’s use is directed.

Computational thinking practices

In 2006, Wing reintroduced the term *computational thinking* into the national lexicon, aligning the term with problem-solving practices and methodologies frequently employed in the discipline of computer science. Wing argued that all students could benefit from learning how to think like a computer scientist. This led to a growing body of literature on ways to introduce computational thinking in K–12 education, especially by integrating it into core subjects (Grover et al., 2020; Jona et al., 2014; Lee et al., 2014; Settle et al., 2012; Yadav et al., 2016). Despite awareness of its importance, computational thinking has remained elusive to operationalize as its definitions have historically been ostensive, or defined through example, and often refer to methods and conventions commonly employed by computer scientists—for instance, “solving problems like a computer

scientist” or “approaching tasks like a programmer” (Barr & Stephenson, 2011; National Research Council, 2010). To operationalize these definitions, a list of the computer science practices associated with computational thinking was needed. In 2010, the National Research Council convened a group to study the scope and nature of computational thinking. A resulting report identified over 20 practices that computational thinking could include (National Research Council, 2010).

Although *computational thinking* is most often associated with computer science, such lists of practices associated with it fail to recognize that computational thinking is not synonymous with computer science, computer literacy, or programming (Bell et al., 2010; Brinda et al., 2009; Computing at School Working Group, 2012; Grover & Pea, 2013; Selby & Woollard, 2013). Conceptualizing computational thinking beyond computer science is essential for its inclusion in K–12 learning environments. There have been several notable efforts to identify the practices associated with computational thinking within K–12 core subjects. For example, Barr and Stephenson (2011) proposed a framework that illustrates how such concepts as *abstraction* and *parallelization* could be incorporated into many K–12 subject areas. Malyn-Smith et al. (2018) as well as Dong et al. (2019) have introduced frameworks to support educators in identifying opportunities to engage students in computational thinking within disciplinary learning.

Others have emphasized specific subject areas. For instance, Weintrop et al. (2016) and Lee and Malyn-Smith (2020) have proposed frameworks for integrating practices related to computational thinking into K–12 STEM subjects. Weintrop et al. (2016) created their *computational thinking in mathematics and science taxonomy* to be an “actionable set of guidelines that can be followed to bring computational thinking into mathematics and science classrooms quickly and effectively” (p. 129). While such frameworks have many affordances, there are limitations that arise from their focus on those activities that *can* motivate computational thinking without attending to *how* those activities engage students in such thinking. For example, many of these frameworks (e.g., K–12 Computer Science Framework, 2016; National Research Council, 2012; Weintrop et al., 2016) list *creating data visualizations* as a computational thinking practice; however, as articulated in the K–12 Computer Science Framework (2016), “a student is not necessarily using computational thinking when he or she enters data into a spreadsheet and creates a chart” (p. 70). That is, it is possible to create a data visualization with or without engaging in computational thinking (e.g., if a student merely follows a teacher’s step-by-step “recipe”), and without clarifying the cognitive processes characteristic

of computational thinking, it is difficult to evaluate computational thinking performance on a task in which it may be occurring (Weintrop et al., 2021). To hypothesize whether a student engaged in computational thinking while creating a data visualization would require a model of cognition (Brown & Wilson, 2011) that can account for the specific cognitive processes one would expect to find evidence of.

The CT-S framework and definition presented here describe a model of cognition for CT-S and thereby better enable assessment and learning designers to identify *how* performance tasks can engage students in *computational thinking for science*.

Theoretical framework

To construct a definition of CT-S that attends to the mediating role of a computational tool in cognition, we draw on Activity Theory. Developed by the Russian theorists Lev Vygotsky, Aleksei Leont'ev, and Alexander Luria, Activity Theory is a framework for investigating human behavior, understood as goal-directed activity in a specific sociocultural setting (Engeström, 2015; Vygotsky, 2012). Our research builds on prior work using Activity Theory to examine technology-centered learning (e.g., Barab et al., 2004; Blin, 2004; Brine & Franken, 2006; Issroff & Scanlon, 2002; Murphy & Rodriguez-Manzanares, 2008). Kaptelinin and Nardi (2006) argue that the use of Activity Theory directs attention away “from the computer as the focus of interest”, allowing for the examination of “technology as part of the larger scope of human activities” (p. 5). Activity Theory positions intelligence as being distributed throughout an activity system: knowledge and meaning-making emerge from a subject's interactions with tools and with others, rather than being created and held entirely by the subject (Pea, 1993). In using Activity Theory to frame our analysis, we understand computational thinking as situated in activity and distributed across the actors and tools involved in the activity system (Engeström, 2015; Greeno, 1998). The intrapersonal cognitive processes we aim to describe here is but a piece of that broader goal-directed activity, but one that is critical to understand and describe (Greeno, 2015).

The basic unit of analysis in modern Activity Theory is the *activity system*. This system is viewed from the perspective of a *subject*, a person or group who serves as the primary actor. The subject's efforts are motivated by and directed toward an *object*. Defined by Leont'ev (1978, as cited in Kaptelinin & Nardi, 2006) as an activity's “true motive” (p. 139), the object can be thought of as an *activity's goal*. Imagine, for example, a science teacher leading a lesson about Newton's law of universal gravitation. In this activity system, the science teacher is the subject.

The teacher's goal is to elucidate gravitational force. In Activity Theory, goals are transformed into *outcomes*—results of the activity. Engeström (2015) clarifies that outcomes may be out of the subject's direct control and can even be unintended. In our example, an outcome may be that the teacher successfully communicated ideas about gravitational force to students. An unintended outcome might be that the students developed misconceptions about gravity.

Drawing from prior work in child development, Vygotsky (1978) devised the concept of *mediation*, which has become central to Activity Theory. He observed that when humans interact with the environment, they do so indirectly through mediating artifacts, or tools. *Tools* can be material or immaterial (e.g., a hammer or a theoretical framework). For instance, the science teacher in our example may use their understanding of the zone of proximal development (theory as tool) to determine which ideas about gravity to communicate next to students. The teacher might also employ tangible items, such as basketballs, to demonstrate gravity. The teacher may likewise use symbols, gestures, or diagrams to illustrate concepts visually. Regardless of materiality, all these mediating artifacts are considered tools. Moreover, tools can, and often do, mediate an activity even if the subject is not consciously aware of their presence or intentionally employing them. In this example, the teacher's activity might be mediated by their learned instructional strategies, their own mental model of gravitational force, or their conceptions about student learning. Notably, tools do not need to be unique to the specific activity; many of the aforementioned tools could operate in other activity systems. The resulting outcome would be the product of this indirect interaction, or mediation, between the subject (the teacher) and the object (elucidating gravitational force) through the use of tools.

In addition to the components of an activity system, Activity Theory includes assumptions about the nature of relationships among components. Notably, in activity systems, a subject does not merely wield tools to accomplish a goal—a one-way direction of influence. Activity systems are dialectically structured, such that the system's components are mutually dependent and influence one another. When a subject is using a tool to work toward a goal, it is not just the goal or the tool that is influenced; the subject is affected as well. The subject may learn something new about the tool or come to new understandings about the goal. This meaning-making process may then inform the subject's decisions and actions. In turn, these decisions and actions affect the tool and the goal. This dialectical exchange repeats until an outcome is produced.

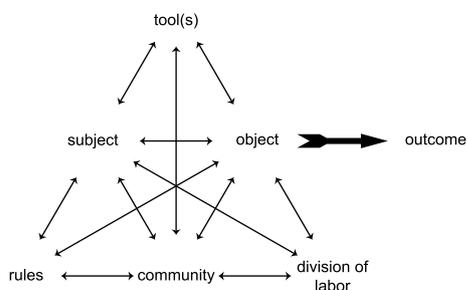


Fig. 1 A single-subject activity system

Modern revisions to Activity Theory have sought to broaden this core Vygotskian subject–tool–object activity system to better reflect the social situativity of an activity system. Along these lines, Engeström and others (see, e.g., Cole et al., 1997; Cole & Engeström, 1993) have drawn on the work of Leont’ev to advance cultural–historical activity theory (CHAT), which includes attention to the mediating role of the norms and rules of interaction, the community of stakeholders involved in activity, and the division of labor among actors within the activity. As its name implies, CHAT offers a framework to analyze the cultural–historical complexities of interpersonal, goal-mediated collective activity. Figure 1 depicts these relationships within a single-subject activity system.

Attention to this complexity has prompted further revision to Activity Theory to account for the tensions, contradictions, and synergies involved in multivoiced human social activity such as those in the workplace: we each bring our histories and our cultural norms to the activity, even as the goal may be shared. This third generation of Activity Theory includes, at a minimum, two intersecting activity systems working toward (ostensibly, at least) the same goal. This third generation is particularly well suited to analyze system-level interactions within and across organizations or social settings.

Our analysis of CT-S focuses on a subject’s interaction with a computational tool toward a goal of creating a mental model of that tool’s functionality with respect to its use in a science activity. For example, a learner (the subject) interacts with a simulation (the tool) to understand how to use the simulation to learn more about the real-world (the goal). Thus, we explore activity systems viewed from the perspective of a single subject as the unit of analysis. Moreover, because we share CHAT’s perspective that the social, cultural, and historical context through which a person acts will mediate that activity in important ways, we include these elements in our analysis and locate our examples in the context of classroom learning. Because the particulars

of those interactions are necessarily mediated by the cultural–historical context in which they occur, future applications of the proposed framework (e.g., in the design of learning experiences, research measures, and assessment, or how the cultural–historical development of a particular tool mediates the computational thinking involved in its use) would be incomplete without direct attention to the specific context(s) in which the framework is being applied.

Context

This section details the crucial processes and decisions that led to the CT-S framework we present. This work is part of a larger study to understand CT-S as both an input to and an outcome of science learning, and is situated within the Activation Lab, which offers a theoretical framework on *science learning activation* that accounts for both the proximal (near-term) and potential distal (long-term) outcomes of science learning that happens both in- and out-of-school (Dorph et al., 2016).

Our work was prompted by a need to identify or develop a conceptual framework that describes CT-S such that we could take an evidence-centered approach (Mislevy et al., 2003) to design a CT-S assessment for administration in middle school science classrooms. That is, we needed a model of cognition (Brown & Wilson, 2011) to ground our development of assessment items likely to elicit CT-S. As this assessment was not going to be designed as part of an instructional intervention, it was important that such a framework could identify the types of cognitive processes already being leveraged in science classrooms that reflect CT-S and not already accounted for in other models of cognition like critical thinking or scientific sensemaking. Such a framework would enable us to (a) delineate subconstructs that specify cognitive processes characteristic of CT-S and (b) operationalize CT-S subconstructs to develop performance tasks likely to elicit CT-S. A review of the extant work in this area revealed several frameworks that defined subconstructs related to computational thinking (Bienkowski et al., 2015; College Board, 2019; Google for Education, 2019; K–12 Computer Science Framework, 2016) as well as computational thinking in science (National Research Council, 2012; Weintrop et al., 2016). Though there was considerable overlap among the subconstructs in these frameworks, they lacked a model of cognition to describe the types of cognitive processes that typify CT-S, thus providing us insufficient traction for operationalizing the identification and measurement of CT-S. That is, beyond knowing which practices are likely to engage students in CT-S, we needed a testable model for *how* those practices engage students in CT-S. Further, we needed to identify which of those practices

CT-S		Cognitive Processes		
		Reflective Use	Design	Evaluation
		of a computational tool for		
Science Activity	Data Collection	How can I use the time-lapse photography feature on my phone to collect data on how fast my plants grow?	How could I change my robot's code so that it would randomly sample the quadrats for my ecological study?	If this pseudocode for getting the temperature sensor readings was running on my computer, under which conditions would it work well, and when might it fail?
	Data Processing	Which features in the spreadsheet software can help me identify the relationship that exists between force, mass, and acceleration?	I have images that show the sun's height above the horizon from every day of the year from a given location taken at solar noon; what would I need software to do to help me find patterns related to seasonal changes?	I am using software to create a data visualization of how the air quality in my city changed due to forest fires; what are the software's affordances and limitations for helping me communicate those data?
	Modeling	How can I use an orrery to predict the next transit of Venus?	What rules would I need to include in a plate tectonics simulation to accurately model different types of plate boundaries?	Which aspects of this digital model accurately reflect natural selection and which do not?
	Problem-Solving	I want to share my experimental procedure for measuring the speed of sound with a lab partner who is blind; how can I use their screen-reading software to make sure the procedure will be communicated correctly?	My classmates gave different arguments about climate change; how could I create an algorithm to evaluate the claims and reasoning of their respective arguments?	I need to test a phone app that was designed to call an emergency contact when it detects a car crash; how should I test it and how will I know I have sufficiently tested it?

Fig. 2 The computational thinking for science framework with examples

students are likely to engage in during standards-aligned middle school science learning experiences. Our team, with input and feedback from a panel of a dozen experts and advisers with relevant expertise (STEM educational research, learning design, computer science and computational thinking, and assessment design), underwent a synthesis and distillation process where we interrogated the existing subconstructs and their relationships in order to define a set that would meet the following criteria: each subconstruct should

- be distinct,
- pertain specifically to activities that could occur in science classrooms,
- specify the cognitive processes that constitute computational thinking.

Through this distillation process, our team, with the expert panel, conducted multiple cycles of interrogation to vet and iteratively refine the set of CT-S subconstructs. To make the resulting framework comprehensive, we worked to ensure that the subconstructs related to computational thinking in science described in prior work (listed above; e.g., Weintrop et al., 2016) could also be

located within this framework. While it is possible that relevant subconstructs may be missing from the CT-S framework, we posit that the framework, in its current form, is operationalizable for identifying learning experiences *where* computational thinking is likely to occur and *how* those learning experiences can engage students in computational thinking.

Computational thinking for science framework

Presented in Fig. 2, the CT-S framework is intended to identify—and delineate—the subconstructs that can be used to inform the design of instructional sequences and assessments that promote or measure CT-S learning, respectively.

The framework is a table of four rows and three columns, which creates 12 cells. The rows represent four categories of science activity (data collection, data processing, modeling, and problem-solving) where computational tools are likely to be leveraged in K–12 science learning.¹ The columns represent three interactions with

¹ While each of these activities occurs in domains other than science, our definition draws on prior work articulating the science discipline-specific instantiations of each activity.

computational tools (Reflective Use, Design, and Evaluation of computational tools) that engage the cognitive processes characteristic of computational thinking. Each cell within the framework, therefore, represents CT-S as the intersection of a row with a column. That is, any time an individual engages in a science learning experience that can be categorized by one, or more, of the cells in the framework, they are engaging in *computational thinking for science*. For each cell in Fig. 2, an example question is given where an individual would likely need to engage in CT-S to successfully answer that question. For example, the individual answering the question in the upper left-hand corner of the framework would need to engage in the Reflective Use of their phone (a computational tool) in order to work toward their *data collection* goal. These examples were chosen to illustrate the cells of the framework and are therefore not necessarily representative of the variety of computational tools, or disciplinary science content, that individuals could engage in that would likely prompt CT-S; notably, the majority of the computational tools described involve computers or digital technologies, but, as is described below, this is not a criterion for computational tools.

Defining computational thinking

The CT-S framework is built off a definition for computational thinking that centers on cognition that occurs during engagement with *computational tools*. From Activity Theory, an artifact is considered a tool when the subject uses it as they work toward a goal. If the subject uses a tool in a way that leverages its computational affordances, then the tool is deemed a *computational tool*. Anything that can compute, or carry out sequences of arithmetic or logical operations, automatically in accordance with a well-defined model (e.g., an algorithm) has computational affordances (e.g., digital and analog artifacts like calculators and slide rules, respectively, have computational affordances).

An artifact with computational affordances can be a noncomputational tool when someone uses it to work toward a goal without leveraging its computational affordances. For instance, using a calculator as a paperweight is still using the calculator as a tool but not as a *computational tool*. Whether a tool is computational in a given use depends on its functionality in that use—what it does as the subject interacts with it to work toward the goal in a given activity system.

With this understanding of a computational tool in mind, we offer the following definition: *computational thinking* is the cognitive processes involved in building or modifying a mental model of a computational tool's functionality.

Defining computational thinking for science

CT-S occurs when an individual engages in computational thinking for their science activity. In the subsections that follow, we present three hypothetical cases of CT-S to illustrate how a student can engage in each of the three cognitive processes (Reflective Use, Design, and Evaluation of a computational tool) during a science activity. We define these cognitive processes as follows:

- *Reflective Use of a computational tool*: building or modifying a mental model of that computational tool's functionality through interaction with that tool.²
- *Design of a computational tool*: building or modifying a mental model of an imagined computational tool's functionality.³
- *Evaluation of a computational tool*: building or modifying a mental model of the affordances and limitations of that computational tool's functionality.

These definitions are grounded in Activity Theory, which stipulates that cognition occurs through the interaction with tools toward a goal and that tools can be immaterial (e.g., the tool could be an individual's mental model of a computational tool). Each of the above definitions assumes that the cognitive processes are happening within a goal-directed activity and, in the case of CT-S, that the activity is necessarily a science activity (i.e., the row headers in Fig. 2).

Hypothetical cases of CT-S activity

Context for the following hypothetical cases

To illustrate how science activities and cognitive processes intersect for our definition of CT-S, we explore three hypothetical cases: *Reflective Use* of a computational tool for *data processing*, *Design* of a computational tool for *data collection*, and *Evaluation* of a computational tool for *modeling*. In each case, a student is engaged in a scientific investigation where they are investigating bacterial growth and their science goal is to better understand how bacterial populations change over time as part of their science class. While there are many different ways a student could conduct an investigation with this same goal, each example illustrates one way the student could engage in CT-S as they work toward their

² Note that the interaction does not need to be direct. That is, an individual could instruct their friend to interact with the tool, and so long as the friend communicates their actions and the computational tool's behavior, the individual could still be engaged in Reflective Use.

³ *Imagined*, in this instance, refers to the fact that the subject is generating a new-to-them mental model of a computational tool's functionality regardless of whether that tool's functionality currently exists in the world.

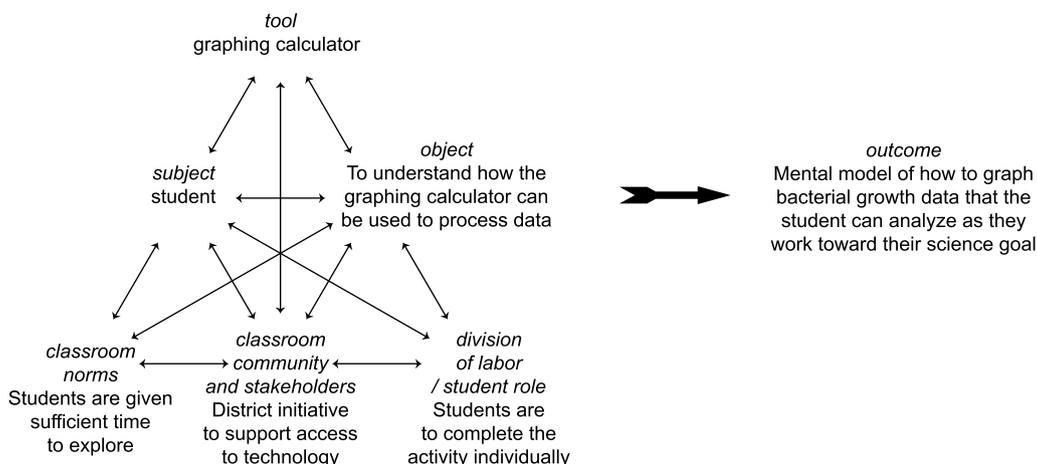


Fig. 3 Reflective use of a computational tool for data processing activity system

science goal. These examples narrowly focus on *how* the student engages in CT-S during their science activity. However, it is crucial to understand that their engagement in CT-S is only a step toward achieving their goal. To achieve their goal would likely involve other tools and cognitive processes than those discussed in the cases.

Reflective use of a computational tool for data processing

Imagine that each student in the class has been provided with a graphing calculator that has data on a bacterial population’s size at different times. The teacher has asked each student to use the calculator to help them identify any relationships between population size and time. This student, however, has never previously used a graphing calculator. Before they could use the graphing calculator to accomplish their science goal, they would need to figure out how to operate it and what computations it can do that may help them toward their science goal. They can do this by engaging in Reflective Use of the graphing calculator as they interact with it. This student begins their Reflective Use by manipulating the calculator, selectively pressing certain buttons, and observing the results of those actions. They then reflect on their manipulations and begin to form a mental model of the graphing calculator’s functionality. As the student continues to interact with this computational tool, their discoveries reinforce, revise, or supplement their developing mental model. In this way, Reflective Use is bidirectional in terms of information transfer: the student takes actions, reflects on what the computational tool does as a result of those actions, and then takes new actions based on the result of that reflection. Through this continued engagement in Reflective Use, the student builds a mental model of the graphing calculator’s functionality and how it can help them toward their science goal; therefore, the

student is engaged in CT-S. Once the student has a working mental model, they can use the graphing calculator more intentionally to process the bacterial growth data so that it is in a form that they can analyze to learn how the bacterial population changes over time. For example, the student could create a scatterplot using the calculator and identify that, over time, the bacterial growth rate increases, reaches a plateau, and then decreases. Figure 3 depicts the activity system described by this narrative where the student (subject) has built a mental model of how to graph bacterial growth data with the calculator (outcome) as they work to understand how a bacterial population changes over time. Importantly, the degree to which the student engaged in CT-S would differ if the activity system had been different—for instance, if the teacher does not typically provide much time for Reflective Use (classroom norms), students may need additional scaffolding to support such reflection (e.g., question stems or discourse routines), or they may not have adequate opportunity for meaningful engagement with this aspect of CT-S if their developing ideas are cut short by the need to move on to the next activity. Along these lines, if the district did not have an initiative to support access to technology (classroom community and stakeholders), classrooms may not have adequate availability of materials (e.g., too few working calculators for hands-on experiences), which would limit opportunities for students to engage in the iterative act/reflect cycles described above.

Reflective Use can also occur when the student starts their activity with an incomplete or inaccurate mental model of a computational tool’s functionality. For instance, if the student faces an unexpected output or error, they may engage in Reflective Use of the computational tool to reinvestigate and modify their

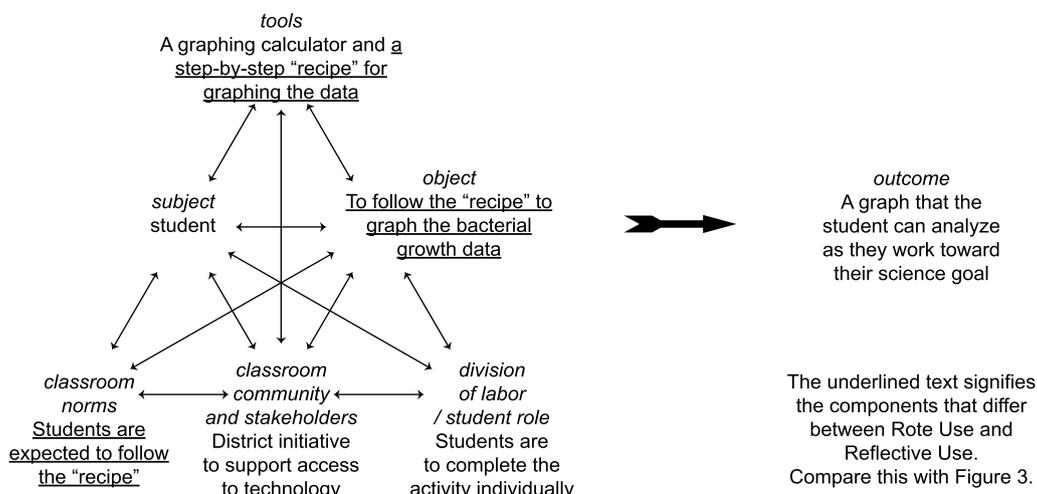


Fig. 4 Rote use of a computational tool for data processing activity system

mental model of its functionality. Additionally, Reflective Use can also occur even when the student is not *directly* interacting with the computational tool. If the student, after their science class, were to continue thinking about the calculator’s functionality by probing and modifying their mental model of it, then they would still be engaged in Reflective Use (e.g., “I bet if I had pressed that button, it would have given me a line of best fit, which could help me determine the growth rate”).

Reflective Use stands in contrast to *rote use* of a computational tool—wherein the student employs the graphing calculator without building or modifying a mental model of its functionality. For instance, imagine that the teacher had provided a step-by-step “recipe” for students to follow in order to graph the relevant data using the graphing calculator. Had the student merely followed that recipe it would be considered rote use, not Reflective Use. Figure 4 depicts the rote use activity system where at no point does the student (subject) build or modify their mental model of the graphing calculator’s functionality; the underlined text in the diagram identifies the key components that differ between this activity system and the Reflective Use activity system in Fig. 3.

Rote use of a computational tool does not mean that the student is inadequately or unsatisfactorily engaged in science or science learning; instead, it means that, for the given activity, the student did not engage in CT-S but likely did engage in other cognitive processes—like scientific sensemaking. In fact, scientists often engage in rote use of familiar tools—those with which they already have a sufficient mental model—enabling them

to focus more on the investigation at hand and less on the tool itself.

Design of a computational tool for data collection

Imagine that each student in the class has been given access to a petri dish, which has been inoculated with bacteria and placed in an incubator; students also have access to other equipment, including the classroom computers. The teacher has asked the students to work individually, or in pairs, to figure out how quickly the bacterial populations grow. This student realizes that to successfully complete this activity, they will need to collect data on the size of the bacterial populations at different times. They think that the populations might grow quickly—might go from being invisible to filling the petri dish over the course of one day. With this in mind, they decide that they will use a classroom computer and camera to collect data on the growth of the bacterial populations. The student engages in Design by envisioning a computational tool’s functionality that would enable them to take a picture of the petri dish every 10 min for 24 h: the student thinks about how the camera and the computer could be programmed to take pictures of the petri dish at 10-min intervals, and determines that as long as the computer saves all the images, with the corresponding timestamps, it will be possible to analyze the images to determine how a given bacterial population changes over time. As a result of this line of thinking, the student builds a mental model of this imagined computational tool’s functionality and how it could be leveraged in their science activity; thus, the student is engaged in CT-S. If the student were to go on to run this experiment, they would be able to determine relationships between the

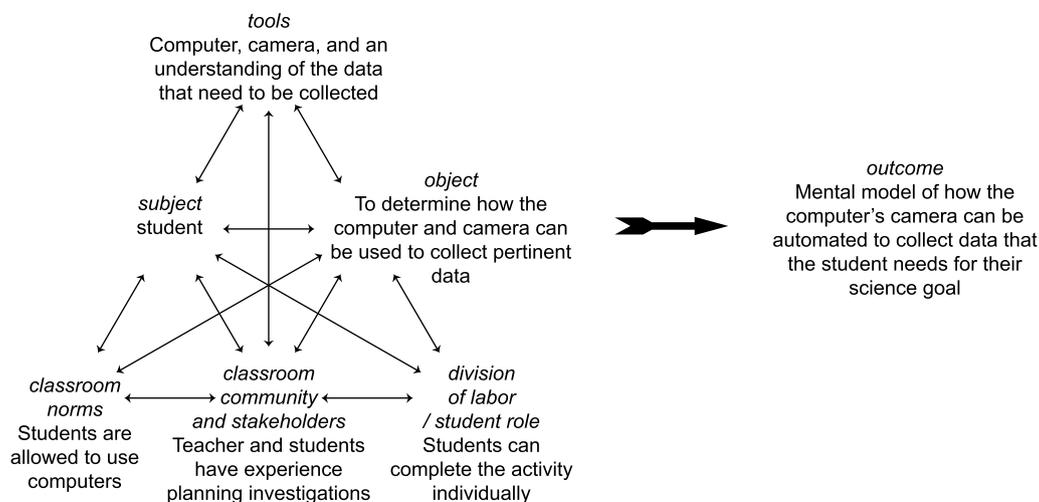


Fig. 5 Design of a computational tool for data collection activity system

bacterial population size and the amount of time it had to grow. Figure 5 depicts the activity system described by this narrative where the student (subject) has built a mental model of how to automate the computer's camera to take the pictures that they need (outcome) in order to learn how the bacterial populations grow over time. The student's CT-S engagement would differ if the activity system had been different—for instance, had there been an expectation that the student would have worked with a partner (division of labor/student role), each student's opportunity to engage in CT-S would depend on how partner roles were articulated and negotiated within the activity, potentially intersecting with long-standing power dynamics and historical inequities in STEM.

Notably, because computational thinking is a form of cognition, an individual can engage in Design *without* physically or digitally constructing their *imagined* computational tool's functionality. For example, the outcome of Design in the previous paragraph did not include the actual programming of the data collection device. Another important aspect of Design is that it need not only precede the creation of a computational tool's functionality or occur only once in a creation process. That is, Design can occur throughout an iterative creation process where the subject has to repeatedly update and modify their mental model of the computational tool's functionality relative to its intended use to support their goal.

Evaluation of a computational tool for modeling

Imagine that the teacher has divided the class into groups where each group will be working with a different bacterial growth simulation, and each group has been asked

to evaluate their simulation so that they can share out its affordances and limitations with the rest of the class. To engage in Evaluation, the students need to know what the simulation should do in different configurations to determine if it is a complete and accurate model. To do this, they conduct some research to determine what they should be comparing the simulation's results to. For instance, they find that bacterial growth curves tend to exhibit distinct phases depending on certain factors—like time elapsed, nutrient concentration, and species. Based on this research, the students determine that the simulation accurately models growth of a population that has unlimited resources, but, since the simulation does not include the ability to control resources, it is unable to model a growth curve for when nutrients are limited. As a result, each student would have built a mental model of the affordances and limitations of the computational tool's functionality and how it could be leveraged in their science activity. Thus, the students are engaged in CT-S. Having completed this Evaluation, the students could then determine how well a given simulation works for a given use case based on what they learn from the other groups in the class. Figure 6 depicts the activity system of one student in this group as described by this narrative where the student (subject) has built their mental model of the simulation to include its affordances and limitations (outcome) with respect to completely and accurately modeling growth of bacterial populations under different conditions. The degree to which the student engaged in CT-S would differ if the activity system had been different—for instance, had the composition of student groups been different (classroom community and stakeholders), there may be very different capacities to conduct the research needed to “ground truth” the model

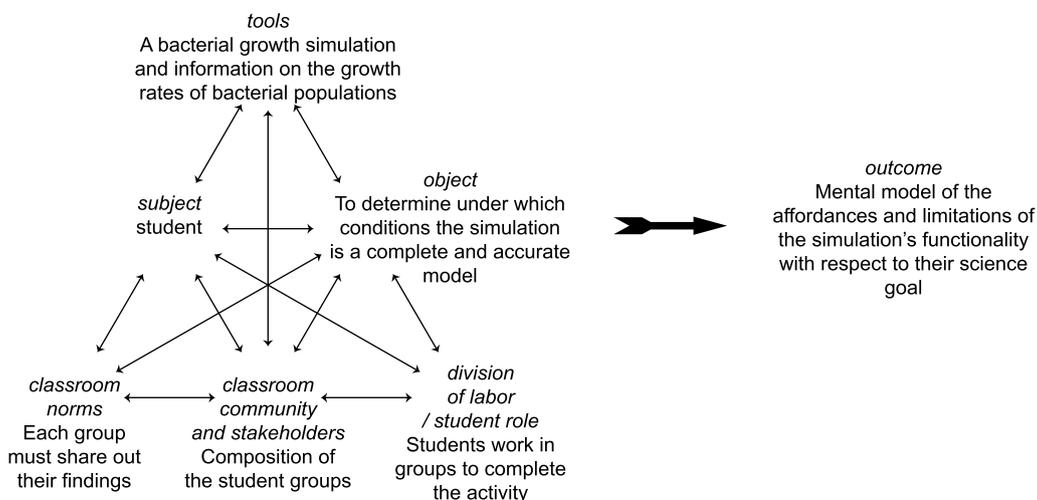


Fig. 6 Evaluation of a computational tool for modeling activity system

in order to evaluate the tool’s affordances and limitations. Along these lines, if the roles within each group (division of labor/student role) had been prescribed in ways that assign different cognitive tasks to individual group members (e.g., background researcher, evaluator, presenter), then opportunities to engage in thinking at the intersection of computational thinking and science to evaluate a computational tool may be limited or inequitably distributed across the group.

Discussion

With measurement development as the primary motivator of our work, the focus of our analysis has been on the individual contributions of the learner within activity: what are the cognitive resources brought to bear in tool-mediated, goal-oriented activity and how might those resources, those mental models, become visible and get revised through activity? In grounding our analysis in CHAT (Engeström, 1999), we position learning, and CT-S, as situated activities that require explicit attention to how the cultural–historical mediators of activity play a role in the cognitive processes we put forward. For example, we would expect CT-S to emerge differently in formal versus informal contexts, or where classroom norms emphasize individual competition versus collaboration, or in relation to the myriad ways in which society has positioned computer science, and computational technologies, as the domain of certain populations but not others. Applications of the proposed CT-S framework, therefore, and any work toward a full understanding of CT-S would be incomplete without attending to the complex situativity of learning represented in the rules, community, and divisions of labor within the activity system (Engeström, 1999).

Tool as object of scrutiny

Along these lines, the bidirectional relationship we propose between learner and tool draws attention to the computational tool as, itself, a sociocultural artifact: whether a mental model, an algorithm, or a software package, the tool is constructed, and it cannot be divorced from the situativity of its construction. We argue, then, that understanding how CT-S is developed or demonstrated in a particular learning context will require inspection of the tool qua artifact. Further, as we consider the role of the tool and its sociocultural development in mediating CT-S, we note that each of the hypothetical cases discussed in this paper relates to a technological tool—something broadly recognizable as a computational artifact. However, computational tools need not be computers or computer programs. In fact, the history of STEM is filled with non-digital computational tools that would serve an analogous role in a CT-S activity system: astrolabes, abacuses, mechanical harmonic analyzers, and so forth. These, and any number of physical manipulatives or mental models, could ground one’s engagement in CT-S. In this commitment, we stand firmly with the field’s growing consensus that computational thinking is not computer-bound. From a measurement development perspective, attention to the situativity of CT-S would mean careful consideration of not only the target set of cognitive processes to be demonstrated at a particular grade level but also the particular set of tools to be utilized and the articulation of the object-oriented context for their use: both how a defined set of cognitive processes may be elicited through use of a computational tool and how that tool-as-artifact is likely to mediate expression of CT-S within a given population and in a given context. Future research, then, could probe

particular tools' mediating roles within different CT-S activity systems—and examine the affordances, tensions, and contradictions that a particular tool may introduce for a learner's (measurable) expression of CT-S.

Opportunities and challenges for assessing CT-S

While our framework centrally locates the computational tool in expressions of CT-S, that engagement with such tools (material or conceptual) can take many forms. Here, we share Sengupta et al.'s (2018) call that “computing and computational thinking should be viewed as discursive, perspectival, material and embodied experiences, among others. These experiences include, but are not subsumed by, the use and production of computational abstractions” (p. 49). In fact, we would argue that such discursive, perspectival, material, and embodied experiences are reflected in our framework inasmuch as they are helping those engaged in the experiences build and revise their mental model of a computational tool's functionality. As in Sengupta et al.'s example, “Embodied modeling introduces the students to the relevant computational rules represented by the agent-based programming commands” (p. 59), or, in other words, as the students modeled the computational tool's functionality with their bodies, they built and modified their mental models. Thus, in designing CT-S assessment tasks, it will be important to consider multiple modes of engagement with a computational tool; not only would this provide alternative ways to demonstrate CT-S, but it would also reflect the variation that exists in practice.

An additional complexity for assessment design is that just as we see CT-S as situated in activity, we see the activity of CT-S as situated within the “mangle of practice” (Pickering, 2010) that characterizes all knowledge construction in science. As summarized by Sengupta et al. (2018), “Scientists struggle continuously in order to get theories and instruments on one hand and the natural world on the other to perform in the ways that their investigations require” (p. 52). Appreciating this requires appreciating the complex ways that CT-S is engaged by the learner and their peers, as well as by the professional scientist and their colleagues, however untidy this may be. For example, the hypothetical cases of Reflective Use, Design, and Evaluation presented above were provided to help illustrate CT-S concretely and simply. Because of this, the above examples did not illustrate their necessary entanglements with other cognitive processes or provide potential distal outcomes of engaging in CT-S. It is important that such outcomes be considered, even if they are not a requirement of CT-S, as they are often cited as a reason to promote CT-S within science and science education. For example, take a student who is engaging in the Design of a simulation of a real-world system. As

they consider the parameters to include in their simulation, they may realize that they do not actually know how to model one of the relationships within the simulation to accurately reflect the real-world system. This would likely lead them to research the real-world relationship until they are satisfied that they could model it correctly in the simulation. In this example, while it was not initially a goal of their activity, in order to continue working on their Design, they determined that they needed to increase their knowledge about a specific real-world phenomenon. This example illustrates how CT-S can motivate science learning beyond engagement with the computational tool. As a second example, we illustrate how CT-S can motivate science learning beyond the initial science goal while still focusing on the engagement with the computational tool: imagine a student is using a simulation to study predator–prey relationships. As they are engaged in rote use, they notice a menu option that allows them to modify the relative speeds of predators and prey. As the student enters into Reflective Use, they start asking new questions that go beyond their original science goal. After modifying their mental model of the simulation's functionality, they engage in a use of the simulation that helps them learn science beyond their original science goal. This example illustrates how CT-S Reflective Use can provide opportunities for students to ask and investigate new science questions. It also reveals one way in which computational tools developed through and for scientific research have enabled scientific discoveries that were otherwise noninvestigatable: in much of modern science, scale, complexity, and observability limitations are mediated by computational tools that calculate, model, and simulate natural phenomena in novel and transformative ways, enabling old problems to be solved and new questions to be asked. CT-S is inextricably wrapped up in the practice of modern science, and its isolation for the purposes of measurement or instructional design should not imply its severability from other scientific practices *in vivo*.

Implications and future directions

One test of this CT-S framework will be its potential usefulness in examining how the mangle of practice within science intersects with that of computer science, a field where computational abstractions are the principal *outcomes* of activity as well as necessary mediating artifacts, and where programming knowledge is the coin of the realm. While we reject an interpretation of CT-S that requires such knowledge, we anticipate variation in how one engages in CT-S according to one's programming knowledge. Concretely, one who has a certain level of programming knowledge could engage in CT-S differently from one who does not, yet both could still engage

in CT-S. For example, a student engaged in CT-S must have some knowledge of, or have made certain types of assumptions about, the computational tool with which they are engaged. As we posit that CT-S is a form of cognition that arises through engagement with computational tools in science-motivated activity, it is important that we consider programming knowledge, for instance, as a separate artifact that could mediate activity for a subject. Our treatment of science activities as integral to the CT-S framework (Fig. 2) is our attempt to operationalize this complexity of practice. However, further development and scrutiny of measures and of the designs of learning experiences grounded within this framework are necessary to examine how useful this attempt at operationalizing CT-S will be. We posit that an operationalizable CT-S framework will advance research and practice in science learning and propel efforts to position the experiences of individual computational thinkers within their situational learning contexts.

To illustrate how the proposed framework can be utilized to design learning experiences likely to promote CT-S, we offer the following example of how learners may be supported in Reflective (versus rote) Use of a computational tool. Imagine, for instance, a science lesson in which students use a computer-based simulation to investigate the Guiding Question: *How are the moon's phases related to the moon's orbit around earth?* The instructor could choose to begin the lesson by teaching students how to manipulate the simulation. The students might then employ the simulation, in accordance with what they have learned from the instructor, to make observations about the moon's phases and the moon's orbit. Through using the simulation, students would arrive at an answer to the Guiding Question. In a rote use scenario, students would engage in thinking about the moon's phases and orbit; however, it is less likely that students would engage in CT-S because the activity did not require students to build a mental model of the simulation's functionality in relation to their science goal. To promote CT-S more effectively, the instructor could instead structure the lesson to begin with Reflective Use of the computational tool. For example, the instructor could start the lesson by asking students to consider: *How could this computer-based simulation support us in answering our Guiding Question?* Answering this type of initial question requires that students investigate the simulation itself to gain an understanding of its functionality—in other words, answering this type of question requires that students reflectively use the tool. Students would need to attend to the affordances of the simulation's functionality in relation to the ultimate goal of answering the Guiding Question. When the lesson is structured in this way, using the computational

tool could promote students' CT-S in addition to their thinking about the moon's phases and orbit. Moreover, we posit that designing lessons that engage students in Reflective Use could motivate additional science learning opportunities beyond those afforded by the initial Guiding Question. For instance, perhaps during their Reflective Use, the student observes that the moon sometimes enters into an eclipse but does not do so every month; this observation could lead them to manipulating the simulation further to figure out how a 2-D model of the sun–earth–moon system is insufficient for understanding the 3-D interactions between the three objects. Critique of, and revision to, the 2-D model in this way could thus lead to a more sophisticated understanding of the sun–earth–moon system as well as the affordances and constraints of different computational representations of that system. This example illustrates how the framework can support a teacher to design tasks that engage students in meaningful CT-S experiences. Additionally, performance on those tasks is likely to reveal student thinking, which creates rich formative assessment opportunities: as CT-S is elicited more clearly, it is easier to inspect and for a teacher to act on to support student learning.

As the above examples suggest, a number of questions still need to be worked out as researchers, instructional designers, and assessment developers make use of the framework. First, we invite exploration of cognitive processes not already included in the current framework, for as the computational tools that scientists use continue to shape the practice of science and expand what modes of inquiry are possible within that practice, we would expect additional subconstructs of CT-S to emerge. It will also be important to explore whether there are discernible gradations within the cells of the CT-S framework. For example, how can an assessment task that elicits CT-S be designed to distinguish between a high CT-S-involving vs low CT-S-involving performance? Similarly, research could explore the possibility of a CT-S progression across columns (e.g., from Reflective Use to Design within the *modeling* row), or variation across rows within a given column (e.g., differences in CT-S for a *modeling* task versus a *data processing* task within the Design column). In addition to enabling more sensitive assessments and/or more targeted instructional design, such research would advance the field's understanding of the cognitive processes involved in CT-S.

Conclusions

The definition of CT-S that we propose was born out of a need to operationalize the construct so that it could be accurately and reliably measured. In working toward a model of cognition for CT-S, this definition

and accompanying framework draw particular attention to the bidirectional, mediating interactions between a learner, a computational tool, and the goals toward which those interactions are directed. We argue that this framework is well positioned to ground research into the nature of *computational thinking for science* and can serve as a guide for the design of assessment and learning experiences likely to elicit it. In addition to further testing of its use for that purpose, we see a need for further research and theoretical work that can apply this definition to ground the design of learning experiences (e.g., designing tasks to provide students practice with CT-S in ways likely to advance learning), program evaluation (e.g., examining how well activities are aligned with goals, and goals with observable outcomes), and policy initiatives and funding decisions (e.g., predicting what set of initiatives is most likely to lead to desired outcomes).

Abbreviations

CHAT	Cultural–historical activity theory
CT-S	Computational thinking for science
e.g.	Exempli gratia (for example)
K–12	Kindergarten through 12th grade
p/pp.	Page/pages
STEM	Science, technology, engineering, and mathematics

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1838992. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank the members of our Expert Panel—Matthew Berland, Debra Bernstein, Marie Bienkowski, Cynthia D'Angelo, Kemi Jona, Leilah Lyons, Tapan Parikh, Jennifer Wang, David Webb, Michelle Honda Wilkerson, and Marcelo Worsley—as well as Richard Correnti, Neal Finkelstein, Michael Horn, Amanda Peel, Christian Schunn, Carissa Romano, Vasiliki Laina, and Ying-Fang Chen for valuable discussions and feedback that contributed to the ideas presented in this material. We thank the Reviewers for their work reviewing the manuscript and for all of their insightful comments and suggestions. We thank Paula Dragosh for her efforts during the copyediting stage.

Author contributions

All authors provided critical, iterative feedback throughout construct development and helped shape the framework and manuscript. Specifically, RD, MCA, and EG initiated the framing of the Computational Thinking for Science (CT-S) construct and set the foundation for the work. AK, TH, and EG developed drafts of the CT-S framework to visually represent findings from domain analysis. MCo and SA led the gathering of feedback from domain experts and AK, TH, and SA revised the CT-S framework based on this expert input. TH and SA led the development of the CT-S definition with significant input from AK, RM, LB, MCA, and EG. EG worked with SA and TH to apply the lens of Activity Theory to examples of CT-S activity. TH and SA led writing the manuscript with significant contributions from EG, AK, and LB. All authors read and approved the final manuscript.

Funding

This work was supported by the National Science Foundation under Grant No. 1838992.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 24 September 2021 Accepted: 16 December 2022

Published online: 05 January 2023

References

- Ackermann, E. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Routledge.
- Ah-Nam, L., & Osman, K. (2017). Developing 21st century skills through a constructivist-constructionist learning environment. *K-12 STEM Education*, 3(2), 205–216. The Institute for the Promotion of Teaching Science and Technology (IPST). <https://www.learntechlib.org/p/209542/>
- Aho, A. V. (2011). Ubiquity symposium: Computation and computational thinking. *Ubiquity*. <https://doi.org/10.1145/1922681.1922682>
- Aksit, O., & Wiebe, E. N. (2020). Exploring force and motion concepts in middle grades using computational modeling: A classroom intervention study. *Journal of Science Education and Technology*, 29(1), 65–82. <https://doi.org/10.1007/s10956-019-09800-z>
- Barab, S., Schatz, S., & Scheckler, R. (2004). Using activity theory to conceptualize online community and using online community to conceptualize activity theory. *Mind, Culture, and Activity*, 11(1), 25–47. https://doi.org/10.1207/s15327884mca1101_3
- 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. <https://doi.org/10.1145/1929887.1929905>
- Bell, T., Andreae, P., & Lambert, L. (2010, January). Computer science in New Zealand high schools. In *Proceedings of the Twelfth Australasian Conference on Computing Education—Volume 103* (pp. 15–22). <https://dl.acm.org/doi/https://doi.org/10.5555/1862219.1862223>
- Bell, T., Rosamond, F., & Casey, N. (2012). Computer science unplugged and related projects in math and computer science popularization. In H. L. Bodlaender, R. Downey, F. V. Fomin, & D. Marx (Eds.), *Lecture notes in computer science: Vol. 7370. The multivariate algorithmic revolution and beyond* (pp. 398–456). Springer. https://doi.org/10.1007/978-3-642-30891-8_18
- Bienkowski, M., Snow, E., Rutstein, D., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report)*. SRI International. <https://pact.sri.com/downloads/Assessment-Design-Patterns-for-Computational%20Thinking-Practices-Secondary-Computer-Science.pdf>
- Blin, F. (2004). CALL and the development of learner autonomy: Towards an activity-theoretical perspective. *ReCALL*, 16(2), 377. <https://doi.org/10.1017/S0958344004000928>
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)*, Vancouver, Canada (Vol. 1, p. 25). <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Brinda, T., Puhlmann, H., & Schulte, C. (2009). Bridging ICT and CS: Educational standards for computer science in lower secondary education. *ACM Sigcse Bulletin*, 41(3), 288–292. <https://doi.org/10.1145/1595496.1562965>
- Brine, J., & Franken, M. (2006). Students' perceptions of a selected aspect of a computer mediated academic writing program: An activity theory analysis. *Australasian Journal of Educational Technology*. <https://doi.org/10.14742/ajet.1305>
- Brown, N. J., & Wilson, M. (2011). A model of cognition: The missing cornerstone of assessment. *Educational Psychology Review*, 23(2), 221–234. <https://doi.org/10.1007/s10648-011-9161-z>
- Cole, M., & Engeström, Y. (1993). A cultural-historical approach to distributed cognition. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 1–46). Cambridge University Press.

- Cole, M. D., Engeström, Y., & Vasquez, O. (1997). *Mind, culture, and activity: Seminal papers from the Laboratory of Comparative Human Cognition*. Cambridge: Cambridge University Press.
- College Board. (2019). *AP computer science principles*. AP Course Overview. <https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-overview.pdf?course=ap-computer-science-principles>
- Computing at School Working Group. (2012). *Computer science: A curriculum for schools*. <https://www.computingatschool.org.uk/resource-library/2009/march/computing-a-curriculum-for-schools>
- Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education*, 18(1), 41–67. <https://doi.org/10.15388/infedu.2019.03>
- Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript, referenced in <https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Denning, P. J. (2017). Computational thinking in science. *American Scientist*, 105(1), 13–17. <https://doi.org/10.1511/2017.124.13>
- Dickes, A. C., Sengupta, P., Farris, A. V., & Basu, S. (2016). Development of mechanistic reasoning and multilevel explanations of ecology in third grade using agent-based models. *Science Education*, 100(4), 734–776. <https://doi.org/10.1002/sce.21217>
- Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robertson, R., & Andrews, A. (2019, February). PRADA: A practical model for integrating computational thinking in K-12 education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 906–912). <https://doi.org/10.1145/3287324.3287431>
- Dorph, R., Cannady, M. A., & Schunn, C. D. (2016). How science learning activation enables success for youth in science learning experiences. *Electronic Journal of Science Education*, 20(8), 89.
- Engeström, Y. (1999). Activity theory and individual and social transformation. *Perspectives on Activity Theory*, 19(38), 19–30. <https://doi.org/10.1017/CBO9780511812774.003>
- Engeström, Y. (2015). *Learning by expanding: An activity-theoretical approach to Developmental Research*. Cambridge University Press.
- Folk, R., Lee, G., Michalenko, A., Peel, A., & Pontelli, E. (2015, December). GK-12 DISSECT: Incorporating computational thinking with K-12 science without computer access. *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). <https://doi.org/10.1109/FIE.2015.7344238>
- Foster, I. (2006). A two-way street to science's future. *Nature*, 440(7083), 419–419. <https://doi.org/10.1038/440419a>
- Google for Education. (2019). CT overview. Exploring Computational Thinking. <https://edu.google.com/resources/programs/exploring-computational-thinking/#lct-overview>
- Greeno, J. G. (1998). The situativity of knowing, learning, and research. *American Psychologist*, 53(1), 5. <https://doi.org/10.1037/0003-066X.53.1.5>
- Greeno, J. G. (2015). Commentary: Some prospects for connecting concepts and methods of individual cognition and of situativity. *Educational Psychologist*, 50(3), 248–251. <https://doi.org/10.1080/00461520.2015.1077708>
- Grover, S., Fislser, K., Lee, I., & Yadav, A. (2020, February). Integrating computing and computational thinking into K-12 STEM learning. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 481–482). <https://doi.org/10.1145/3328778.3366970>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6), 1–165. <https://doi.org/10.2200/S00684ED1V01Y201511HC1033>
- Huang, W., & Looi, C.-K. (2021). A critical review of literature on “unplugged” pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, 31(1), 83–111. <https://doi.org/10.1080/08993408.2020.1789411>
- Hutchins, N. M., Biswas, G., Maróti, M., Lédécezi, Á., Grover, S., Wolf, R., & McElhaney, K. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 29(1), 83–100. <https://doi.org/10.1007/s10956-019-09804-9>
- Issroff, K., & Scanlon, E. (2002). Educational technology: The influence of theory. *Journal of Interactive Media in Education*. <https://doi.org/10.5334/2002-6>
- Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014, January). *Embedding computational thinking in science, technology, engineering, and math (CT-STEM)*. Future Directions in Computer Science Education Summit Meeting. <https://ccl.northwestern.edu/papers/2014/OrtonKaiNorthwestern-1.pdf>
- Jonassen, D. H. (2000). Revisiting activity theory as a framework for designing student-centered learning environments. *Theoretical Foundations of Learning Environments*, 7, 89–121.
- Jonassen, D. H., & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and Development*, 47(1), 61–79. <https://doi.org/10.1007/BF02299477>
- K-12 Computer Science Framework. (2016). Retrieved from <http://www.k12cs.org>
- Kaczmarczyk, D., & Dopplick, R. (2014). *Rebooting the pathway to success: Preparing students for computing workforce needs in the United States*. Education Policy Committee, Association for Computing Machinery. https://pathways.acm.org/ACM_pathways_report.pdf
- Kapelinin, V., & Nardi, B. A. (2006). *Acting with technology: Activity theory and interaction design*. MIT Press.
- Kite, V., & Park, S. (2020, March). Secondary science teachers' conceptualizations of computational thinking and perceived barriers to CT/content integration. In *Prepared for the 2020 annual meeting of the National Association for Research in Science Teaching and teacher education (NARST)*. <https://www.researchgate.net/publication/340175597>
- Krugel, J., & Hubwieser, P. (2018). Strictly objects first: A multipurpose course on computational thinking. In M. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp. 73–98). Berlin: Springer.
- Kuutti, K. (1995). Activity theory as a potential framework for human-computer interaction research. In B. Nardi (Ed.), *Context and consciousness: Activity theory and human-computer interaction* (pp. 17–44). MIT Press.
- Lealdino Filho, P., & Mercat, C. (2018). Teaching computational thinking in classroom environments: A case for unplugged scenario. In *Proceedings of the Resources 2018 International Conference* (pp. 296–299).
- Lee, I., & Malyn-Smith, J. (2020). Computational thinking integration patterns along the framework defining computational thinking from a disciplinary perspective. *Journal of Science Education and Technology*, 29(1), 9–18. <https://doi.org/10.1007/s10956-019-09802-x>
- Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K-8 curriculum. *ACM Inroads*, 5(4), 64–71. <https://doi.org/10.1145/2684721.2684736>
- Leont'ev, A. N. (1978). *Activity, consciousness, and personality*. Prentice Hall.
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research*, 3, 1–18. <https://doi.org/10.1007/s41979-020-00030-2>
- Lowe, T., & Brophy, S. (2017, October). An operationalized model for defining computational thinking. *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). <https://doi.org/10.1109/FIE.2017.8190682>
- Malone, K. L., Schunn, C. D., & Schuchardt, A. M. (2018). Improving conceptual understanding and representation skills through Excel-based modeling. *Journal of Science Education and Technology*, 27(1), 30–44. <https://doi.org/10.1007/s10956-017-9706-0>
- 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* (p. 5). <http://www.cs.uml.edu/~fredm/papers/CT%20from%20a%20disciplinary%20perspective%20Malyn-Smith%20et%20al%202018.pdf>
- Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A brief introduction to evidence-centered design. *ETS Research Report Series*, 2003(1), i–29. <https://doi.org/10.1002/j.2333-8504.2003.tb01908.x>
- Murphy, E., & Rodriguez-Manzanares, M. A. (2008). Using activity theory and its principle of contradictions to guide research in educational technology. *Australasian Journal of Educational Technology*. <https://doi.org/10.14742/ajet.1203>
- Nardi, B. A. (Ed.). (1996). *Context and consciousness: Activity theory and human-computer interaction*. MIT Press: New York.
- Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM*, 62(2), 32–35. <https://doi.org/10.1145/3231587>

- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.
- National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*, pp. 65–66. National Academies Press. <https://www.nap.edu/catalog/13165/a-framework-for-k-12-science-education-practices-crosscutting-concepts>
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Basic Books.
- Pea, R. D. (1993). Practices of distributed intelligence and designs for education. *Distributed Cognitions: Psychological and Educational Considerations*, 11, 47–87.
- Peel, A., Sadler, T. D., & Friedrichsen, P. (2019). Learning natural selection through computational thinking: Unplugged design of algorithmic explanations. *Journal of Research in Science Teaching*, 56(7), 983–1007. <https://doi.org/10.1002/tea.21545>
- Peel, A., Sadler, T. D., & Friedrichsen, P. (2022). Algorithmic explanations: An unplugged instructional approach to integrate science and computational thinking. *Journal of Science Education and Technology*, 34, 1–14. <https://doi.org/10.1007/s10956-022-09965-0>
- Pickering, A. (2010). *The mangle of practice: Time, agency, and science*. University of Chicago Press.
- Rutstein, D. W., Snow, E., & Bienkowski, M. (2014, April). Computational thinking practices: Analyzing and modeling a critical domain in computer science education. In *Proceedings of the 2014 Annual Meeting of the American Educational Research Association (AERA)*, Philadelphia, PA. https://pact.sri.com/downloads/Rutstein_2014_AERA_Preso_CTP.pdf
- Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE 2014*. ACM. https://www.researchgate.net/publication/299450690_Computational_thinking_the_developing_definition
- Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. In M. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp. 49–72). Springer: Berlin.
- 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. <https://doi.org/10.1007/s10639-012-9240-x>
- Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., & Wildeman, B. (2012, July). Infusing computational thinking into the middle- and high-school curriculum. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education* (pp. 22–27). <https://doi.org/10.1145/2325296.2325306>
- Shaffer, D. W., & Clinton, K. A. (2006). Toolforthoughts: Reexamining thinking in the digital age. *Mind, Culture, and Activity*, 13(4), 283–300. https://doi.org/10.1207/s15327884mca1304_2
- Solvie, P., & Kloek, M. (2007). Using technology tools to engage students with multiple learning styles in a constructivist learning environment. *Contemporary Issues in Technology and Teacher Education*, 7(2), 7–27.
- UlrichHoppe, H., & Werneburg, S. (2019). Computational thinking—more than a variant of scientific inquiry! In: S.C. Kong & H. Abelson (Eds.), *Computational thinking education* (pp. 13–30). Springer, New York
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.). Harvard University Press.
- Vygotsky, L. S. (2012). *Thought and language*. MIT Press.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Weintrop, D., Rutstein, D., Bienkowski, M., & McGee, S. (2021). Assessment of computational thinking. *Computational Thinking in Education*, 23, 90–111. <https://doi.org/10.4324/9781003102991-6>
- Wiese, E. S., & Linn, M. C. (2021). "It must include rules": Middle school students' computational thinking with computer models in science. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 28(2), 1–41. <https://doi.org/10.1145/3415582>
- Wilson, K. G. (1989). Grand challenges to computational science. *Future Generation Computer Systems*, 5(2–3), 171–189. [https://doi.org/10.1016/0167-739X\(89\)90038-1](https://doi.org/10.1016/0167-739X(89)90038-1)
- Wing, J. (2010). Research notebook: Computational thinking—what and why. *The Link Magazine*, 6, 89.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- 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. <https://doi.org/10.1007/s11528-016-0087-7>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)