Working Group 6: Advancing computational thinking in 21st century learning

Chris Dede, Harvard University

Punya Mishra, Michigan State University

Joke Voogt, University of Twente

Computational thinking is seen as a skill set that every child needs to develop. It is related with a number of other 21st century competencies (problem solving, critical thinking, productivity, and creativity). In EDUsummIT 2013, we aim to advance the discussion about computational thinking by focusing its core competencies, its relation with and distinction from other 21st century competences, and its place in the curriculum.

Introduction

The theme of TWG 6, Advancing Computational Thinking, is a specific follow-up of the more general discussion held at the EDUsummIT 2011 on the importance and implementation of 21st century competences for teaching and learning in a digitally networked world (Voogt, Erstad, Mishra & Dede, 2011; Voogt, Erstad, Dede & Mishra, in press). "Computational thinking represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (Jeannette Wing, 2006, p. 33). Overall, Wing argues that this new competency should be added to every child's analytical ability as a vital ingredient of science, technology, engineering, and mathematics (STEM) learning. Several professional bodies and think tanks in the US, the UK, and the Netherlands have called for more attention to computational thinking in the curriculum. In EDUsummIT 2013, we aim to advance this discussion by focusing on the core components of computational thinking, its relation with and distinction from other 21st century competences, and its place in the curriculum.

Background

In March, 2006, Jeaneatte Wing published an influential article, "Computational Thinking," in the *Journal of the Association for Computing Machinery*. Wing posited that "computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science… It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (pg. 33). Overall, the article argues that this new competency should be added to every child's analytical ability as a vital ingredient of science, technology, engineering, and mathematics (STEM) learning.

In the UK, the Royal Society (2012) published an influential report about the importance of computational thinking. Computational Thinking was defined in their report as " the process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes" (p. 29). The Royal Society advocated for more attention to Computer Science in the primary and secondary school curriculum. The Royal Society report was embraced by the Royal Netherlands Academy of Arts and Sciences (2013) in their plea for more attention to digital literacy in the secondary school curriculum.

This focus on computational thinking is not new. Its roots go back, most significantly, to Papert's work on LOGO programming language and the idea of the computer being the children's machine that would allow them to develop procedural thinking through programming (Papert 1980, 1991). The advent of the Internet, increasing bandwidth, the convergence of digital media across platforms and devices, the dot com economy, the rise of gaming and social media, the DIY movement, the availability of better computational tools and software, and the attention to big data all have led to a resurgence in interest in computational thinking. Recent work in the field is deeply inspired by Papert's pioneering work but also has a distinct 21st century flavor—building on the advances that have occurred in recent decades.

That said, as Grover and Pea (2013) discuss, attempts to define in detail what "thinking like a computer scientist" means have proven quite problematic. Building on two National Research Council workshops (2010, 2011) and considerable discussion by various professional societies, the College Board and the National Science Foundation are developing a high school Computer Science Principles course based on practices of computational thinking. In the Advanced Placement Computer Science Principles Draft Curriculum Framework (2013), six computational thinking practices are identified (pp. 7-8):

Computational Thinking Practices

P1: Connecting computing
Developments in computing have far-reaching effects on society and have led to significant innovations. These developments have implications for individuals, society, commercial markets, and innovation. Students in this course study these effects and connections, and they learn to draw connections between different computing concepts. Students are expected to:

- Identify impacts of computing;
- Describe connections between people and computing; and
- Explain connections between computing concepts.

P2: Developing computational artifacts
Computing is a creative discipline in which the creation takes many forms, ranging from remixing digital music and generating animations to developing websites, writing programs, and more. Students in this course engage in the creative aspects of computing by designing and developing interesting computational artifacts, as well as by applying computing techniques to creatively solve problems. Students are expected to:

- Create an artifact with a practical, personal, or societal intent;
- Select appropriate techniques to develop a computational artifact; and
- Use appropriate algorithmic and information-management principles.

P3: Abstracting
Computational thinking requires understanding and applying abstraction at multiple levels ranging from privacy in social networking applications, to logic gates and bits, to the human genome project, and more. Students in this course use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity. Students are expected to:

- Explain how data, information, or knowledge are represented for computational use;
- Explain how abstractions are used in computation or modeling;
- Identify abstractions; and
- Describe modeling in a computational context.

P4: Analyzing problems and artifacts
The results and artifacts of computation, and the computational techniques and strategies that generate them, can be understood both intrinsically for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic, mathematical, pragmatic, and other criteria. Students in this course design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work that others have produced. Students are expected to:

- Evaluate a proposed solution to a problem;
- Locate and correct errors;
- Explain how an artifact functions; and
- Justify appropriateness and correctness.

P5: Communicating
Students in this course describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts. Communication includes written and oral descriptions supported by graphs, visualizations, and computational analysis. Students are expected to:

- Explain the meaning of a result in context;
- Describe computation with accurate and precise language, notation, or visualizations; and
- Summarize the purpose of a computational artifact.

P6: Collaborating
Innovation can occur when people work together or independently. People working collaboratively can often achieve more than individuals working alone. Students in this course collaborate in a number of activities, including investigation of questions using data sets and in the production of computational artifacts. Students are expected to:

- Collaborate with another student in solving a computational problem;
- Collaborate with another student in producing an artifact; and
- Collaborate at a large scale.

Clearly, many problems with articulating the nature of "computational thinking" remain, despite the detail in these competency-based definitions. For example, the skill of collaboration is hardly unique to computational thinking, and examples such as "collaborating with another student to produce an artifact" could equally well apply to engineering. As another illustration, competencies such as "select appropriate techniques to develop a

computational artifact" and "use appropriate algorithmic and information-management principles" beg the issue on precisely what thinking skills are involved.

A core issue involves how to separate the cognitive activity of computational thinking from the action of merely working on a computer or other digital device. For instance, word processing and/or creating webpages do use digital technologies; however, it is unclear that these involve the kinds of conceptualization that are unique to computational thinking. So, a deeper analysis is needed of the kinds of thought that are key to computational thinking and how they are distinct from other cognitive thinking skills. A focus on problem solving through "seeking algorithmic approaches to problem domains; a readiness to move between differing levels of abstraction and representation; familiarity with decomposition; separation of concerns; and modularity" (Barr & Stephenson, 2011, p. 49-50) can be argued is unique to computational thinking. In fact some even assert that computational thinking is an approach that does not necessarily need programming of computers, but rather is an approach to problem solving that uses strategies such as algorithms, abstraction and debugging (Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011). Along the same lines, Bundy (2007) posited that the ability to think computationally is essential to conceptual understanding in every discipline, through the processes of problem solving and algorithmic thinking.

Mishra, Yadav, and the Deep-Play group (2013) have argued that computational thinking goes beyond typical human computer interactions, in which humans initiate the actions that are then computed by the machine. The new forms of expression that are emerging today have significant implications for how we engage and interact with machines. "Human creativity," they argue can be "augmented by computational thinking, in particular the automation of problem solving and algorithmic thinking."

Grover and Pea (ibid) assert that empirical investigations are a good way of moving forward to resolve these definitional challenges. They recommend building on research from the 1980s on how children and novices learn computational concepts to develop a new set of studies on learning trajectories in the development of computational thinking skills, computing as a medium for teaching other subjects, and dispositions for, attitudes toward, and stereotypes concerning computational thinking.

Finally, any discussion of computational thinking must factor in the issue of human knowledge, expertise, and intuition. As a leader in big data analysis recently suggested, success in using computational thinking requires knowledge not just of computer science and mathematics but also imaginative capacities like innovative thinking and "a deep, wide ranging curiosity" (quoted in Lohr, 2012).

Issues/unresolved questions/concerns

1. An important issue in integrating computational thinking in the curriculum is delineating its boundaries with respect to other disciplines and the other 21st century competences. Some (Royal Society, 2012, Royal Netherlands Academy of Arts and Sciences, 2013) position computational thinking in the Computer Science curriculum The Royal Society, for instance, stated, **"**Every child should have the opportunity to learn concepts and principles from Computing (including Computer Science and Information Technology) from the beginning of primary education onwards, and by age 14 should be able to choose to study towards a recognized qualification in these areas" (p.44). Others (e.g., Hemmendinger, 2010) argued that the goal of teaching computational thinking is not for everyone to think like a computer scientist, but instead, "it is to teach them how to think

like an economist, a physicist, an artist, and to understand how to use computation to solve their problems, to create, and to discover new questions that can fruitfully be explored" (p. 6). It involves developing ways of thinking that allow learners to use computational tools in creative ways within the disciplines. Thus, delineation of computational thinking's boundaries with other fields is clearly an area that needs greater attention.

2. As an alternative way to formulate this challenge, Fishman and Dede (in preparation) relate computational thinking research issues to the larger context of learners informally engaged as makers and creators (including Scratch programming, DIY digital textiles, and robotics competitions). They argue that computational thinking may span more than the types of skills used in computer science, so using a discipline-centered model may be off-target.
3. Another important issue is that of seeing how computational thinking can help students develop their creativity. It has been argued that computational thinking can foster creativity, as one important 21st century competency, by enabling students not only to be consumers of technology, but also to build tools that can have significant impact on society. As discussed above, this emphasis on creativity can be seen in a new computer science course called *CS Principles*, currently being developed and piloted by the College Board. The core argument for including creativity in this mix is that computing not only extends traditional forms of human expression, but also allows the creation of new forms of expression (The College Board, 2012).

Overall, much remains to be done in defining computational thinking, studying its characteristics, and establishing its role and value in STEM preparation and in life.

Recommended additional reading

- 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.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing, 1*(2), 67-69.
- College Board. (2013). AP computer science principles draft curriculum framework. New York, NY: College Board. http://www.csprinciples.org/home/about-the-project/docs/csp-cf-2013.pdf?attredirects=0&d=1
- Fishman, B., & Dede, C. (in preparation). Teaching and technology: New tools for new times. In D. Gitomer & C. Bell (Eds.), *Handbook of Research on Teaching, 5th Edition* (American Educational Research Association). New York, NY: Springer.
- Grover, S., & Pea, R. (2013), Computational thinking in K-12: A review of the state of the field. *Educational Researcher 42*(1), 38-43.
- Hemmendinger, D. (2010*). A please for modesty*. *ACM Inroads, 1*(2), 4-7.
- Koninklijke Nederlandse Academie van Wetenschappen (2013). *Digitale geletterdheid in het voortgezet onderwijs. Vaardigheden en attitudes voor de 21e eeuw* [Digital literacy in secondary education. Skills and attitudes for the 21st century]. Amsterdam: Koninklijke Nederlandse Academie van Wetenschappen. https://www.knaw.nl/nl/adviezen/adviezen-en-verkenningen/recent-afgeronde-adviezen/digitale-geletterdheid-in-het-voortgezet-onderwijs
- Mishra, P., Yadav, A., & the Deep-Play Research Group (2013). Of Art and Algorithms. *Tech Trends, (57)* 3. p. 10-14.
- National Research Council. (2010). Committee for the Workshops on Computational Thinking: *Report of a workshop on the scope and nature of computational thinking*.

Washington, DC: National Academies Press. http://www.nap.edu/catalog.php?record_id=12840
- National Research Council. (2011). Committee for the Workshops on Computational Thinking: *Report of a workshop of pedagogical aspects of computational thinking*. Washington, DC: National Academies Press. http://www.nap.edu/catalog.php?record_id=13170
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism*. (pp. 1–11). Norwood, NJ: Ablex.
- The College Board (2012). Computational thinking practices and big ideas, key concepts, and supporting concepts. Retrieved from http://www.csprinciples.org/home/about-the-project.
- The Royal Society (2012). *Shut down or restart? The way forward for computing in UK schools.* London: The Royal Society. http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf
- Voogt, J., Erstad, O., Mishra, P., & Dede, C. (2011). *TWG6 21st century learning – expanded brief paper.* EDUsummIT 2011, Paris.
- http://www.edusummit.nl/fileadmin/contentelementen/kennisnet/EDUSummIT/Document en/2011/6_EDUsummIT_2011_21st_century_learning_expanded_brief_paper.pdf
- Voogt, J., Erstad, E., Dede, C., & Mishra, P. (in press). Challenges to Learning and Schooling in the Digital Networked World of the 21st Century. *Journal of Computer Assisted Learning.*
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–36.
- Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). *Introducing computational thinking in education courses*. In Proceedings of ACM Special Interest Group on Computer Science Education, Dallas, TX.