

Chapter 5

Computational Thinking Processes and Their Congruence with Problem-Solving and Information Processing



A. Labusch, B. Eickelmann and M. Vennemann

Abstract Computational thinking is emerging as twenty-first century's key competence, especially for today's students as the new generation of digital learners. The acquisition of this competence requires in-depth knowledge in learning and computational thinking processes (Labusch & Eickelmann, 2017), both of which feature in various frameworks (e.g., Kong et al., 2017). Problem-solving theories can also be taken into account to obtain a more precise analysis of these processes since previous research has suggested a large degree of congruence between problem-solving and computational thinking (e.g., Yadav, Stephenson, & Hong, 2017b; Wing, 2008). In this contribution, to analyze computational thinking processes, computational thinking will be first considered in a general problem-solving context and then in an information processing context. Research of this kind is of immense importance to pave the way for further computational thinking in schools. In this regard, the approach provides the theoretical grounding for a national extension (Labusch & Eickelmann, 2017) to the *International Computer and Information Literacy Study 2018* (ICILS 2018) (Eickelmann, 2017; Fraillon, Schulz, Friedman, & Duckworth, 2019) which enables for the first time the cross-national measurement of student achievement in computational thinking. This contribution, therefore, serves as a theoretical basis for understanding computational thinking processes regarding problem-solving and cognitive information processing.

Keywords Computational thinking · Problem-solving · Information processing · ICILS 2018

A. Labusch (✉) · B. Eickelmann · M. Vennemann
Institute for Educational Science, Paderborn University, Paderborn, Germany
e-mail: amelie.labusch@upb.de

B. Eickelmann
e-mail: birgit.eickelmann@upb.de

M. Vennemann
e-mail: mario.vennemann@upb.de

5.1 Introduction

Recent studies consider computational thinking to be an important competence of the twenty-first century (e.g., Voogt et al., 2015) that “everyone, not just computer scientists, would be eager to learn” (Wing, 2006, p. 33). Given that computational thinking is an important skill that every student needs to a certain extent for his or her future professional and personal life, the very first question is whether he or she really possesses this skill. In order to clarify this question, it is necessary to carry out an appropriate study. This, in turn, requires a framework that includes the definition and conceptualization of computational thinking. Common sense would suggest that computational thinking can be described as a way of thinking and of solving problems in a modern and technology-filled world (Curzon & McOwan, 2017). However, agreement usually ends when it comes to the question of what constitutes the core elements of computational thinking. The knowledge of the core elements and the associated skills that a student should possess is indispensable in order to be able to measure them at all and for a definition and conceptualization of computational thinking. While researchers have considered several definitions in their endeavors to specify computational thinking, no generally accepted definition has yet been established (Mannila et al., 2014). This also makes assessing it a challenge (Brennan & Resnick, 2012). Voogt et al. (2015) argue that trying to define computational thinking by setting conditions that it has to fulfill makes it “difficult to implement in practice” (p. 718). The role of education and especially of schools (e.g., as a set of transferable, cross-curricular skills) is also a topic for discussion in this context (Fraillon et al., 2019). It raises new tasks and challenges for schools. However, some experts have already developed frameworks (e.g., Brennan & Resnick, 2012; Kong et al., 2017) “to nurture students’ ability to see the world through a computational lens” (Kong et al., 2017, p. 84). These frameworks include discipline-specific aspects of programming tasks that can serve as a basis for curricula. While it is true that computational thinking does have a close affinity to computer science (Curzon & McOwan, 2017), it does not necessarily follow that it is solely confined to this area. Indeed, Voogt et al. (2015) argue that the impression that computational thinking can be equated to programming that comes from the fact that many studies or discussions consider programming as its context.

Grover and Pea (2013) emphasize the need to consider whether all students should learn computational thinking, “including those who allege no interest in pursuing CS [computer science] and STEM [science, technology, engineering and mathematics] careers” (p. 40). However, their assertion recognizes the deeper problem, namely that the default way of thinking about computational thinking assigns it solely to computer science or STEM subjects. In fact, everyone should be given the opportunity to gain competences in the field of computational thinking (Wing, 2017) to allow them to successfully participate in a digitalized world. This basically confirms the argument put forward by Wing (2006), who adds weight to the argument that computational thinking is of central importance for all sciences, not only for computer science (Yadav et al., 2017a).

However, this poses major challenges for schools and teachers, because then noncomputer science teachers must also have skills that allow them to teach computational thinking adequately. However, it turns out that schools are often already “struggling with the way in which to use information technology in the classroom” (Faber, Wierdsma, Doornbos, & van der Ven, 2017, p. 13). Faber et al. (2017) suggest “unplugged” lessons in computational thinking. When discussing the question of what unplugged activities of computational thinking might look like, based on existing definitions, problem-solving, and cognitive abilities are important aspects. Taking a large congruence for granted gives researchers the added advantage of being able to resort to problem-solving and cognitive abilities theories. Problem-solving, especially complex problem-solving, has been explored for decades (Fischer, Greiff, & Funke, 2017), thus providing a sound body of well-proven research.

Assuming that the study will show that not all students possess this competence equally and to a sufficient degree, the question also arises as to which influencing factors can explain differences in students’ achievement in computational thinking. Personal and social factors can be taken into account, but when it comes to learning and teaching computational thinking, the question of the role of schools and especially teachers arises. Therefore, the school context should not be disregarded as a possible influence variable on students’ achievement of computational thinking. It has been a controversial issue in the recent discourse on the implementation of computational thinking into curricula of how to learn and teach it (Grover, 2017). Some experts argue that knowledge of what constitutes computational thinking and how to make full use of it makes it accessible (Barr, Harrison, & Conery, 2011). From this perspective, it is assumed that understanding computational thinking and its core elements “would enable teachers to integrate it in their teaching concepts” (Labusch & Eickelmann, 2017, p. 103).

5.2 Current State of Research

In summary, core elements of computational thinking should be identified in order to obtain a definition and conceptualization of computational thinking in the course of a framework for a study that measures whether students have the necessary competences. This is done in the following sections (5.2 and following) when considering the core elements, which are adopted within the framework and the study based on it. The study has been described in detail in the previous chapter (see Chap. 4 by Eickelmann in this book) and is mentioned again in this chapter in 5.3.1 and extended by an explanation of a national extension. This is followed by considering computational thinking in a general problem-solving context (5.2.1) and then in an information processing context (5.2.2) to show how these, as “unplugged aspects” of computational thinking, are related to its processes. Subsequently, computational thinking processes can be further analyzed (5.2.3). Finally, it is considered which

aspects of the school context are to be taken into account in the context of the in school acquisition of competences in the field of computational thinking (5.2.4) to be able to explain variation in students' achievement in computational thinking.

5.2.1 Computational Thinking and Problem-Solving

In recent years, computational thinking has often been mentioned in the problem-solving context (e.g., Korkmaz, Çakir, & Özden, 2017; Román-González, Pérez-González, & Jiménez-Fernández, 2017). Problem-solving is described as a transformation from an undesirable initial state to a desirable final state (Beecher, 2017) by overcoming a barrier. Doing so means operating at high levels of thinking and reasoning (Spector & Park, 2012). Binkley et al. (2012) turn their attention to the challenge of teaching additional competences such as sophisticated thinking and flexible problem-solving to put students in the best possible starting position for participation in work, life, and society.

While the acquisition of problem-solving skills is a current topic of discussion, studies into problem-solving skills and problem-solving processes have been being conducted for decades. A problem-solving process is frequently described as a seven-stage cycle (Pretz, Naples, & Sternberg, 2003): (1) the recognition or identification of a problem, (2) the definition and mental representation of the problem, (3) the development of a strategy to solve the problem, (4) the organization of knowledge concerning the problem, (5) the allocation of mental and physical resources to solving the problem, (6) the monitoring of progress toward the goal, and (7) the evaluation of the solution for accuracy. Although there might be several ways to solve a problem—and the one chosen will depend on the nature of the actual problem in hand—it is assumed that they will all have the same processes in common. The first stage of the problem-solving process is therefore of huge importance: The problem needs to be both recognized and identified in order to select the best possible solution since how a problem is solved depends on the actual problem itself. Since these processes are very similar to computational thinking processes, it is important to investigate their similarities and differences by exploring the congruence between general problem-solving and computational thinking.

5.2.2 Computational Thinking and Information Processing

Many researchers and educators contend that computational thinking changes our way of thinking (Bundy, 2007). Information processing theories and models offer deeper insights into this subject: ACT-R (Adaptive Control of Thought—Revised) (Anderson, 2010) and MEKIV (Model for elementary and complex human information processing) (Hussy, 1998), for instance, are two global models that incorporate a useful cognitive architecture for modeling complex problem-solving processes.

They both have in common that they try to integrate multiple mental functions and thus take a more realistic approach to the complexity of action control and thinking. While ACT-R has a symbolic structure and can be used to create cognitive models that simulate human cognitive processes (Hussy, 1998), MEKIV provides a more theoretical description of elementary and complex human information processing, starting with the sensory organs and ending in the motoric system. All information stored in the sensory register is passed on to the long-term memory, where the incoming pieces of information are compared with the stored patterns that have meaning. Stimuli received via sensory organs are thus encoded and interpreted by translating them into cognitive representations, which are then stored in memory. In doing so, the stimulus is correlated with existing knowledge, whereby it is enriched with information. Prerequisites for this are the accessibility and retrievability of stored knowledge. The process of categorizing the stimulus is of particular importance when it comes to encoding. The encoded perception is itself stored in memory and creates new memory contents in interaction with the knowledge already stored there. Human behavior then represents the output of this information processing (Bless, Fiedler, & Strack, 2004).

5.2.3 Computational Thinking Processes

When computational thinking processes are compared with problem-solving processes and information processing, a closer inspection reveals a high degree of similarity. Accordingly, this contribution focuses on the following definition: “In computational thinking, the students demonstrate the ability to identify a problem, break it down into manageable steps, work out the important details or patterns, shape possible solutions and present these solutions in a way that a computer, a human, or both, can understand. Computational thinking can also involve structuring and manipulating data sets to support the solution process” (IEA, 2016, p. 1). This definition includes problem identification and formulation, a breakdown process (decomposition), the process of pattern recognition and pattern matching, the shaping of a possible solution with several processes such as abstraction, evaluation, and debugging, as well as algorithmic thinking by presenting solutions. It tells a lot about the cognitive structure of computational thinking, containing a structured, conceptual, and strategic way of thinking and solving problems. The computational thinking processes described in the following are all related to the aforementioned definition.

5.2.3.1 Problem Identification and Problem Definition

The chosen definition is based on a computational thinking assessment framework that contains the two strands of conceptualizing problems and operationalizing solutions (Fraillon et al., 2019). The problem-solving process cycle outlined above

includes both the identification and the definition of problems. That means that problems have to be conceptualized in understanding and framed before solutions can be formed (ibid.). Identifying the problem also permits a decision on whether it can be solved using computational thinking. Understanding the problem to be solved is an important prerequisite for the problem-solving process. Pólya (2004) points out that “it is foolish to answer a question that you do not understand” (p. 6). While a scientific investigation process begins with a hypothesis (Riley & Hunt, 2014), a problem-solving process begins with a problem definition. Michaelson (2015) maintains that characterizing the problem is the most challenging part of problem-solving. When a person presents a problem to someone else and asks them to solve it (e.g., in school, when a teacher asks her/his students to solve a problem), it is best if they formulate the problem in their own words (Pólya, 2004). While it is not the rule that computing activities begin with problem identification and definition or are always thought in terms of problems to be solved, according to the above definition, problem identification and problem definition may be included in computational thinking processes.

5.2.3.2 Decomposition and Data Collection, Analysis, and Representation

Miller (1956) has found that the human memory is limited to 7 ± 2 items, which means that some problems are too complex for the human brain to solve unless they are first decomposed into subproblems to be processed in the brain. This process of breaking a problem “down into manageable steps” (IEA, 2016, p. 1) is called decomposition. It is a core element of a computational thinking process and ensures that even complex problems can be understood. While some computational thinking processes belong to only one strand (conceptualizing problems or operationalizing solutions), others—like pattern recognition and pattern matching—are relevant to almost all aspects of a computational thinking process. Decomposing a problem by analyzing it as a whole also requires knowledge of what constitutes manageable steps and how they are related to each other and the whole problem. The decomposition process is well known in language arts, where it is referred to as outlining (Barr & Stephenson, 2011), i.e., organizing work by decomposing it into its main ideas (Riley & Hunt, 2014). Data can support the computational thinking process by providing important information. This means finding a data source, analyzing the data, and using data structures to represent the data (Barr & Stephenson, 2011).

5.2.3.3 Modeling and Revising Solutions

The problem fragments of the decomposition process are shaped into possible solutions by modeling and revising them. A model can serve as an abstract representation of a real-world situation (Frigg, 2002), which is why a modeled solution might also be applied to real-world problems. Modeling a solution involves different processes and

is also part of a problem-solving process. Pattern recognition and pattern matching processes also form part of information processing. They mean finding similarities and differences and accessing something known. Existing knowledge is used to categorize the problem fragments. A pattern indicates whether or not a particular string is part of a category or class defined by the pattern (Riley & Hunt, 2014). Pattern recognition methods also find application in other domains, e.g., medicine, where they are used to recognize patterns in DNA strands (Fink, 2014). In problem-solving processes, pattern matching saves work, since findings can be transferred to a similar problem so that this does not have to be tackled from the very beginning (Curzon & McOwan, 2017).

The categorized problem fragments are then generalized by eliminating unimportant details (Barr & Stephenson, 2011) in an abstraction process based on inductive logic. Inductive logic is based on the principle that general conclusions can be drawn from observations and experiences. Since these conclusions contain uncertainty, the degree of certainty depends on the number of observations/experiences. Inductive reasoning is the process of concluding individual cases into general rules. If fixed rules are consulted and formulated as premises from which conclusions are deduced, deductive reasoning takes place. This requires sufficient knowledge about these fixed rules and a need to organize knowledge regarding the problem. However, the main part of an abstraction process is the elimination of unimportant details and thus the focus on the essential. This allows a particular emphasis to be chosen and the simplification of complex structures. In social studies, abstraction can be a useful means of summarizing facts and using them to draw conclusions (Barr & Stephenson, 2011).

Even if the chosen definition does not actually reflect this in detail, a computational thinking process, like a problem-solving process, contains an evaluation component (Fraillon et al., 2019). During the process, but above all after it has been shaped, the solution has to be tested to make sure that the previously formulated goal can be achieved. Debugging is important here in order to test the shaped solution. Murphy et al. (2008) identify various different debugging strategies, including testing and pattern matching. They recognize pattern recognition and pattern matching processes through the students' intuitive observation and action. Riley and Hunt (2014), in turn, find similarities between a debugging process and the proof of a mathematical theorem. In the education context, debugging also provides a deeper understanding of the problem-solving behavior of students (Liu, Zhi, Hicks, & Barnes, 2017).

5.2.3.4 Algorithmic Thinking

Denning (2009) suggests that the basic idea behind computational thinking is in essence algorithmic thinking. In the field of computer science, an algorithm is defined as "any well-defined sequence of actions that takes a set of values as input and produces some set of values as output" (Riley & Hunt, 2014, p. 130). An algorithmic view of life is assumed to be very valuable because it involves many essential activities in life by following simple and discrete steps. The thinking process required to formulate an algorithm differs in one aspect from the formulation of a well-defined rule of

action: another form of language is needed. An algorithm run by a computer requires a programming language that ensures that an algorithm can only have one possible interpretation so that there is no misunderstanding (ibid.). While human language does not fulfill this requirement, the formulation of an action rule can be deemed algorithmic thinking when only the process itself is taken into consideration. The formulated solution is usually to be evaluated for correctness, efficiency, elegance, and usability (Beecher, 2017, p. 100). The advantage of algorithmic thinking is that the solution can be delegated to another person or a computer. Barr and Stephenson (2011) advocate the inclusion of algorithmic thinking in other subjects and not just in computer science. A good example here would be writing an instruction. Since it cannot be presupposed that students are familiar with a programming language, writing an instruction helps them to practice the logical reasoning used in writing an algorithm.

The above sections emphasize three things. First, computational thinking processes apply to processes in other domains. Second, there is a strong similarity between problem-solving processes and computational thinking processes, whereby it should be noted that the process of algorithmic thinking represents a clear extension of the pure solution formulation. Third, computational thinking processes can also be practiced and carried out in other subjects in school.

5.2.4 In-School Acquisition of Competences in the Field of Computational Thinking

The above-mentioned computational thinking processes describe the cognitive part of computational thinking. By demonstrating that five factors of personality correlate with computational thinking, Román-González et al. (2016) extend their finding that problem-solving is close to computational thinking. As a result, noncognitive factors that influence computational thinking at student level also need to be considered and investigated. Ramalingam, Labelle, and Wiedenbeck (2004) contend that programming experience influences programming self-efficacy. Since there is a kinship between programming and computational thinking, these results add weight to the assumption that the students' self-efficacy in computational thinking may have an impact on their acquisition of competences in the field of computational thinking. Another student-level variable that might have an impact on student achievement in computational thinking can include background characteristics (e.g., gender). Gender differences are evident in the programming field (Sullivan & Bers, 2016), and it can be assumed that differences are also found in the field of computational thinking.

Sanford and Naidu (2016) emphasize that computational thinking is not self-evident and calls for training and guidance. Indeed, many experts maintain that computational thinking should be integrated into teaching and learning. Focusing on students' individual needs, requirements, and learning environments is important because studying students' achievement in computational thinking sets them in the

context of different effects on the development of computational thinking. These effects can be classified as factual, attitudinal, and behavioral and take place on different levels (Fraillon, Ainley, Schulz, Friedman, & Gebhardt, 2014).

5.3 Research Concept

An analysis of the above reveals the emergent need for a study which measures the in-school acquisition of computational thinking as a cross-curricular competence, using a research concept that integrates general problem-solving and school-related parameters. The findings indicate that while a congruence between computational thinking and problem-solving is often taken for granted, few scholars have actually investigated this assumption. There is also an absence of a study with a substantial database that explores the congruence between computational thinking and problem-solving as well as individual and school-related factors relating to the acquisition of competences in the field of computational thinking with the aim of creating a holistic picture of computational thinking in schools.

As a result, the following research questions are addressed within this contribution:

1. To what extent are students' achievements in computational thinking related to their general problem-solving skills?
2. To what extent are students' achievements in computational thinking related to their general problem-solving skills under control of individual characteristics?
3. To what extent are students' achievements in computational thinking related to their general problem-solving skills under control of the school context?

It is assumed that, as has already been shown theoretically, there is a strong positive relationship between students' achievement in computational thinking and their problem-solving skills (research question 1). However, it is also assumed that there are other variables at the student and school level influencing the relationship (research question 2 and 3). These assumptions have to be verified or rejected in a suitable study.

5.3.1 Study and Data Basis

The methods and instruments to be used for the analyzes will be aligned with the aforementioned research questions. The data basis is provided by the international comparative study IEA-ICILS 2018 (*International Computer and Information Literacy Study*), in which the authors are participating as members of the national study center in Germany. In ICILS 2013, students' computer and information literacy (CIL) was measured on a representative basis in an international comparison. In the course of the second cycle of ICILS in 2018, the IEA (*International Association for the Evaluation of Educational Achievement*) is for the first time implementing a new,

additional option to assess “computational thinking” by applying computer-based student tests and questionnaires to determine in addition to other variables individual student background characteristics and school-related parameters (Fraillon et al., 2019). Some 20 countries are participating in ICILS 2018; those making use of the additional “computational thinking” option include the USA, France, South Korea, Germany, Denmark, Portugal, and Luxembourg.

The data will be gathered from representative samples, which will be designed—as in 2013—as two-stage cluster samples, first sampling schools and then selecting about 20 students, 15 teachers, 1 principal, and 1 ICT coordinator in each school. The German subsample in 2018 will comprise around 4,500 students. The ICILS 2018 student sample is drawn from Grade 8 students; the teacher population is defined as all teachers teaching Grade 8 at the sampled schools. Both student and teacher data will be used to answer the research questions. While the former will be used to identify students’ achievement in different fields (e.g., computational thinking) and background data, the latter will serve to identify the school-related parameter “teachers’ attitude to teaching computational thinking.” After the main survey and data submission in 2018, the first results of the study will be presented in the national report in 2019.

To explore the congruence between students’ achievement in computational thinking and their problem-solving skills (research question 1), both areas need to be tested. This is also the approach taken in the German national extension to ICILS 2018 (Labusch & Eickelmann, 2017), which uses computer-based tests to measure the achievement in computational thinking of Grade 8 students and in addition applies paper-based tests to assess their cognitive, problem-solving, pattern recognition, pattern matching, and debugging skills. The problem-solving processes worked out are used as a theoretical basis for the conceptual design of the problem-solving scale. A four-point Likert scale, ranging from “strongly agree” to “strongly disagree” is used to assess how well students solve problems and contains three subscales: problem-solving confidence (to assess self-perceived confidence in solving problems), approach-avoidance style (to assess whether the students tend to approach or avoid problems), and personal control (to assess elements of self-control). Since the German national extension to this study also collects data about students’ computational thinking self-concepts, these can also be used to answer the underpinning research questions.

5.3.2 Methodology and Expected Outcomes

Since contextual factors influence variations in human cognition, affect and behavior, it will be needed to use, in addition to descriptive primary and secondary analyses, a multilevel statistical model to avoid problems of parameter misestimation and model mis-specification (Rowe, 2009). It is now widely accepted that studies that assess the impact of school education on students’ learning outcomes must take into account the fact that students’ progress is influenced by complex, multilevel,

multidimensional, and interrelated factors (ibid.). To examine the extent to which different factors at the student and school level are associated with variations in the relationship between students' achievement in computational thinking and their problem-solving skills, a multilevel structural equation model will be used. Factors of interest at student level include those related to students' general problem-solving skills, their basic cognitive abilities, their self-reported proficiency in computational thinking as well as their personal and home background. Aggregated at school level, the school-related use of computational thinking and teachers' attitudes to teaching computational thinking are two key factors of interest. The theoretical model has been developed using corresponding research literature to determine which influencing factors on students' achievement in computational thinking are to be taken into account for multivariate analyses. The types of results can be divided into three categories: First, those that illuminate the computational thinking construct and contribute to clarifying the congruence with general problem-solving; second, due to the extensive data basis that will be generated in ICILS 2018, additional student variables such as basic cognitive skills, self-concept, and background characteristics will also be examined; third, the school context can be included.

5.4 Summary and Outlook

For the first time, the congruence between computational thinking and general problem-solving will be examined under the control of other variables at student and school level on the basis of a representative sample in the context of a national extension of the IEA-study ICILS 2018. The data has been collected in 2018, and initial results will be available in 2019. Using a cognitive approach, these results might provide a starting point for implementing computational thinking into curricula as a multidisciplinary and cross-curricular key competence of the twenty-first century and contribute to moving school systems into the digital age.

References

- Anderson, J. R. (2010). *How can the human mind occur in the physical universe?* New York: Oxford University Press.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Learning with Technology*, 38(6), 20–23.
- 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.
- Beecher, K. (2017). *Computational Thinking: A beginner's guide to problem-solving and programming*. Swindon, UK: BCS Learning & Development Limited.
- Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining twenty-first century skills. In P. Griffin, B. McGaw & E. Care (Eds.), *Assessment and teaching of 21st century skills* (pp. 17–66). Dordrecht: Springer.

- Bless, H., Fiedler, K., & Strack, F. (2004). *Social cognition. How individuals construct social reality [Social psychology: A modular course]*. Philadelphia, PA: Psychology Press.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Paper Presented at the Annual Meeting of the American Educational Research Association*, Vancouver, Canada.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.
- Curzon, P., & McOwan, P. W. (2017). *The power of computational thinking: Games, magic and puzzles to help you become a computational thinker*. London, UK: World Scientific Publishing Europe Ltd.
- Denning, P. J. (2009). The profession of IT: Beyond computational thinking. *Communications of the ACM*, 52(6), 28–30.
- Eickelmann, B. (2017). Computational Thinking als internationales Zusatzmodul zu ICILS 2018 – Konzeptionierung und Perspektiven für die empirische Bildungsforschung [Computation Thinking in the context of ICILS 2018 – the perspective of educational research]. *Tertium Comparationis. Journal für International und Interkulturell Vergleichende Erziehungswissenschaft*, 23(1), 47–61.
- Faber, H. H., Wiersma, M. D. M., Doombos, R. P., & van der Ven, J. S. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, 12, 13–24.
- Fink, G. A. (2014). *Markov models for pattern recognition. from theory to applications*. London: Springer.
- Fischer, A., Greiff, S., & Funke, J. (2017). The history of complex problem solving. In B. Csapó & J. Funke (Eds.), *The nature of problem solving: Using research to inspire 21st century learning* (pp. 107–121). Paris: OECD Publishing.
- Frailon, J., Ainley, J., Schulz, W., Friedman, T., & Gebhardt, E. (2014). Preparing for life in a digital age. The IEA international computer and information literacy study. In *International Report*. Amsterdam: International Association for the Evaluation of Educational Achievement (IEA).
- Frailon, J., Schulz, W., Friedman, T., & Duckworth, D. (2019). *Assessment framework of ICILS 2018*. Amsterdam: IEA.
- Frigg, R. (2002). *Models and representation: Why structures are not enough. Measurement in physics and economics project discussion paper series*. London: London School of Economics.
- Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. In P. Rich & C. B. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 269–288): Springer Publishing Company.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Hussy, W. (1998). *Denken und Problemlösen [Thinking and problem-solving]* (2nd ed.). Stuttgart: Kohlhammer.
- IEA (2016). *The IEA's international computer and information literacy study (ICILS) 2018*. What's next for IEA's ICILS in 2018? Retrieved December 12, 2017, from http://www.iea.nl/fileadmin/user_upload/Studies/ICILS_2018/IEA_ICILS_2018_Computational_Thinking_Leaflet.pdf.
- Kong, S. C., Abelson, H., Sheldon, J., Lao, A., Tissenbaum, M., Lai, M., Lang, K., & Lao, N. (2017). Curriculum activities to foster primary school students' computational practices in block-based programming environments. In S. C. Kong, J. Sheldon & K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education 2017* (pp. 84–89). Hong Kong: The Education University of Hong Kong.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558–569.
- Labusch, A., & Eickelmann, B. (2017). Computational thinking as a key competence—A research concept. In S. C. Kong, J. Sheldon & K. Y. Li (Eds.), *Conference Proceedings of International*

- Conference on Computational Thinking Education 2017* (pp. 103–106). Hong Kong: The Education University of Hong Kong.
- Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, 27(1), 1–29.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 1–29).
- Michaelson, G. (2015). Teaching Programming with computational and informational thinking. *Journal of Pedagogic Development*, 5(1), 51–66.
- Milller, G. A. (1956). The magical number seven plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2), 81–97.
- Murphy, L., Lewandowski, G., McCauley, R., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: The good, the bad, and the quirky—A qualitative analysis of novices’ strategies. *SIGCSE Bulletin*, 40(1), 163–167.
- Pólya, G. (2004). *How to solve it: A new aspect of mathematical method*. Princeton: Princeton University Press.
- Pretz, J. E., Naples, A. J., & Sternberg, R. J. (2003). Recognizing, defining and representing problems. In J. E. Davidson & R. J. Sternberg (Eds.), *The psychology of problem solving* (pp. 3–30). Cambridge, UK: Cambridge University Press.
- Ramalingam, V., Labelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *SIGCSE Bulletin Inroads*, 36(3), 171–175.
- Riley, D. D., & Hunt, K. A. (2014). *Computational thinking for the modern problem solver*. Boca Raton, FL: CRC Press.
- Román-González, M., Pérez-González, J.-C., & Jiménez-Fernandez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691.
- Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2016). Does computational thinking correlate with personality? The non-cognitive side of computational thinking. In *Paper presented at the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, Salamanca, Spain.
- Rowe, K. J. (2009). Structural equation modeling in educational research. In T. Teo & M. S. Khine (Eds.), *Structural equation modeling in educational research: concepts and applications* (pp. 201–239). Rotterdam, The Netherlands: Sense Publishers.
- Sanford, J. F., & Naidu, J. T. (2016). Computational thinking concepts for grade school. *Contemporary Issues in Education Research*, 9(1), 23–32.
- Spector, J. M., & Park, S. W. (2012). Argumentation, critical reasoning, and problem solving. In S. B. Fee & B. R. Belland (Eds.), *The role of criticism in understanding problem solving* (pp. 13–33). New York: Springer.
- Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: Gender differences in young children’s performance on robotics and programming tasks. *Journal of Information Technology Education: Innovations in Practice*, 15, 145–165.
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728.
- Wing, J. M. (2006). Computational thinking. *Communication in ACM*, 49(3), 33–35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725.
- Wing, J. M. (2017). Computational thinking’s influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14.
- Yadav, A., Gretter, S., Good, J., & McLean, T. (2017a). Computational thinking in teacher education. In P. Rich & C. B. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 205–220). Springer Publishing Company.

Yadav, A., Stephenson, C., & Hong, H. (2017b). Computational thinking for teacher education. *Communications of the ACM*, 60(4), 55–62.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

