Finding the Right "Look": Charting the Capacity of "Look For" Documents to Discern CT Integration in Elementary and Middle School Classrooms

Kelly Mills, Pati Ruiz, and Quinn Burke

I. Overview

This workshop focused on the development of tools that can support teachers to overcome barriers and develop the competencies necessary to implement district-wide equitable computing pathways. After a round of introductions, workshop participants learned about the "Look for" documents developed by Digital Promise (DP) in collaboration with a range of school district partners through DP's national network of the League of Innovative Schools, a coalition of 114 school districts in 34 states. These "Look for" documents are currently being used by League teachers to plan lessons and reflect on their practice, as well as school administrators to develop a better understanding of what computing looks like in the classroom as well as a guide for designing and conducting appropriate evaluations. After the initial presentation of the "Look for" documents created for teachers to integrate computational thinking into middle school science classrooms, workshop participants shared examples of tools documents that they have used in their own projects to operationalize CT - the content shared is included in the workshop slide deck since each participant developed a slide to share their work. The participants in this workshop represented seven different RPPs and held a variety of roles on their projects. The next section describes the artifacts and resources shared by participants.

II. Artifacts and Resources Shared

"Look for" documents

The **Computational Thinking Micro-credentials and Next Generation Science Standards (NGSS)** project is a two year effort by Digital Promise to integrate computational thinking specifically to middle school science. Funded by the Carnegie Corporation of New York, DP is currently working with five districts from across the United States to develop sustained professional learning for teachers embedded with micro-credentials and corresponding pedagogical resources. One such resource is defining “Look-fors” to prompt teachers and students to find opportunities to practice CT in middle school science, with applications to other disciplines. Teachers have used these successfully as a self-reflective measure to gauge their own understanding on key CT practices and their capacity to enact these with their middle school students as well as to plan and reflect on CT lessons (Mills, 2020). District instructional leaders have also found these look-for documents very useful to develop a better understanding of what computing looks like in the classroom and to guide classroom observations.

The “Look for” documents align to each computational thinking practice in our stack of educator micro-credentials (Burke et al., 2020; O’Donnell, 2019; Proctor, 2019). In addition to “Look fors,” there are additional resources that provide seeds of activities that teachers can adapt for multiple contexts. While these resources were developed in the context of middle school science, they can be adapted for other content areas and/or grade bands. Each resource collection contains the following components:

- **“Look fors”** are discernable student behaviors for each computational thinking practice. Teachers may reference "Look fors" in order to consider opportunities to integrate computational thinking into their classroom and/or reflect on how students engaged in lessons integrated with computational thinking.
- **Prompting questions** are examples of questions to ask students to reflect on their process or progress. Teachers may use one or more of these questions to more explicitly integrate computational thinking into their instruction.
- **Activities** are pedagogical routines in science that can be enhanced with computational thinking. We also clarify how each activity may be applicable to science instruction under the “You might use this when…” column within the resource.
- **Templates** are student resources that can be adapted for classroom use. Teachers may use or modify templates to integrate computational thinking in their classroom.

An example resource document is pictured in Figure 1 for the computational thinking practice Creating Algorithms (Mills, 2020). Digital Promise has designed resources for additional practices including working with data, creating computational models and understanding systems, all of which will be published in summer 2020.

Creating Algorithms

Use this resource to reflect on your practice or identify opportunities to integrate algorithms in your classroom. This resource was developed to support middle school science teachers to integrate computational thinking into NGSS-aligned lessons. The content can be adapted for other content areas, grade bands and contexts.

What is an algorithm?

A repeatable process that delivers an expected result.

Look fors:

Students may be creating algorithms when they are:

- ❑ Decomposing problems or tasks
- ❑ Identifying essential steps
- ❑ Testing and debugging
- ❑ Considering efficiency

Prompting questions:

Ask students to reflect on their process or progress with these prompting questions:

- ❑ What do you need to know to be able to solve this problem/do this task?
- ❑ What should be the result of the problem or task?
- ❑ Why is each step required to solve the problem/task?
- ❑ Would you include additional words or details to explain this process to a partner?
- ❑ Does each step have the result you intend it to?
- ❑ Does a partner testing your algorithm get the same results as you?
- ❑ Are there certain inputs where you do not get the intended result?
- ❑ Can your algorithm have the same result with less steps?
- ❑ Do you notice any patterns in your procedure?

Figure 1. “Look for” document for Creating Algorithms

Practitioner Adapted Framework for Integrating CT in Elementary Science

Integrating Computational Thinking Into Elementary Science Pre-service Teacher Education, an NSF-funded project from University of Maryland, has facilitated ongoing professional development to 120+ pre-service elementary science teachers and 20+ in-service mentor teachers over the course of two years (Ketelhut et al, 2019). Merijke Coenraad, a doctoral candidate, shared a framework the team iteratively developed to clarify computational thinking practices that are applicable to teachers and students in elementary school (Figure 2). The team initially defined computational thinking for teachers using existing documents from the field (e.g. CSTA & ISTE, 2011; Weintrop et al., 2016). They adapted this definition based on points of confusion that arose during the professional development and subsequent development of lesson plans. The modified definition included the following revisions: (1) elimination of practices not applicable to grades K-5 (2) modification of computer science jargon and (3) application of practices to computer science (Mills et al., 2020; Cabrera et al., In preparation). Teachers have reported that the modified framework has helped them to define and conceptualize CT, serving as a guiding document in lesson planning, in conjunction with professional development activities (Cabrera et al., In preparation).

The Maryland State Department of Education and Maryland Center for Computing Education has been supporting districts to respond to the state-wide mandate for all students to engage in computational thinking with a specific requirement in Middle School (Maryland State Department of Education, 2020). Many districts have chosen to meet the requirement with a standalone course (e.g. Code.org) but some districts are integrating into other curricular areas such as science and the arts.

Elissa Hozore, a computer science specialist from the Maryland State Department of Education (MSDE), shared resources developed to discern if computational thinking was intentionally and rigorously being integrated into middle school curricula. MSDE developed a rubric that has three levels - beginning, approaching and at standard. Hozore explained, “what we were expecting from the computer science standards was a really big reach for a lot of them [teachers/districts]...we created a beginning, approaching and at standard so they could see even if they were just at the beginning phase they could see the pathway we were hoping they would go on to do a little bit more rigorous job of integrating the computational thinking.” The three levels are outlined for each of six practices: decomposition, pattern recognition, abstraction, algorithmic design, data collection and visualization and troubleshooting and evaluation. For each practice, additional resources provide examples of lessons that fall into the beginning, approaching and at standard categories. The rubrics are currently available to curriculum writers across the state to identify, clarify and develop opportunities for computational thinking integration in middle school.

CS Equity Guide and CS Explainers

The CS Equity Guide and CT and CS Explainer were shared by Roxana Hadad who joined the session with the Supporting Computing Access, Leadership, and Equity in California (SCALE-CA). SCALE-CA addresses the needs of administrators, counselors, as well as teachers in the CS for all movement. The CS Equity Guide was written in collaboration with education leaders throughout California and is used by schools, districts, and counties within and beyond California. The guide was designed for administrators interested in implementing equity-minded CS in their schools, districts, or counties and describes issues like funding, student recruitment, the difference between a career training track for CS and a more academic CS track, and navigating the systemic issues that prevent CS from being equitable. For example, this guide helps administrators and counselors make sure that they don't only offer CS as an elective because students who have to use their elective time to take their English as a second language courses.

The CT and CS Explainer was developed by Hadad (2020) and is the first in a series of "CS explainers" produced by CSforCA. This explainer was developed for use with policy makers in California who need to understand these concepts in order to legislate effectively. The CT and CS Explainer and future "CS Explainers" can help legislators understand CT and CS and where in the curriculum this content should be integrated so that all students have access to it.

III. Breakout Groups

Having shared a range of tools and experiences, the participants then broke out into two discussion groups to identify and discuss the development of tools to support CT integration. One group focused on reviewing and giving feedback on CT integration tools that were developed in the state of Maryland and brought to the workshop by participant Elissa Hozore. These documents were originally developed for school districts as a way of supporting teachers as they integrate CT. The other group focused on discussing the development of tools for administrators and brainstormed how they might conduct classroom observations and what support they might benefit from. The group discussed how the tools developed for administrators need to go hand in hand with other support like professional development and ongoing learning communities. The focus group discussions are described in more detail below.

Mills, Hozore and Coenraad composed one breakout group. Together, they examined the resources developed by the Maryland State Department of Education to clarify and operationalize computational thinking practices for district leaders, curriculum writers and teachers. There were three main points of discussion: (1) identifying competencies, (2) selecting terminology, and (3) highlighting examples of integration.

The group began by discussing how the competencies were selected. As a starting point, the State of Maryland had previously identified four practices for computational thinking: decomposition, pattern recognition, abstraction, algorithmic design. MSDE added data collection and visualization and troubleshooting and evaluation based on evolving conceptualizations from the field and connections to existing curriculum. The group discussed the importance of helping curriculum writers and teachers to realize connections between practices and iteratively redefining the competencies based on feedback from teachers and districts.

The conversation then turned to the language used to describe the competencies in the tools. There was concern that some of the more technical language in the tools may be inaccessible to novices. The group discussed the importance of considering the audience when selecting appropriate terminology and clarifying technical terms when possible.

There was group consensus that providing relatable and concrete lessons and activities, both plugged and unplugged, at each level of the tools was an especially helpful resource for districts and teachers. The group suggested that the structure of the document should highlight the examples in relation to each other for each practice to help define a progression toward strong integration.

Hadad, Sayler, Wortel-London, and Zarch participated in the second breakout group. This group began with a discussion of what an administrator walk-through might look like in a middle school and how a “Look For” document might help administrators newer to CS and CT understand where CT is being taught or integrated into instruction. This group also discussed how teachers have found the “Look For” documents helpful as they plan for instruction. The conversation then shifted to discussing what other tools might support administrators and what challenges administrators might have in supporting the integration of CT in the middle school context in particular. Hadad suggested a need to offer rich examples and exemplars for norming administrators - this could come in video or slide deck form. It is important to help administrators understand what is unique in teaching computational thinking and computer science and what the overlap is between “good teaching practice” and “good teaching practice for computational thinking.”

IV. Discussion

Whereas the first breakout group focused more on the language of and process for identifying relevant CT competencies, the second breakout group was largely centered on the necessary tools to increase administrator buy-in and capacity to effectively assess high-quality CT instruction within school classrooms. Both groups however mutually focused on the need to develop **more substantial examples of what CT actually looks like in classrooms**. The need for such examples is such that even examples of what CT is NOT was encouraged, with one workshop participant in the post-survey remarking, “I'd like more examples of what teachers might incorrectly believe to be CT that really isn't.”

This issue of definition has proven to be a consistent challenge for school districts ever since Jeanette Wing popularized the term “computational thinking” in her seminal article (2006) in the *Communications of the ACM*. Whereas some districts have reduced CT to simply learning to code, other districts have conceptualized CT into far more nebulous realms such as "problem solving" and "21st centruy learning skills" which offer little in terms of actual substance and cross-disciplinary integration. Organizations like the International Society for Technology in Education (ISTE) and companies like Google have made strides in terms of developing modest repositories for educators, but clearly there needs to be more considerable developments here that focus in particular on K-8 integration. Certainly robust examples that focus on what CT is (and isn't) would be beneficial to teachers, but one of the key-takeaways from the session was how such a repository would also be useful to administrators. Indeed, to enact clear and cumulative K-12 computing pathways (Burke et al., 2020), it is the role of administrators to monitor for consistent implementation that takes into account the grade level of the students, prior lessons in CT, and the subject matter itself.

In the post-survey to the workshop, one participant mentioned the potential of coaching to help support teachers more ably integrate CT into their existing lessons. To the degree that the "Look for" documents presented to the group can facilitate such coaching would represent a significant step forward for districts committed to developing their own K-12 computing pathways. While the question of assessing CT surfaced during this session, the purpose of the "Look for" document is not to be a rigid punitive check-list but rather a means to attend to *discernible classroom practice.* Not unlike CSforAll's (Vogel, Santo, & Ching, 2017) development of the Strategic CSforALL Resource & Implementation Planning Tool (i.e., SCRIPT), **"Look for" documents offer educators and administrators a mutual starting point for collaborative visioning, self-assessment, and collective goal-setting** for classroom practice.

How can future PD "be paired with these 'Look for' documents," wondered one attendee over the final third of the session, and specifically how can such PD be inclusive of not just teachers but administrators? This represents a tenable next step for the team. While the majority of federal and state funding for CS/ CT education has been focused on teacher training and student learning platforms, programs such the National Center for Women in Technology's (NCWIT) Counselors for Computing (C4C) program has had demonstrated success broadening PD to K-12 administrators as well, leveraging their roles as conduits between grade level computing activities and course offerings. Going forward with "Look fors" in computing, the team is committed to forming district and state partnerships to incorporate these into existing PD offerings as well as identifying funding opportunities to more systematically evaluate their effect on classroom practice and the integration of computing into existing subject matters.

References

Burke, Q., Roschelle, J., Bailey, C., Angevine, C., Weisgrau, J., & Mills, K. (2020, March). The why, who & how of teaching K-12 computing? Examining teacher perspectives on computational thinking in K-12 classrooms. Paper presentation at the *2020 RESPECT Conference*, March 11th, Portland, OR.

Burke, Q., Angevine, C., Proctor, C., & O'Donnell, K. (2020). "Credentialing computation: Empowering teachers in computational thinking through educator microcredentials." In *Preparing Teachers to Teach Computer Science: Models, Practices and Policies* (Eds., C. Mouza, A. Yadav, & A. Leftwich). Information Age Publishing.

Cabrera, L., Mills, K. M., Coenraad, M., Killen, H., Ketelhut, D. J., McGinnis, J. R., & Plane, J. (In preparation). *Designing a Framework for the Integration of Computational Thinking into Elementary Science Teacher Education*.

Ketelhut, D. J., Cabrera, L., McGinnis, R. J., Plane, J., Coenraad, M., Killen, H., & Mills, K. M. (2019). Exploring the Integration of computational Thinking into Preservice Elementary Science Teacher Education. *National Science Foundation STEM+C PI Meeting*.

Mills, K., Coenraad, M., Cabrera, L., Killen, H., Ketelhut, D.J., & McGinnis, J.R. *A Design Based Research Approach to Integrating Computational Thinking into Elementary Science Teacher Education.* American Education Research Association, San Francisco, California, USA, April 17-21, 2020. (conference canceled).

Hadad, R. (2020). What is Computational Thinking and What Does It Have to Do with K-12 Computer Science Education? Retrieved from: https://docs.google.com/document/d/1sSH86l_E4QMEwjvyF3lMajMYGLfe4X7R2oTgSDcsjLw/edit?ts=5e442228

Hug, S. & Krauss, J. (2016). Engaging school counselors, creating computing allies. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, February, p. 697. https://doi.org/10.1145/2839509.2850567.

ACCESS and CSforCA (2019). CS Equity Guide: An Administrator's Guide to Implementing Equitable K-12 Computer Science Education in California. Retrieved from: http://access-ca.org/wp-content/uploads/sites/4/2019/04/CSforCAguide-opt.pdf

Maryland State Department of Education (2020). Every Student Succeeds Act. Retrieved from: http://marylandpublicschools.org/about/Pages/DAPI/ESSA/index.aspx

Mills, K. (2020). *Creating Algorithms.* Retrieved from: https://digitalpromise.org/wp-content/uploads/2020/03/Creating-Algorithms-Collection-v2.pdf

Mills, K. (2020, March 3). *New Computational Thinking Resources for Powerful Learning*. Retrieved from https://digitalpromise.org/2020/03/03/new-computational-thinking-resources-for-powerful-learning/

Proctor, C., Angevine, C., Kabaker, J., O'Donnell, K., Weisgrau, J. (revised 2019). Computational Thinking: Pedagogies (Micro-credentials). Digital Promise. Retrieved From: https://microcredentials.digitalpromise.org/explore?page_size=24&page=1&tag=Computational%20Thinking:%20Pedagogies

O'Donnell, K., Angevine, C., Kabaker, J., Proctor, C., Weisgrau, J. (revised 2019). Computational Thinking: Practices (Micro-credentials). Digital Promise. Retrieved From: https://microcredentials.digitalpromise.org/explore?page_size=24&page=1&tag=Computational%20Thinking:%20Practices

Vogel, S., Santo, R., & Ching, D. (2017, March). Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Seattle WA, March, (pp. 609-614).

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), pp. 33-35.