Computational thinking education: Issues and challenges

Charoula Angeli, University of Cyprus
Michail Giannakos, Norwegian University of Science and Technology

Abstract

Computational Thinking is a term applied to describe the increasing attention on students' knowledge development about designing computational solutions to problems, algorithmic thinking, and coding. It focuses on skills children develop from practicing programming and algorithms, and enables the development of qualities such as abstract thinking, problem solving, pattern recognition, and logical reasoning. Contemporary educational and infrastructural developments, like "CS for All" (https://www.csforall.org/), ISTE's Standards for Students in Computational Thinking (https://www.iste.org/explore/Solutions/Computational-thinking-for-all?articleid=152), Computer Science Teachers Association's Concepts of Computational Thinking (http://advocate.csteachers.org/2014/09/15/computational-thinking-and-beyond/), and the appearance of tools such as robotics, 3D printing, microprocessors, and intuitive programming languages posit Computational Thinking as a very promising area to support these learning competences. In this special issue of *Computers in Human Behavior*, the Editors report four studies conducted by interdisciplinary teams. This introduction to the special issue also draws attention to the great potential and need for further research in the area of Computational Thinking Education to engage students in meaningful learning so as to develop useful thinking skills and digital competences. Finally, the Editors of this special issue propose directions for future research and practice in Computational Thinking Education.

Keywords: Computational Thinking; Digital competences; Coding; Technological fluency; Algorithmic thinking; Robotics

1. Introduction

Computational Thinking (CT), a term used since the 1950s, describes the notion of using structured thinking or algorithmic thinking to produce appropriate output to a given input (Denning, 2009). Recent efforts to revitalize the importance of CT aim at democratizing computing knowledge as an important body of knowledge that learners need to have in order to cope well with the challenges of the 21st century. In 2006, Wing relaunched the term and interest in the area by defining CT as a process that involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science (Wing, 2006). This definition has been adopted widely due to its generic nature, but also created a need for a more specific definition that can be used in CT Education (CSTA & ISTE, 2011, Selby & Woollard, 2014).

During the last years, there has been an increasing interest about CT Education in K-12 schools, and its role in children's acquisition of thinking skills and digital competences. In accordance with this need, computational thinking and coding have, in recent years, become an integral part of school curricula in many countries. Estonia, Israel, Finland, and the United Kingdom are only a few examples of the growing efforts of governments to integrate coding as a new literacy and to support students in creative problem-solving tasks (Hubwieser, Giannakos, Berges, Brinda et al., 2015). In addition, Computer Science Teachers Association (CSTA, 2011), International Society for Technology in Education (CSTA & ISTE, 2011), Cyber Innovation Center (https://cyberinnovationcenter.org/), and National Math and Science Initiative (https://www.nms.org/) have developed conceptual guidelines for CT Education. Similarly, organizations such as "code.org," and "codeacademy.com" offer learning environments to promote coding activities and CT Education.

While it is well accepted in the literature that CT involves a number of skills, like problem decomposition (breaking down complex problems to simpler ones), developing algorithms (step-by-step solutions to problems), and abstraction, there is still limited evidence around the several issues and challenges someone needs to be aware of in order to design appropriate learning experiences for CT competences. In this contribution, the editors present four research studies covering different aspects of CT research, and, discuss challenges for both research and practice in CT Education as well as raising important new research questions for the researchers in the field.

2. The contributions in this special issue

While CT is an area of growing significance, scholarly work on CT is emerging, both conceptually and empirically. In response to the need for accelerating research foundations and developments in CT Education, *Computers in Human Behavior* presents a special issue that disseminates the latest research findings. The special issue consists of four contributions addressing the topic of CT from different perspectives and disciplinary backgrounds as well as covering different research areas and needs. The articles provide insights about: a) the importance of metaphors in CT education, b) putting into practice CT activities to empower both girls and boys, c) the importance of employing empirical experimentation in furthering CT

Education research and d) the development of young children's CT skills using scaffolds and educational robotics.

2.1. Embodied Metaphors for Computing Education

In the first article, Manches, McKenna, Rajendran, and Robertson (2019, this issue) investigated elementary computing concepts using metaphors through the lens of Embodied Cognition. A metaphor is a figure of speech that describes an object or action that is not literally true, but helps to explain an idea or make a comparison. Conceptual metaphors are extremely important in learning sciences (e.g., energy transfer, thermodynamics, and mathematics), as they offer an explanation of our ability to think and reason about abstract concepts. Manches et al.'s analysis showed that participants drew upon two overarching embodied metaphors in their explanations, namely: a) computing constructs as physical objects, in which participants simulated manipulating physical objects (e.g., pinching) when referring to a range of computing concepts, and b) computing processes as motions along a path, whereby participants moved their hands along one of three body-based axes when referring to temporal sequences. The authors concluded that embodiment might shape students and teachers' CT understanding and learning. In addition, there may well be other examples of integrated metaphors that can be used to communicate the meaning of the construct of CT, and such representations will allow us to better support CT teaching and learning techniques as well as the development of technologies and interfaces (e.g., embodied interfaces and interactions) for the teaching of CT.

2.2. The Use of Metaphors to Introduce Children to Programming

The second article by Pérez-Marín, Hijón-Neira, Bacelo, and Pizarro (2019, this issue) further expanded on this notion of using metaphors to teach CT, within the context of computer programming. The authors put into practice a methodology called MECOPROG using metaphors, such as, recipe/program, pantry/memory, and boxes/variables, to teach programming following an empirical experiment with 132 primary education students between 9 and 12 years of age. Their findings validated that coupling the use of metaphors with a block-based programming environment (e.g., Scratch) has the potential to improve CT knowledge acquisition in primary education.

2.3. Learning Strategies as a Pathway for Fostering CT

In the third study, Papavlasopoulou, Sharma, and Giannakos (2019, this issue) designed and evaluated a workshop for K-12 students to learn how to code. The design and development of activities that successfully scaffolded CT concepts and motivated both boys and girls proved to be critical for the teaching and learning of CT skills. In this study, the goal was to examine differences between boys and girls (if any) using eye-tracking as an objective measure and triangulating the findings with qualitative data coming from children's interviews. The results of their study showed no statistically significant difference between girls' and boys' gaze and learning gain during the CT activity. Interestingly, the qualitative data showed differences in the strategies and implemented practices during coding, and in perceptions about those CT activities. The results provided objective evidence that female students did not lack in competences or behavior (based on their gaze data) compared to boys, but simply that they had a different approach/strategy during CT activities and different perspectives about coding. Thus, it's important if this approach is taken into consideration during the design of CT activities and assist girls in mastering CT concepts.

2.4. Children's Computational Thinking with Educational Robotics: An Interaction Effect between Gender and Scaffolding Strategy

The fourth study by Angeli and Valanides (2019, this issue) examined the effects of learning with Bee-Bot, a floor programmable robot, on young boys' and girls' computational thinking. It was hypothesized that scaffolding would play a significant role in the development of children's computational thinking skills during learning with Bee-Bot, because Bee-Bot does not provide a visual representation of the commands children use to program it. The two scaffolding techniques were designed taking into consideration gender differences, anticipating that both genders would benefit from at least one of the two techniques. The results showed statistically significant learning gains between the initial and final assessment of children's computational thinking skills. Also, according to the findings, while both boys and girls benefited from the scaffolding techniques, a statistically significant interaction effect was detected between gender and scaffolding strategy showing that boys benefited more from the individualistic, kinesthetic, spatially-oriented, and manipulative-based activity with the cards, while girls benefited more from the collaborative writing activity. The research contributes to the body of knowledge that can be used to inform the teaching of computational thinking skills. In addition, the study has practical significance for curriculum developers, instructional leaders, and classroom teachers, as they can use the results of this study to design curricula and classroom activities with a focus on the broader set of computational thinking skills, and not only coding.

3. Challenges in Computational Thinking Education: Future Research Directions

The findings from the studies suggest that in order to adopt CT as a powerful educational concept, researchers need to invest further systematic research efforts in addressing several issues related to:

1. Defining CT competencies for each school grade level or students' developmental level

As the contributors in this special issue discussed, efforts have been made to define competencies, guidelines, and curricula for CT (e.g., CSTA, ISTE). What is currently missing from the literature is how CT skills, such as abstraction, problem decomposition, and data structures, might map to different abilities, grade level, disciplines, gender, and educational level. Thus, further work is needed in order to solve inconsistencies (Denning, 2017) and to develop and validate a robust theoretical conceptualization about the construct of CT.

2. The use of metaphors in teaching CT concepts efficiently and effectively

According to Manches et al. (2019, this issue) and Pérez-Marín et al. (2019, this issue) the use of learner-centered metaphors enhance students' understanding and learning of CT concepts. While this special issue provides preliminary evidence about the importance of metaphors in teaching and understanding CT, more research is needed in order to create more metaphors that can be used effectively in teaching students and teachers about CT concepts.

3. The use of pedagogical strategies and technologies in teaching CT

The articles by Papavlasopoulou et al. (2019, this issue) and Angeli and Valanides (2019, this isse) point to the need to scaffold students' learning during their engagement with CT activities, and, the importance of alignment between teaching activity and gender. Considering the fact that more and more student-friendly programming environments (e.g., Alice, Scratch, BlueJay, Greenfoot, Kodu), hardware materials (3D printers, educational robotics) and other initiatives (e.g., code.org, codeacademy.com) appear as means to promote CT Education, future research needs to be undertaken to investigate the interrelationship between CT skills and competencies, CT representations, CT activities, CT tools, and CT teaching practices.

4. Teacher CT professional development

For CT education to further develop, teachers need to be systematically prepared in terms of how to design CT learning activities, how to teach CT, how to assess CT, and how to use technologies to teach CT concepts. Thus, teacher professional development programs need to be implemented for in-service teachers, while at the same time teacher educators need to find ways to integrate the teaching of CT in their pre-service courses for the better preparation of pre-service teachers.

5. Assessment of CT competencies and skills

Lastly, as the articles in this special issue mentioned, the assessment of CT skills and competencies is well under-developed. Thus, there is a need for future research to identify ways about how CT can be assessed either as a holistic measure or as an array of sub-skills within the context of authentic problem-solving across all subjects and disciplines.

Accordingly, Figure 1 presents a five-step plan about how these five research areas can be addressed in future research studies. The five-step plan is presented as a cycle, because it is expected that through intense research and practice progress in each area will inform one another and evolve over time. The first step tackles the definition of CT competencies in order to provide a baseline and common language across different contexts (e.g., different countries, educational levels, school subjects, disciplines, etc.) about the concept of CT. The next step is that of creating powerful metaphors as a mechanism for transforming abstract CT concepts to more concrete and easier notions to understand. The third step is to research the effectiveness of pedagogies and technologies in enhancing and enabling the development of CT competencies. The fourth step focuses on the crucial issue of preparing teachers and instructors to teach CT as well as integrate appropriate technological tools to enable the teaching of CT in their respective teaching contexts.

Lastly, the fifth step deals with the measurement and assessment of CT competencies, an area of research that is currently in its infancy.

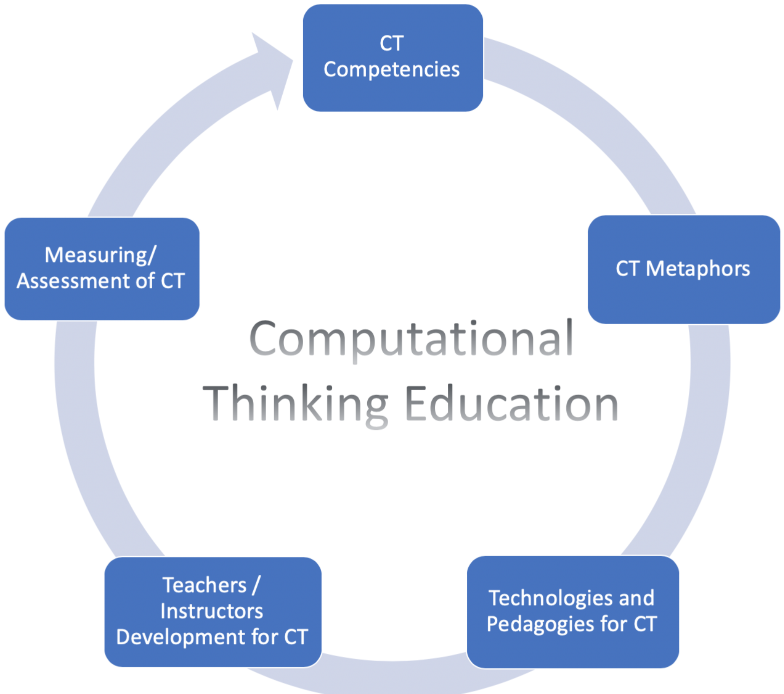

Figure 1. A five-step research plan for CT Education

Acknowledgements. We would like to thank the reviewers of this special issue for their timely reviews, and the Commissioning Editor of *Computers in Human Behavior* Paul A. Kirschner for his ongoing support and guidance during the preparation of the special issue. This work was partly supported by the COMnPLAY-Science project, under the European Commission's Horizon 2020 SwafS-11-2017 Program (Project Number: 787476).

References

Angeli, C., & Valanides, N. (2019, this issue). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*.

CSTA & ISTE. (2011).*Operational definition of computational thinking for K-12 education*. Available at: https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

CSTA. (2011). *Operational definition of computational thinking*. Available at: http://www.csta.acm.org/Curriculum/ sub/CurrFiles/CompThinkingFlyer.pdf

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33-39.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., ... & Jasute, E. (2015, July). A global snapshot of computer science education in K-12 schools. *In Proceedings of the 2015 ITiCSE on working group reports* (pp. 65-83). ACM.

Manches, A., McKenna, P. E., Rajendran, G., & Robertson, J. (2019, this issue). Identifying embodied metaphors for computing education. *Computers in Human Behavior*.

Papavlasopoulou, S., Sharma, K., & Giannakos, M. (2019, this issue). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*, 1-11.

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2019, this issue). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*.

Selby, C., & Woollard, J. (2014) Computational Thinking: The developing definitions. *In Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, *SIGCSE 2014*. ACM.

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