The State of the Field in Computational Thinking Assessment

Mike Tissenbaum (organizer), Massachusetts Institute of Technology, miketissenbaum@gmail.com
Josh Sheldon (organizer), Massachusetts Institute of Technology, jsheldon@mit.edu
Mark Sherman (organizer), Massachusetts Institute of Technology, shermanm@mit.edu
David Weintrop, University of Maryland, weintrop@umd.edu
Kemi Jona, Northeastern University, k.jona@northeastern.edu
Michael Horn, Northwestern University, michael-horn@northwestern.edu
Uri Wilensky, Northwestern University, uri@northwestern.edu
Satabdi Basu, SRI International, satabdi.basu@sri.com
Daisy Rutstein, SRI International, daisy.rutstein@sri.com
Eric Snow, SRI International, eric.snow@sri.com
Linda Shear, SRI International, linda.shear@sri.com
Shuchi Grover, shuchig@cs.stanford.edu
Irene Lee, Massachusetts Institute of Technology, ireneannelee@gmail.com
Eric Klopfer, Massachusetts Institute of Technology, klopfer@mit.edu
Debora Lui, University of Pennsylvania, deb.lui@upenn.edu
Gayithri Jayathirta, University of Pennsylvania, gayithri@gse.upenn.edu
Mia Shaw, University of Pennsylvania, mshaw12@gse.upenn.edu
Yasmin B. Kafai, University of Pennsylvania, kafai@upenn.edu
Deborah A. Fields, Utah State University, deb.lui@upenn.edu
Nichole Pinkard, Northwestern University, nichole.pinkard@northwestern.edu
Caitlin K. Martin, Digital Youth Network, cmartin@digitalyouthnetwork.org
Sheena Erete, Depaul University, serete@depaul.edu
Eni Mustafaraj, Wellesley College, emustafa@wellesley.edu
Clara Sorensen, Wellesley College, csorense@wellesley.edu
Will Temple, University of Colorado Boulder, William.Temple@colorado.edu
R. Benjamin Shapiro, University of Colorado Boulder, Ben.Shapiro@colorado.edu
Rogers Hall (discussant), Vanderbilt University, rogers.hall@vanderbilt.edu

Abstract: While interest in computational thinking (CT) education has grown globally in the past decade, there lacks a single unified definition of CT. This can pose significant challenges for researchers, teachers, and policy makers trying to decide which assessment methods are appropriate for their specific CT interventions. Rather than trying to create a single unified definition of CT, this symposium brings together a broad spectrum of leading CT researchers to share what CT means for them, how it influenced their learning designs, and the methods for assessing CT learning. This interactive session will showcase these different views of CT in a single place and serve as a rich opportunity for comparison and discussion.

Introduction

The learning sciences community has shown considerable interest in the role that learning to code should play in preparing students for the 21st century workforce. However, within computing education, there is a growing chorus of researchers, administrators, and policy makers who are advocating for inclusion of a broader set of computational skills, often referred to as computational thinking (Wing, 2006). Still, developing computational thinking curricula and assessing computational thinking learning is a persistent challenge. In part, this is because the broader community does not have a common definition of computational thinking (National Academy of Sciences, 2010). In some cases, definitions of CT focus on key concepts such as abstractions, algorithms, and conditional logic (Grover & Pea, 2013; Brennan & Resnick, 2012). Other definitions have focused on learners' developing their sense of belonging as members of the broader computational community (Erete, Martin & Pinkard, 2017), or feeling empowered to develop solutions to problems in their daily lives (Tissenbaum, et al, 2017), yet others on using CT to engage with powerful ideas in scientific disciplines (Wilensky, Brady & Horn, 2014). This also raises critical questions for researchers, teachers, and administrators about the appropriate methodology for assessing CT learning, given each group's particular goals.

The current global landscape of computational thinking education offers us an ideal opportunity to bring together leading researchers in the field to critically discuss current approaches for assessing computational thinking. To this end, this symposium brings together a spectrum of learning sciences researchers investigating

CT learning. In particular, the presenters in this symposium will provide detailed examples and insights into 1) what computational thinking means to them and how these varied definitions have informed their research; and 2) the methods and assessments they use to evaluate changes in learners' computational thinking.

Objectives

Together, these contributions aim to provide a single venue for advancing understanding of the range of methods available for assessing changes in computational thinking. These include interactive online assessments (Weintrop et al.); evidence-centered design (Basu et al.); "systems of assessment" (Grover); portfolios (Lui et al.); triangulating data mining and qualitative measures (Tissenbaum et al.); digital ethnographies (Pinkard et al.); incremental problem-solving strategies (Mustafaraj & Sorensen); and utility-based assessments (Temple & Shapiro). While all the contributions will emphasize their assessment approaches, the design and context of each intervention will provide added insight into researchers' assessment decisions. This symposium will provide opportunities to examine similarities and differences in perspectives on CT, theoretical and methodological frameworks, and pedagogical goals for assessing CT learning.

Session format

To promote active and productive discussion, the symposium will be conducted as an interactive demonstration. Following brief teaser introductions on each project, attendees will be invited to explore stations at which presenters will have posters showing their respective works around computational thinking assessment. This will provide attendees ample opportunities to examine and discuss the methodological decisions made by the presenters, and how they may be adapted for attendees' own designs in a way that traditional talk do not allow. The symposium will close with an open discussion period.

Implications

Given the increased interest in K-12 computing education, this symposium comes at an important time for the learning sciences. Around the world, governments and businesses are realizing the importance of teaching computing to students, which has resulted in a wave of new pedagogical approaches and computing curricula. Despite this growth there have been too few discussions about *what the learning is* that we are looking for in these interventions, and more importantly how we can reasonably assess if such learning is taking place. This symposium brings together researchers with a diverse set of approaches that tackle this challenge head-on, from a variety of perspectives and with methods that are grounded in real-world contexts. As a result, this symposium provides an important collection of exemplar cases to inform the broader learning sciences community's own research into computational thinking education.

Assessing computational thinking embedded in mathematics and science contexts

David Weintrop, Kemi Jona, Michael Horn, and Uri Wilensky

As we seek to provide opportunities for all learners to engage with computational thinking, we face a number of challenges. Computational thinking, if it is taught explicitly at all, is conventionally embedded in computer science classrooms. This means students at schools that do not offer computer science classes (often due to a lack of resources, infrastructure, or qualified teachers) and learners without access to or interest in computer science courses never learn these important 21st century skills. This situation perpetuates existing inequalities in computing. The strategy we pursue for addressing these challenges is to integrate computational thinking into existing mathematics and science classrooms (Weintrop et al., 2016). Given that all schools teach these subjects and all students are enrolled in these classes, this integrative approach reaches a much greater range of learners (Orton et al., 2016). Further, computational thinking and high school disciplines can be mutually supportive: computational thinking can deepen content learning (Wilensky, Brady, & Horn, 2014) and high school disciplines can offer meaningful contexts for situating computational thinking concepts (Weintrop et al., 2016). Additionally, given the growing role of computing in contemporary mathematics and science, this approach better prepares learners to be scientifically literate citizens. Our conceptualization of computational thinking is based on Weintrop et al.'s (2016) *Computational Thinking in Mathematics and Science Practices Taxonomy*.

To evaluate students' emerging computational thinking abilities in math and science classrooms, we developed a series of interactive, online assessments (Weintrop et al., 2014). In the creation of these assessments, we followed two central design principles. First, each assessment is situated in a mathematics or science scenario so as to be authentic with respect to the nature of the computational thinking practices students are being evaluated on while also not being dependent on specific domain content knowledge. For example, our assessments include scenarios such as gathering data on a fictitious local bird populations and analyzing public datasets related to human development grouped by nation. Second, the assessments challenge learners to enact the computational thinking practices they are being assessed on. This means the assessments often have embedded computational tools, such as interactive visualizations, simplified programming tools, or models of scientific phenomena. For example, one assessment includes embedded computational models and asks students to run the models using a variety of parameter settings, make sense of the data the models produce, and reflect on design decisions related to the model. Our goal in developing these assessments is to help educators and researchers better characterize students' emerging understanding of and competencies in computational thinking practices in mathematics and science contexts. In doing so, we hope to stimulate further integration of computational thinking across a growing set of disciplinary contexts and to facilitate a deeper understanding of the developmental progression of computational thinking skills and practices across contexts, settings, and populations.

Evidence-centered design: A principled approach to creating assessments of computational thinking practices

Satabdi Basu, Daisy Rutstein, Eric Snow, and Linda Shear

Computational thinking (CT) refers to the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer (Wing, 2006). CT is increasingly being recognized as an essential form of literacy for informed citizens, not just for computer scientists, so that they feel empowered to leverage the power of computation to solve problems in their daily lives. This viewpoint initially motivated a body of research on CT assessments that measure students' interests in computing and their likelihood of pursuing CS courses in the future (Basu et. al, 2016).

While measuring the extent to which students feel computationally empowered and interested is important, our research on creating CT assessments has focused on designing ways to measure student behavior and thought processes while engaging in CT practices. In creating measures of CT practices, we try to move away from the assessment of factual knowledge about CS concepts, and toward applying the knowledge to solve problems (Snow et al., 2017). Our approach for developing assessments that measure CT focuses less on programming syntax and constructs and more on foundational problem solving skills that transcend disciplines. For example, choosing appropriate representations for problems, modeling relevant aspects (and ignoring irrelevant aspects) of problems to make them tractable, and using different levels of abstraction for problem solving are all examples of CT practices that are useful across disciplines.

We employ a principled approach to systematically produce the required evidence of students' CT practices. Evidence-Centered Design (ECD) is a systematic design process that improves the coherence of assessments by explicitly linking task features, the evidence of student performances generated by the tasks, and the knowledge and skills implicated by the evidence (Mislevy & Haertel, 2006). The ECD process typically starts with a domain analysis to identify and analyze the domain, constructs, and underlying skills of interest. For domain analysis of CT practices in the context of a specific curriculum, we refer to our CT practice design patterns from prior work (Bienkowski et al., 2015), review the learning objectives and lesson activities specified in the curriculum, and obtain input from the curriculum design team, experienced teachers, and experts in Computer Science, Learning Sciences and assessment design. The results from the domain analysis give us the information we need to specify the focal knowledge, skills and abilities (FKSAs) in the ECD domain modeling phase. While CT practices can be instantiated in different contexts, the knowledge and skills at the curriculum unit level are at a finer grain size and are related to the particular learning objectives of the unit. Along with articulating FKSAs, we also specify characteristic features that the task must contain and features of tasks that can be varied to make new, related tasks. A "task" refers to an authentic scenario with a related set of questions. Additionally, we also create examples of the types of responses students might produce and what quality of their response will be used to score the response. Once the task features are specified, we develop tasks to assess CT practices using these as a guide. We make sure the tasks comprise a mix of programming-construct-independent ones, and ones that are in the context of the programming language used in the curriculum. During our presentation, we will describe how ECD principles were used to design assessment tasks for CT Practices that are being used in a pilot program (*CoolThink@JC*) for upper primary students in Hong Kong.

Multifaceted views on CT learning through "Systems of Assessment"

Shuchi Grover

"Deeper learning" (Pellegrino and Hilton, 2012), seen as an imperative for helping students develop robust, transferable knowledge and skills for the 21st century, acknowledges the cognitive, intrapersonal, and interpersonal dimensions of learning. Barron and Darling-Hammond (2008) contend that robust assessments for meaningful learning must include: (1) intellectually ambitious performance assessments that require application of desired concepts and skills in disciplined ways; (2) rubrics that define what constitutes good work; and (3) frequent formative assessments to guide feedback to students and teachers' instructional decisions. Conley and Darling-Hammond (2013) assert that in addition to assessments that measure key subject matter concepts, assessments for deeper learning must measure both higher-order cognitive skills and abilities such as complex problem solving, planning, reflection, collaboration, and communication. These assertions imply the need for multiple measures or "systems of assessment" for CT (Grover, 2017) that are complementary, encourage and reflect deeper learning, and contribute to a comprehensive picture of student learning.

This presentation describes the systems of assessment for assessing algorithmic thinking and CT skills designed for an introductory computer science course for middle school students. These include open-ended and directed programming assignments with accompanying rubrics, innovative programming exercises inspired by Parson's puzzles (Denny et al., 2008), low-stakes quizzes for formative assessment (targeting individual concepts and constructs) with auto-grading and feedback, a summative assessment with MC and open-response items, final project of students' choosing, final project presentation to the whole class along with individual written student reflections and a shared "studio" of students' final projects, "artifact-based interviews" (Barron et al., 2002) around their final projects, and open-ended responses to questions related to identity and interest.

This presentation also follows on work that represents a refinement of how we can design programming tasks specifically aimed at measuring CT Practices or CTP (Grover, Basu, & Bienkowski, 2017). For this we are guided by assessment design patterns of CTP as mapped out by Bienkowski, et al. (2015) that employ Evidence-Centered Design (ECD; Mislevy & Haertel, 2006), a principled framework for assessment design. In particular, we designed assessment tasks to elicit evidence about the specific CTP such as (but not limited to), the ability to— use predefined methods to achieve a goal, create a generalized solution to a problem (as opposed to hard coding to meet a very specific case), identify the appropriate place in code to modify given a new specification, design an abstraction to represent a problem or solution, implement testing and debugging methods to test and fix a computational solution, use programming constructs including conditionals, Boolean logic expressions, loops, parallel execution in algorithmic instructions. We will describe the design of two such tasks and findings from our empirical studies involving their use in three high school classrooms using the Exploring CS curriculum.

Assessing youth's computational thinking in the context of modeling and simulation

Irene Lee and Eric Klopfer

Dave Moursund (2009) suggests that "the underlying idea in computational thinking is developing models and simulations of problems that one is trying to study and solve." At the MIT Scheller Teacher Education program we examine how these ways of thinking take shape for middle school youth specifically in the context of modeling and simulation. The terms of *abstraction*, *automation*, and *analysis* (Cuny, Snyder, and Wing, 2010) are useful for understanding how youth approach novel problems using Computational Thinking (CT) in modeling and simulation. While we adopt these principles, we adapt them to this context. Abstraction is "the process of generalizing from specific instances." In modeling and simulation abstraction is a necessary process for working from a specific example to building a more general model. Automation is a labor saving process in which a computer is instructed to execute a set of repetitive tasks quickly and efficiently compared to the processing power of a human. In this light, computer programs are "automations of abstractions." In modeling and simulation this takes the form of being able to rapidly run many iterations of a model. Analysis is a reflective practice that refers to the validation of whether the abstractions made were correct. In modeling and simulation one needs to think about how to collect and analyze the data that comes out and what are the right comparisons from real data to make against the data.

This paper aims to contribute to the discussion and development of tools used to assess youth's computational thinking by sharing methods including an exploratory method used in Project GUTS: Growing Up Thinking Scientifically. We present a method to assess near transference of computational thinking as an approach to problem investigation and problem-solving when faced with a new scenario.

Supporting metacognitive awareness of the process of making: Portfolio assessment in high school e-textiles classrooms

Debora Lui, Gayithri Jayathirta, Mia Shaw, Yasmin B. Kafai, and Deborah A. Fields

In recent years, the Maker Movement has drawn much attention from educators and policy makers because of its potential for rich learning in solving problems that arise throughout the process of digital fabrication. Yet assessing learning in making has proven a challenge, in part because student creations are inherently personal and distinct. Researchers have taken several different directions to assess student learning in making, each with affordances and limitations. Content tests and surveys allow documentation of learning or interest development across multiple classrooms (Tofel-Grehl et al, 2017) but tend to limit what counts as learning to standards-based content, discounting the ongoing processes of learning. In contrast, case studies of student design processes and clinical interviews (e.g., Lee & Fields, 2017) show depth of learning and students' uptake of process-based practices but are time-consuming and limit student agency in shaping their own narratives of learning.

Here we share our initial efforts to develop assessments authentic to the context of making e-textiles but scalable to multiple classrooms. In 2017, we piloted an eight-week curriculum where students created e-textiles (programmable circuits sewn with conductive thread) as part of the year-long Exploring Computer Science course. The unit included a final portfolio assignment where students reflected on and presented their own learning. The portfolio assignment served two main goals: 1) study student learning in a manner authentic to the creative making process, 2) support students' metacognitive awareness as a type of equity-based learning (Darling-Hammond, 2008). Portfolios included three elements: a video summarizing how the final project worked, a reflection on a challenge or revision in making the final project, and a reflection on learning in the e-textiles unit as a whole. Three teachers from diverse schools in an urban center of California piloted the unit. Data consisted of four types: portfolio collection, interviews with select student focus groups, interviews with teachers reflecting on the portfolios, and observation of two portfolio-creation days in each classroom.

Our analysis revealed the types of problems students identified in their portfolios as well students' reflections on their own learning. We identified the degree to which students were explicit (versus generic) in discussing the problems they faced. Further, students' reflections on creating the portfolios revealed a new cognizance of their own learning process, pointing to the potential for the portfolios to support metacognitive awareness, though this was not universally present. Finally, teachers' reflections demonstrated different interpretations of the value of portfolios for students' learning and as a classroom tool to assess their learning. In the full presentation we consider the utility of the portfolio assignment to students, teachers, and researchers as well as the potential of this type of process-based portfolio to other types of making scenarios.

Combining data mining and qualitative analysis to reveal learners' computational practices in open-ended student-driven curricula

Mike Tissenbaum, Mark Sherman, and Josh Sheldon

There is increasing advocacy for the need to emphasize the processes of learning and creating in addition to final products (Vossughi et al., 2013). This is particularly true in computing education, as learners' growth as computational thinkers often occurs during the processes of thinking and designing. The environments in which learners write their code offer a unique opportunity for understanding the processes, as many can capture traces of student program development through their incremental edit operations.

Learning analytics and data mining have been shown as fruitful for unpacking this progression, providing insight into computational thinking education and problem-solving approaches (Berland et al., 2013). While learning analytics and data mining can help reveal these patterns in student learning, they cannot detect what is happening "above the screen", meaning that inferences drawn can miss many of the important causes of these patterns. On the other hand, qualitative approaches have been successful in revealing nuanced computational thinking patterns that learners engage in "off of the screen," particularly in collaborative settings (Bienkowski et al., 2015; Grover & Pea, 2013). Yet, these approaches can require intensive labor and may miss commonalities in student programming. In particular, they cannot capture differences in "off of the screen" enactment patterns. Such differences could help us to understand the variations between groups of learners.

This paper offers an approach that combines these two methods for assessing learners' computational thinking practices in open-ended learner-driven computing curricula. We looked at a group of 23 youth (ages 15-19) engaged in a 5-week summer camp to build solutions that solved personally relevant problems using MIT App Inventor (a blocks-based programming environment for building fully-functional mobile apps). Using two data mining approaches- block selection mining and moments of flailing (Sherman, 2017), we highlighted moments in the learners' app building for analysis to understand their computational thinking practices. We then applied a coding scheme derived from the computational thinking practices framework developed by Bienkowski et al. (2015) to analyze discourse among group members. Using this analysis, we can discuss the affordances of blending learning analytics with qualitative coding. This helps us understand the nuanced ways students engage in collaborative computational thinking during open-ended projects.

Valuing and revealing networks of engagement around computational making

Nichole Pinkard, Caitlin K. Martin, and Sheena Erete

Multidimensional views of computational thinking (e.g. diSessa, 2001) that pay attention to the larger learning ecology (Barron, 2006) and social networks are especially relevant to encouraging and supporting participation from underrepresented groups, such as women and non-dominant populations. Repeated studies have found that pervasiveness of social orientations, including negative stereotypes about girls and non- dominant students' STEM abilities and interests and a lack of sense of community and belonging within STEM classrooms and fields (e.g. Margolis & Fisher, 2002) impact interest, engagement, and ultimately participation.

In this presentation, we share strategies for assessing computational thinking within Digital Youth Divas, an out-of-school program for middle school girls designed to build and support a social ecosystem of computational making in communities that have been underrepresented in engineering and computing fields.

There is a growing understanding that supplementing traditional approaches to STEM assessment with socially-grounded strategies is critical, especially to better understand and design for unique populations of learners. These strategies include biographical and narrative approaches as well as social network representations. In addition to traditional pre-post knowledge measures, we look deeply into the two environments girls inhabit in the program, across different levels of analysis. In the online space, where girls access activities, submit work, and give and receive feedback, we conduct *digital ethnographies* to qualitatively document online activity and interactions to better understand *social learning analytics* of platform user trace log data representing all girls in the program. In the face-to-face space, where they work through projects together, we document a focal classroom through *field notes and photographs* and engage in *artifact-based reflection interviews* with individuals in that classroom using projects as shared references to discuss work process, connections to people or practice in and out of the program, and plans for learning more.

Our findings indicate that girls connected aspects of their computational projects with people in their networks at home and school, and shared, designed for, and taught peers, siblings, and younger members of their community. These moves engaged family members in the computational activities the girls worked on, playing roles of audience, learners, and collaborators. This internal advocacy could serve to broaden participation in communities as these young girls from underrepresented populations take on the role of visible computational makers in their personal networks.

A case study of observing and identifying computational thinking practices in the wild

Eni Mustafaraj and Clara Sorensen

Many researchers have proposed computational thinking frameworks composed of different components. One such example is that of (Brennan & Resnick, 2012), who identify concepts, practices, and perspectives as possible components. How these components interact with one-another is an area of active research, but one credible hypothesis is that learners are engaged simultaneously in a kind of feedback loop that encompasses all these components. Concretely, when they are learning about specific concepts, such as expressions and statements, they are also learning about the practices of testing and debugging.

Given that this is a continuous feedback loop that leads to deeper understanding of concepts and practices, it is important to study the learning trajectories as students move from one level of understanding to another. One would typically expect that middle school students who are learning how to solve problems through computation are at a very different level of understanding than college students who have completed two programming courses. But college students are also still learning, and although they know how to program, they struggle considerably when asked to solve real-world problems through computation. These struggles point to gaps in their computational thinking, which need to be exposed and recognized, in order for them to deepen their understanding. Our research interest is in studying whether we can observe and capture "in-the-wild" gaps in computational thinking of advanced learners, so that such gaps can be made explicit to learners. With "in-the-wild", we mean the everyday learning process that happens in the classes that students take.

This presentation will discuss findings from studying the learning of college students in a data science class. Data science, with its current focus on large amounts of automatically captured data, provides a rich context for observing computational thinking in practice, because it offers a wide range of problems that are new and challenging, but also meaningful to explore, something that motivates learners. For example, one of the datasets used in the course were the personal email inboxes of students (each student analyzed her own inbox). When dealing with this dataset, students were motivated by the curiosity about what it can reveal about their habits (e.g., am I contacting my professors more from one year to the other?), but also by the potential for changing their habits (can I build a model that it will predict the rate of incoming emails, so that I don't constantly check my email).

Finding answers to these questions is not trivial. Students have to learn new libraries to become efficient in data analysis, and because there is no given algorithms for each problem, they need to be creative, work incrementally, and iterate often, all practices inherent to computational thinking. The computing environment that they were using, the Jupyter notebooks, allowed us to capture automatically and without any instrumentation cost, practices of incremental and iterative problem solving. A quantitative analysis of the notebooks from their class projects revealed two distinct patterns of learners: "explorers" versus "goal accomplishers".

Computational rethinking – Applying CT in new contexts

Will Temple and R. Benjamin Shapiro

The emergent computational landscape is inhabited by large-scale communications systems that students interact with *constantly*. In response, we need how new paradigms for Computer Science that expose networked communications technologies to students as platforms for play, exploration, and learning. Within these new systems, basic assumptions about what's important for beginning students to learn (e.g. loops in Brennan & Resnick, 2012) diverge from the platforms of yesterday. Our own BlockyTalky project shows that students, even at a young age, reason competently about networked systems when provided appropriate abstractions (Shapiro *et al.*, 2017). Online systems such as MIT App Inventor empower many students—even those with minimal experience—to create complex mobile technologies that enhance their lives and communities.

In an interdisciplinary context full of new notions about which computational ideas are the most powerful, *how will we know if our students are learning CS?* When we apply CT to other disciplines, we ought to orient our assessments towards the application domain, as conventional ideas about the necessity of particular concepts in fundamental CS may not suit the modern, diverse applications of CT. For example, our BlockyTalky WeJam music toolkit does not include generic looping constructs, since they are not generally useful in the design of distributed computer-music. An assessment in that context might examine how students create different topologies of networked systems and then use computational constructs such as message passing to explore the computer-music domain, to synchronize rhythms and melodies, and to support real-time control over musical parameters through tangible interfaces. In other words, we expect students to reason *in the application domain* using whatever computational constructs serve them.

The application of CT to music, as suggested above, affords an opportunity to assess the way that students model and implement musical phenomena in terms of computational processes. Using our BlockyTalky toolkit, students design musical compositions as a system of networked instruments. In this environment, we can assess students' uses of computational models and ability to design effective synchronization strategies by analyzing their output: a musical composition. Our toolkit deliberately requires students to organize their music into a networked system, allowing us to observe properties of the resulting music (do the notes play in the desired order or tempo, and at the same time?) and thereby learn about how our students built an understanding that utilizes the underlying CS as a model for their creation. By redefining our assessments of CT to emphasize utility to the application domain, we will empower students to *construct* the CT skills that are relevant to their creative goals.

References

Barron. B. (2006). Interest and self-sustained learning as catalysts of development: A learning ecologies perspective. *Human Development, 49*, 193-224.

Barron B., & Daring-Hammond, L. (2008). How can we teach for meaningful learning.? In Daring-Hammond, L., Barron, B., Pearson, P. D., Schoenfeld, A. H., Stage, E. K., Zimmerman, T. D., Cervetti, G. N., & Tilson, J. L. Powerful learning: What we know about teaching for understanding. Jossey-Bass.

Barron, B., Martin, C., Roberts, E., Osipovich, A., & Ross, M. (2002). Assisting and assessing the development of technological fluencies: Insights from a project- based approach to teaching computer science. In Proceedings of the Conference on Computer Support for Collaborative Learning (pp. 668-669). ISLS.

Basu, S., Kinnebrew, J. S., & Biswas, G. (2014, June). Assessing student performance in a computational-thinking based science learning environment. In *International Conference on Intelligent Tutoring Systems* (pp. 476-481). Springer, Cham.

Berland, M., Martin, T., Benton, T., Petrick Smith, C., & Davis, D. (2013). Using learning analytics to understand the learning pathways of novice programmers. Journal of the Learning Sciences, 22(4).

Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A First Look*. Menlo Park, CA: SRI International.

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, Vancouver, Canada*(pp. 1-25).

Conley, D. T., & Darling-Hammond, L. (2013). Creating Systems of Assessment for Deeper Learning.
Cuny, J., Snyder, L., and Wing, J. (2010). Computational Thinking: A Definition. (in press)
Darling-Hammond, L. 2008. *Powerful learning: What we know about teaching for understanding*. Jossey-Bass.
Denny, P., Luxton-Reilly, A., & Simon, B. (2008). Evaluating a new exam question: Parsons problems. In Proceedings of the fourth international workshop on Computing education research (pp. 113-124).
DiSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.
Erete, S., Martin, C. K., & Pinkard, N. (2017). Digital Youth Divas: A Program Model for Increasing Knowledge, Confidence, and Perceptions of Fit in STEM amongst Black and Brown Middle School Girls. In *Moving Students of Color from Consumers to Producers of Technology* (pp. 152-173). IGI.
Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43.
Grover, S. (2017). Assessing Algorithmic and Computational Thinking in K-12: Lessons from a Middle School Classroom. In Emerging Research, Practice, and Policy on Computational Thinking (pp. 269-288). Springer International Publishing.
Grover, S. Basu, S., & Bienkowski, M. (2017). Designing Programming Tasks for Measuring Computational Thinking. Presented at the Annual Meeting of the American Educational Research Association.
Lee, V. R. & Fields, D. A. (2017). Changes in undergraduate student competences in the areas of circuitry, crafting, and computation after a course using e-textiles. *International Journal of Information and Learning Technology*, 34(5), 372-384.
Margolis, J. and Fisher, A. (2002). *Unlocking the clubhouse: women in computing*. Cambridge, MA: MIT Press.
Mislevy, R.J. and Haertel, G.D. 2006. Implications of Evidence-Centered Design for Educational Testing. Educational Measurement: Issues and Practice 25, 4 (December 2006), 6–20.
Moursund, D. (2009). Computational Thinking. IAE-pedia. Available online at *http://iae-pedia.org/Computational_Thinking*. Accessed August 8, 2010.
National Academies of Science. (2010). Report of a workshop on the scope and nature of computational thinking. Washington DC: National Academies Press.
Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing Computational Thinking Into High School Mathematics and Science Classrooms. In Proceedings of ICLS 2016 (pp. 705–712). Singapore.
Pellegrino, J. W. & Hilton, M. L. (2013). Education for life and work: Developing transferable knowledge and skills in the 21st century. National Academies Press.
Sherman, M. (2017, April). Detecting student progress during program activities by analyzing edit operations on their blocks-based programs. Ph.D. dissertation, University of Massachusetts Lowell.
Shum, S. B., & Ferguson, R. (2012). Social learning analytics. *Jrnl of educational technology & society*, *15*, 3
Snow, E., Rutstein, D., Bienkowski, M., & Xu, Y. (2017, August). Principled Assessment of Student Learning in High School Computer Science. In Proceedings of the 2017 ACM Conference on International Computing Education Research(pp. 209-216). ACM.
Tissenbaum, M., Sheldon, J., Soep, L., Lee, C.H. Lao, N. (2017). Critical computational empowerment: Engaging youth as shapers of the digital future. *Proceedings of the IEEE Global Engineering Education Conference*, Athens Greece, April, 1705-1708,
Tofel-Grehl, C., Fields, D. A., Searle, K., Maahs-Fladung, C., Feldon, D., Gu, G., & Sun, V. (2017). Electrifying engagement in middle school science class: Improving student interest through e-textiles. *Journal of Science Education and Technology*.
Vossoughi, S., Escudé, M., Kong, F., & Hooper, P. (2013, October). Tinkering, learning & equity in the after-school setting. In annual FabLearn conference. Palo Alto, CA: Stanford University.
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.
Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., & Wilensky, U. (2014). Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms. In D. Reidsma, I. Choi, & R. Bargar (Eds.), Proceedings of Intelligent Technologies for Interactive Entertainment, Chicago, IL, USA (pp. 22–25)..
Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering Computational Literacy in Science Classrooms. Communications of the ACM, 57(8), 24–28.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.