Book Review

Programming and Computational Thinking in Technology Education: Swedish and International Perspectives

Hallström, J. & de Vries, M. (Eds.) (2024). *Programming and Computational Thinking in Technology Education: Swedish and International Perspectives* (International Technology Education Studies, Vol. 20). Brill Academic Pub. ISBN: 978-90-04-68790-5 (Hardback), $172.00, 341 pages.

Preface

The growing importance of computational thinking in an increasingly digitalized world is conveyed through Wing's perspective on why every child should develop analytical abilities.

> Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking (Wing, 2006, p. 33).

For nearly two decades Wing's perspective has contributed to promoting widespread awareness of the importance of computational thinking in an increasingly digitalized world. Throughout these years the concept of computational thinking has evolved significantly, transitioning from its origins in manual computation to its contemporary application in solving complex problems. Historically, computational thinking was often associated with mathematical calculations, for example determining the values of sine or cosine. Today, however, it is understood as a set of mental skills and practices used to design computations that enable computers to perform tasks and to interpret the world as a network of information processes (Denning & Tedre, 2019).

Parkpoom Pengsuwan (parkpoom@vt.edu | ORCiD: 0009-0001-5387-5447) is a doctoral candidate in the Integrative STEM Education graduate program at Virginia Tech and the recipient of the Royal Thai Scholarship. He holds a position at the National Science and Technology Development Agency (NSTDA) in Thailand.

Sweden has recognized the critical importance of computational thinking and integrated this knowledge and content area into its national education curriculum, including the Swedish technology subject. The Swedish government emphasizes computational thinking as a way for students to develop problem-solving skills and translate ideas into creative actions. Similarly, within the K-12 educational system of the United States, technology education programs are viewed as a key platform for fostering computational thinking. Specifically, the International Technology and Engineering Educators Association (2020) states that "while other fields (e.g., computer science) are also engaged in computational thinking practices, the unique approach of technology and engineering is connected to the automation of physical devices and systems that result in, from, or through design" (p. 92). These shared international perspectives underscore the distinctive role technology education holds in leveraging design processes to address real-world problems, integrating computational thinking into classroom lessons, particularly through the application of programmed technological solutions. Moreover, these perspectives reflect a growing recognition that technology education serves as a natural home for the development of computational thinking.

Introduction

With a particular focus on Sweden, the edited book *Programming and Computational Thinking in Technology Education: Swedish and International Perspectives* delves into the integration of digital technology, programming, and computational thinking within the technology education curriculum at all grade levels. It provides a historical perspective on the need to incorporate programming into the Swedish curriculum, while concurrently exploring the philosophical and educational dimensions of teaching and learning programming. Readers gain insights into the progression of integration efforts, from the early development of computers to the internet era and the modern digital world. Throughout the book, authors of the 16 chapters emphasize the gradual shift toward leveraging technology education to cultivate computational thinking and equip students with digital competencies. Moreover, it highlights the dual nature of technology, its physical and functional aspects, through examination of the connections between technology, design thinking, systems thinking, and computational thinking. Furthermore, the book offers recommendations for pedagogical practices to enhance the teaching and learning of technology education and programming in schools.

Organization

This book is organized into three parts, with the first focusing on the definition, philosophy, and history of programming and computational thinking in the context of technology education. Chapters in Part 1 explore the significant role computers have played in education, particularly in Sweden, and examines

the factors that have elevated computational thinking as a vital 21st-century skill. Discussed through these chapters is how Sweden and other countries have incorporated programming and computational thinking (P&CT) into their curricula, emphasizing the role of technology education in fostering opportunities for teaching and learning these skills while addressing the broader context of integration into educational systems. Part 2 of the book narrows the focus to teacher perspectives on integrating P&CT into the Swedish curriculum. These chapters highlight views held by teachers regarding curricular content, the progression of learning, and their experiences in embedding digital competencies into technology education. Also addressed are the challenges teachers face and the resources available to support their efforts. Part 3 of the book provides recommendations for teaching and learning P&CT in technology education. Chapters included in Part 3 underscore the importance of helping students understand the dual nature of technology, using systems thinking to connect its physical and functional aspects, and applying computational thinking to transform inputs into outputs. Additionally, these authors offer insights into teaching methods and strategies for assessing student learning in technology education classrooms.

Five key themes are explored through this book: (1) the historical and contemporary integration of computers into education, (2) the role of technology education in preparing students with digital competencies, (3) teacher experiences embedding P&CT into the curriculum, (4) strategies for teaching P&CT, and (5) approaches to assessment. The following paragraphs review the chapters of this book through the lens of these five themes.

Thematic Lens

Theme 1: Historical and Contemporary Integration of Computers into Education

The opening chapter, written by Hallström, provides an overview of the book and explains the importance of computational thinking. This chapter also discusses how technology education serves as a platform for cultivating computational thinking. Chapters 2 through 4 delve into the historical and contemporary integration of computers into education including the perspectives from other countries.

In chapter two Nissen and Stenliden examine the *push and pull* factors driving the need for digital competence from both demand and supply perspectives. They analyze the progression of these factors across different eras. In the early 1980s, computers were primarily viewed as tools for learning, such as performing calculations. By the late 1990s, with the advent of the internet, computers became a key source of information. This period is comparable to the transformative impact of the printing press during the Age of Enlightenment (Castells, 2010; Man, 2009). In the present day, the emphasis has shifted to digital competencies, which extend beyond basic computer literacy to include

using computers for creating technology. This is reflected in the Swedish curriculum, which highlights the importance of solving problems and translating ideas into action through creative programming and technological solutions (programmed technological solutions, or PTS). Chapter two underscores the critical need to prepare students with digital competencies and sets the stage for deeper discussions about the knowledge and skills required to achieve this goal.

Rolandsson, Kilhamn, and Bråting, address in Chapter 3 the process of knowledge transposition wherein expert knowledge is adapted by early adopters for educational purposes. The authors analyze debates between experts and early adopters regarding what, how, and why computational thinking and programming should be taught, drawing on the concept of praxeology. Their analysis reveals that, while both groups focus on similar tasks, their approaches differ. Experts emphasize authentic and structural thinking due to their deep understanding of computer science and programming languages, whereas early adopters prioritize the practical transition from algorithms to code and favor easy-to-understand coding practices over highly efficient ones. These differences reflect contrasting perspectives on teaching computer programming between experts and early adopters.

Expanding on the discussions in Chapter 2, Mannila and Heintz provide in Chapter 4 an exploration of how programming and computational thinking have been introduced not only in Sweden but also in other countries, including Finland, Estonia, the United Kingdom, Norway, and South Korea. Their chapter addresses key questions such as who should learn programming, what content should be taught, and how it should be taught. For instance, Sweden has integrated programming and computational thinking across the curriculum, particularly in mathematics and technology for all students to be equipped with digital competency. Other countries, however, may adopt different approaches, each with its own advantages and challenges.

Theme 2: Role of Technology Education in Preparing Students with Digital Competencies

Another point that this book discusses is the critical role of technology education in equipping students with digital competency. One notable aspect discussed is the dual nature of technology, which Hallström explores in Chapter 5 within the context of integrating coding into design and making. *Design and make* is a signature pedagogy of technology education, and incorporating programming into this process enables the creation of products that can run automatically. This integration shifts students' focus beyond coding, emphasizing practical applications of programming to solve real-world problems, thereby making learning more meaningful.

In Chapter 6 Persson and Pears underscore the need to prepare future generations for navigating the digital transformation of society through curriculum reform. These authors provide an historical account of technology

education in the Swedish compulsory school system, highlighting its pivotal role in positioning Sweden as a leader in technological advancement. Persson and Pears argue that equipping students with digital competency through technology education is essential for maintaining Sweden's technological excellence. The Swedish National Agency for Education (NAE) identifies computational thinking as a core skill for problem-solving and programming. When combined with technology education, computational thinking provides a framework for deconstructing and examining computational designs and systems, making intangible processes visible.

This need for technology education to equip students with digital competency is further discussed in Chapter 10, where Pérez and Svensson present an investigation of pre-service teacher experiences with Programmed Technological Solutions (PTS) in an attempt to reveal their understanding of the dual nature of technology. Through interviews, they identify four categories of understanding among pre-service teachers: (1) physical interface, (2) components as parts of a process, (3) connected, controlled, and regulated components, and (4) components as part of a system. Their findings revealed that many pre-service teachers focus primarily on the physical interface, often overlooking the broader systems and processes. Pérez and Svensson recommend that teacher education programs emphasize the connection between the dual nature of technology, systems thinking, and computational thinking. They noted that "It is important for student teachers, during their technology courses, to learn about all three dimensions and how aspects of these are connected and are understood as a whole" (p. 231).

Cederqvist, in Chapter 14, also advocates for using systems thinking to help students connect PTS to the dual nature of technology. This approach allows learners to view technological solutions as a combination of inputs, processes (the *black box*), and outputs. In this model, the black box represents computational processes guided by coding and computational thinking. Cederqvist emphasizes the importance of teaching students to understand the logic behind the code and its role in controlling technological systems, as well as the interaction between components within these systems. Systems thinking facilitates students' comprehension of how programmed technological solutions operate and how inputs are transformed into outputs.

Theme 3: Teacher Experiences Embedding P&CT into the Curriculum

The book presents unique teacher experiences with the new curriculum, with emphasis placed on computational thinking and programming. In chapter eight, Vinnervik explores how technology teachers incorporate programming into their teaching and the pedagogical challenges they face. Teachers highlighted the importance of stepwise problem-solving as a reflection of computational thinking. Tools such as LEGO Mindstorms, Micro:bit, and block-based programming were commonly mentioned as supports for teaching

programming. However, only a few teachers emphasized the importance of designing technological solutions to address authentic problems. Teachers also reported intrinsic and extrinsic challenges, including a lack of professional training, as expressed by one teacher: "When it comes to programming, we cannot be the best in the room, just forget about it. We have too little training for that" (p. 183).

In chapter nine, Engström and Björkholm focus on the content and values of programming as they emerge from professional development courses. The authors outline the progression of programming content, starting with sequencing abilities, advancing to reasoning abilities, and culminating in an understanding of systems. Key concepts include decomposition, abstraction, logical thinking, algorithm creation, and debugging. They emphasize that teaching programming should focus on logic and stepwise problem-solving. Students must have opportunities to practice, test, and troubleshoot, which allows them to separate the logic of programming from the code itself.

Chapter twelve, written by Sparf, examines teachers' experiences visiting Science Centers (SCs). Through interviews with five teachers, the chapter highlights the potential for SCs to support programming education. While teachers appreciated the activities at SCs, they noted gaps in their own knowledge of teaching programming. Teachers valued the practical activities by SC educators but emphasized the need for activities that align more closely with school curricula and learning objectives. The study concludes that collaboration between SC educators and school teachers is essential to ensure that SC activities support curriculum goals effectively.

Theme 4: Strategies for Teaching Programming and Computational Thinking

The fourth theme is an investigation into the pedagogical strategies educators use to teach programming. In chapter seven, Stolpe and Hallström conducted a literature review on visual programming and student learning in STEM disciplines. Visual programming, which includes tools such as block coding, Scratch, and Lego Mindstorms, is utilized to teach programming. The findings of the review suggest that visual programming can support students' learning of systems thinking, mathematics, technology, and programming. In chapter eleven, written by Brink, the focus shifts to exploring teachers' understanding of computer programming as a form of modeling. The study reveals that teachers perceive the concept of digital models as vague, but they believe that problem-solving can be represented through both programming and modeling. Moreover, the use of modeling supports student learning by providing an external representation that simplifies reality. Models can take various forms, such as material, verbal, or visual representations, which allow students to observe intentions and functions. The use of modeling enables students to examine, predict, and optimize outcomes.

Berg and Axell in chapter thirteen focuses on a classroom observation in which teachers teach students about Theory of Mind (ToM) and Theory of Artificial Mind (ToAM). ToM refers to the ability to understand and imagine the thoughts of others, while ToAM indicates that a robot lacks independent will and operates only based on programming. The teachers employed activities and questions to distinguish between ToM and ToAM. For example, the teacher turned off the lights and opened the windows in the classroom. When the students entered, she remarked that it was cold and dark. The students responded by turning on the light and closing the windows. The teacher then used the opportunity to discuss how humans can understand the implied meaning behind others' actions, unlike robots, which require explicit programming. The author recommends using engaging activities that encourage students to think about ToM and ToAM, alongside asking contrasting, concrete questions such as, "Do people need to be programmed?"

In chapter fifteen, Larsson examines the role of figurative language in teaching programming. Figurative language, such as metaphors, can help simplify the abstract concepts of programming. There are two levels of understanding the role of metaphors: the discursive level (knowing how to use a metaphor) and the cognitive level (understanding why a particular metaphor is used). However, the use of metaphors can sometimes lead to misconceptions.

Theme 5: Approaches to Assessment

The final theme of the book speaks to assessment. In chapter sixteen. Björklund and Nordlöf investigate the approaches used by teachers in the classroom for assessing student learning resulting from their engagement in programming. The study identifies two groups of teachers: one group emphasizes the product, valuing statements such as "The program fulfils its purpose" (p. 331) or "The program is complete" (p. 331). In contrast, the second group focuses on the process, highlighting values such as "The student demonstrates endurance" (p. 332) or "The student analyzes and makes a plan" (p. 333). Despite these differences, both groups share a common value in the importance of students being able to explain their work. As one teacher noted, "If the student can't explain the code, it is clear proof that he doesn't understand the program. If you can't explain the code, you don't know what you have done" (p. 335). The authors conclude their chapter by recommending that assessment of the process is more important, as it evaluates what students learn rather than simply their final product.

Collectively, the chapters comprising this book offer valuable insights into integrating computational thinking into the curriculum. It covers topics ranging from the need for integration and practical examples of implementation, to the perspectives held by classroom teachers on the integration of computational thinking, and pedagogical recommendations regarding teaching and assessing computational thinking. In its entirety, this book is without question a must-read for anyone wanting to better understand computational thinking and the role of

technology education in connecting it with finding solutions for authentic global problems.

Pengsuwan, P. (2025). Book Review: *Programming and Computational Thinking in Technology Education: Swedish and International Perspectives*, *36*(2), 138-145. https://doi.org/10.21061/jte.v36i2.a.7

References

Castells, M. (2010). *The Rise of the Network Society: Vol. The Information Age: Economy, Society, and Culture* (2nd ed.). Wiley-Blackwell.

Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. The MIT Press. https://doi.org/10.7551/mitpress/11740.001.0001

International Technology and Engineering Educators Association. (2020). *Standards for technological and engineering literacy: The role of technology and engineering in STEM education.* https://www.iteea.org/STEL.aspx

Man, J. (2009). *The Gutenberg Revolution:The story of a genius and an invention that changed the world*. Transworld Publishers.

Wing, J. M. (2006). Computational thinking. *Commun. ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215