About Computational Thinking Assessment: a Proposal for Primary School First Year from a Pedagogical Perspective

Marina Sartor Hoffer
Free University of Bozen/Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
marina.sartor@unibz.it

Sara Baroni
Free University of Bozen/Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
sara.baroni@unibz.it

Ilenia Fronza
Free University of Bozen/Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
ilenia.fronza@unibz.it

Claus Pahl
Free University of Bozen/Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
claus.pahl@unibz.it

Abstract

Computational Thinking is a key set of skills, which actually represents an obstacle to the clear definition of an effective assessment strategy. In this work, first we explain why designing an assessment framework is even more challenging for first year elementary school. Based on these premises, we propose to combine the Computational Thinking educational contribution and the problem-solving skills connected to it with the specific needs of the educational context. Taking into account age and expected educational outcomes, we propose to evaluate algorithmic thinking, problem-solving, and creativity. Finally, we discuss possible challenges related to this approach, and report a set of lessons learned that could contribute to solving these challenges.

1 Introduction

After Jeannette Wing's seminal article in 2006 [Win06], the term Computational Thinking (CT) assumed a broad meaning, and it took some years of discussion in the research community to obtain an agreed operational definition [BR12, Win14]. In 2016, the International Society for Technology in Education (ISTE) defined CT as the ability to develop creative solutions to a problem, which can be described through an algorithmic strategy [A+16].

Computational Thinking is considered as a key set of skills that should be learned by everybody [Gro17] regardless of the chosen career path. However, being a set of skills represents an obstacle to the precise definition of an *effective strategy for CT learning assessment* and evaluation, which is in turn of paramount importance to incorporate CT in a curriculum [HL15].

For some years, many of the existing assessment strategies have been based on code analysis [Cro14, Net13], which can result in misunderstanding the development of CT skills [RGMLR17] by ignoring a set of skills that cannot be measured by looking just at one's code. Therefore, it is suggested to consider multiple measures that are complementary, encourage and reflect deeper learning, and contribute to a comprehensive picture of students' learning [Gro15, BFCP18].

In general, the advantage of defining an algorithmic solution of a problem lies in the re-usability of the defined solution, which can be useful at some point both to who solved the problem and to those who may face the same problem in the future. From this perspective, finding a solution can be considered as a social advantage, and this opens the possibility to motivate people to work together towards a solution, which requires at the same time the skill of effective communication.

In: I. Fronza, C. Pahl (eds.): Proceedings of the 2nd Systems of Assessments for Computational Thinking Learning Workshop (TACKLE 2019), co-located with 14th European Conference on Technology Enhanced Learning (EC-TEL 2019), 17-09-2019, published at http://ceur-ws.org.

Based on these premises, we are working on a research project, called *COmbining COmputational thiNking didAcTics and Software engineering in K-12* (COCONATS[1]). The project aims at designing activities for K-12, which have as a principal output the acquisition of a reasoning approach that leads to clear programming, display, and implementation of the product. This not only allows pupils to address any discipline systematically and effectively but also promotes a dimension of meta-cognitive reasoning, which in turn allows them to address and connect further complexities.

In this work, we propose to combine the CT educational contribution and the problem-solving skills connected to it with the specific needs of the educational context. For the specific case of primary school first year, taking into account age and expected educational outcomes, we propose to evaluate algorithmic thinking, problem-solving, and creativity. Moreover, we discuss possible challenges related to this approach and report a set of lessons learned that could contribute to solving these challenges.

Section 2 reports the state of the art of existing CT systems of assessments. Section 3 briefly describes the objectives of the COCONATS project. Section 4 details the approach proposed in this paper, while Section 5 discusses possible problems that might emerge while developing this solution. Section 6 draws conclusions from this work, also proposing possible directions for future work.

2 State of The Art

As above mentioned, one of the main barriers to the implementation of Computational Thinking in the school context is the absence of an agreed assessment strategy. Indeed, assessment determines whether or not educational goals are being met and, at the same time, it drives the design of the curriculum itself [HL15].

In 2017 Román-González et al. [RGMLR17] provided an overview of the existing research works on this topic, and classified them based on their perspective (e.g., summative assessment, perceptions-attitudes scales, etc.). Some relevant examples are the Computational Thinking Test [Gon15, RGPGJF17], the Test for Measuring Basic Programming Abilities [MRH15], and the Commutative Assessment Test [WW15]. All the tests mentioned above have been designed for middle- and/or high-school students.

A large number of proposals and perspectives for Computational Thinking assessment reflects the extreme difficulty of measuring it, due to a large number of variables both at a personal and social level, especially in lower school levels. Indeed, Román-González and his co-authors [RGPGMLR18] asserted that CT is currently still a confused psychological construct and that its evaluation remains a thorny and unresolved issue.

In recent years several evaluation tools have been developed from different approaches and various operational tools have been proposed [FP18]; nevertheless, very little research has been conducted to study whether these tools provide convergent measures and how to combine them in an educational environment [RGPGMLR18, BFCP18, FIC17].

Moreover, recent literature for assessment is present for universities or high schools. Nevertheless, there are just a few examples designed for lower school grades. The reason is clear: until the end of the primary school, pupils do not yet have the concept of abstraction, which is a founding concept of Computational Thinking [KA19]. The skills that are being built at this age are so many that it is very problematic to understand whether a positive result in the field of CT is to refer to the activities prepared for that purpose or if a series of external factors affect the assessment, such as the richness of the vocabulary, the familiar context, the extra-school opportunities to use technologies, and so on.

To this consideration, it should be added the peculiarity of the first class of primary school (age: 6) when compared to the other classes of primary school. This class represents, in fact, a delicate phase (sometimes with many obstacles) in which children need to develop many skills and knowledge at the same time, such as learning to write, read, count, and be with others. This learning phase requires great commitment and psycho-physical energy. Moreover, the learning process does not proceed at the same pace for each individual; therefore, any assessment should take into account the differences in individual learning styles, which represent an obstacle to obtaining comparable levels in any result. It is therefore tough to think of an evaluation of the *process* because there are too many individual variables that affect the process in this age group.

One alternative possibility would then be to evaluate the *outcome* of the performed activities. However, Israel et al.[ICD+95] argued that in the outcome evaluation a very large sample of subjects would be required to obtain statistically significant results and, above all, the objectives that guide an outcome search often mature too long before being evaluated.

Moreover, when the outcome consists of a piece of software, as above mentioned evaluating the outcome by focusing only on the analysis of the code is not the optimal solution when the goal is evaluating the

[1] coconats.inf.unibz.it

impact that CT has on ordinary activities, being CT by definition a set of competences [BFCP18], also described by Corradini et al. [CLN17] and classifiable in four general categories:

- mental processes,
- methods,
- practices,
- transversal objectives (such as creative, communicative and collaborative skills).

3 The COCONATS project

The COCONATS project involves both the Faculty of Computer Science and the Faculty of Education of the Free University of Bolzano, Italy.

Part of the project objectives is designing a set of educational activities to promote computational thinking in primary and secondary schools. The plan is to design a progression of activities during the curriculum, which starts with unplugged activities and ends-up with hands-on exercises [CFPB18].

Computational Thinking has several concepts in common with the promotion of cognitive and relational Life Skills .Thus, COCONATS aims at promoting the acquisition of this second set of skills, which a vast literature indicates as essential not only for the structuring of ordinary life but also for the future workers.

Moreover, in line with the recent research interest in bringing Software Engineering to K-12 [PM19] to foster design skills and ability to manage the process towards the solution, the COCONATS project aims at understanding how Software Engineering can be fostered at different ages, at individual and collaborative level.

4 Proposed approach

The research work that inspired us in formulating our approach is the one by Siu-Cheung Kong [Kon19]; in this work, Kong asked elementary school pupils to write simple reflections on what types of Computational Thinking concepts they used in carrying out their tasks and projects. The author considers the following components of CT among those that could be possibly assessed at the end of the primary school curriculum:

1. problem formulating,
2. problem decomposition,
3. abstracting and modularizing,
4. algorithmic thinking,
5. reusing and remixing,
6. being iterative and incremental,
7. testing and debugging.

Among the existing ones, this approach is probably the most suitable for the age group 6-11, but it is still too complicated for first-grade children. For this reason, we propose to match the educational contribution of the CT model and the problem-solving skills connected to it with the curriculum, and therefore also with the educational needs of the specific school. The strength of this approach is not fragmenting these educational objectives.

Taking into account these considerations, the age of the pupils and the expected educational outcomes, we propose to evaluate the activities specifically prepared explicitly concerning these concepts: algorithm, generalization, problem-solving, creativity.

As shown in Figure 1 we, therefore, break up the problem of abstraction by evaluating: algorithmic thinking, problem-solving, and creativity. These aspects are further detailed in the remaining part of this section.

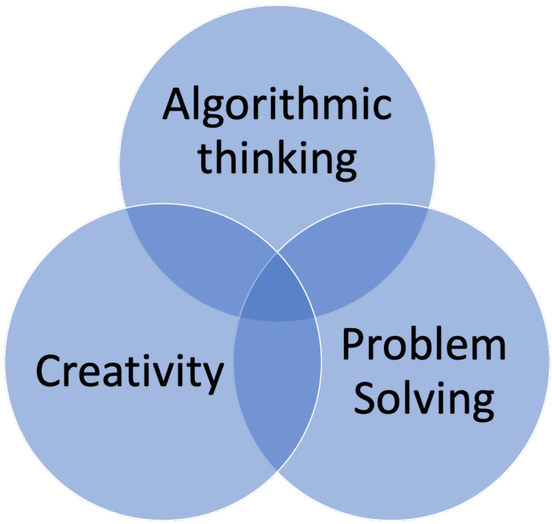

Figure 1: The three aspects considered in the proposed CT assessment model.

4.1 Algorithmic Thinking

In 2009, P. J. Denning [Den09] stated that algorithmic thinking is the basic idea behind computational thinking. In the computer science field, an algorithm is defined as any well-defined sequence of actions that take a set of values as input and procedures some set of values as output [RH14]. An algorithmic view of the problem-solving process is valuable because it facilitates many activities essential to daily life. Problem-solving has been associated to CT in recent literature, for example in [KCÖ17, RGMLR17].

For the evaluation of the algorithm, it has been shown that the researchers' observation plays a crucial role. There are references to the validity, in this field, of the researchers' observation [Bur12, FGM13] and to the progress of the children's work; however, the object of assessment is only their finished product.

One possible criterion, in this case, could be represented by the information collected by using a think-aloud protocol [EA17]: when the researcher thinks that a child has made with this activity a relevant experience of progressive sequences to achieve a goal (for example, an artifact), she/he could ask the child to verbalize that sequence of actions.

4.2 Problem-solving

In recent literature, Computational Thinking has been associated with problem-solving [KCÖ17, RGPGJF17]; nevertheless, the acquisition of problem-solving skills is still under discussion. A process frequently described is the 7-step process of Pretz et al. [PNS03], which consists of the following seven steps:

1. recognition or identification of a problem,
2. definition and mental representation of the problem,
3. development of a strategy to solve the problem,
4. organization of knowledge concerning the problem,
5. allocation of mental and physical resources to solving the problem,
6. monitoring of progress toward the goal,
7. evaluation of the solution for accuracy.

4.3 Creativity

This concept shares with Computational Thinking the ability to add original solutions or improvements to a simple work or artifact. However, we are persuaded that creativity also helps to overcome problems creatively or, in other words, it can be considered a part of the problem-solving skill. The creative ability is a powerful resource to face personal and social situations, converting them in growth and learning opportunities. In this respect, WHO defines creative thinking as a fundamental life skill that "contributes to both decision making and problem-solving by enabling us to explore the available alternatives and various consequences of our actions or non-action"; furthermore, it "can help to respond adaptively and with flexibility to the situations of our daily lives" [O$^+$94].

5 Emerging problems and possible solutions

The main question emerging is: *which methods are appropriate for evaluating the CT components in each dimension for first-year elementary school?*

To answer this research question, we believe that we first need to address some sub-questions:

1. Which didactic activities are functional for the purpose?
2. Which activities are the most appreciated and effective?
3. Which components of the model could present difficulties?
4. What indications emerge for the creation of a pattern for assessment in this age group?

In the remaining part of this section, we describe some lessons learned from the first part of the COCONATS project, together with possible problems (and solutions) that might emerge while working to provide an answer to this research question.

5.1 Activities.

The first activities that we have designed in the context of the COCONATS project for first-year elementary school are manipulative activities using cubes (see Figure 2).

Specifically, we have adopted cubes with square holes on each face, and a single, connecting stud located off-center so that children can connect the faces in any way they choose. The cubes come in two basic shapes. The first is a cube that measures two centimeters on each side; the second a right prism where the base is a right-angled isosceles triangle with two equal rectangular faces of the same measure as those of the cube. A quick note on the technical side refers to the fact that cubes used in pre-school environments are exactly four times bigger than those we use with higher school level and for educational robotics experiments.

We believe that this is a particularly suitable tool, because children have to solve various problems to carry out the construction, as the cubes are equipped with a single protuberance for the attachment between one and the other, so it is necessary not only design the object to be made but also found a solution to hook the cubes. An algorithm of this type, the phases of which can be observed and a finished product can be obtained, can subsequently be obtained also through other activities, linked to other disciplinary fields, such as, the natural sciences.

Figure 2: An example of manipulative activities using cubes.

5.2 Setting.

Our opinion is that children should carry out these activities in groups of 2, to be observed and assisted by researchers almost individually.

The activities should be carried out not in a class, but in a specially prepared setting, bright and welcoming, to make children feel at ease and to present the activity as a moment of creative play. Children can build on the ground or a bench. The cubes that we have chosen are usually welcome by children because they look like Lego cubes, which are generally already known and accepted as suitable play.

5.3 Outcome assessment.

As above mentioned, the skills that the children built in this age are already many. Thus, it is very problematic to understand if a positive result in the field of CT is to refer to the activities prepared for the purpose or if a series of extra-school factors affect the correct assessment. Moreover, this age represents a delicate phase in which children learn at the same time how to write, read, count, and be with others. In this process, each child has a different pace; therefore, individual differences should be taken into account by an assessment strategy.

As previously noted, it is tough to think of an evaluation of the process because too many individual variables affect the process in this age group. As to the process, the only description, but not the evaluation, is grounded on the observation of the researchers.

We believe that the alternative of the outcome's assessment remains the only possible for this age-level. In our context, for the realization of constructions with cubes, outcome assessment means:

- Building the object according to the project, respecting its operational sequences;
- Finishing the work and present it.

5.4 Strategy.

Based on these considerations, for the three aspects that we proposed to evaluate (i.e., algorithm, problem-solving, and creativity), we recommend proceeding as follows.

The *Algorithm* that we believe to be correct to evaluate is how the product fits the original project. This allows us to observe the following phases:

- construct by following instructions and topological concepts;
- in the game with the cubes, establish a sequence of actions to complete the chosen construction;
- build freely with the cubes explaining the project before starting, with the possibility of using the instruction booklet;
- reproduce a figure according to a two-dimensional model;
- debugging: knowing how to correct.

As to *Problem-Solving*:

- When there is a problem, pupils stop and think about how to solve it;
- They produce various options to solve it;
- They correct wrong solutions;
- They detect a problem;
- They elaborate a strategy (verbalizing it);
- They apply a strategy;
- They check if it works.

As to *Creativity*:

- children find original solutions and add personal and relevant elements to the construction;
- they try to combine similar artifacts by categories (for example, a tree and a house).

The following section draws conclusions from this work, also proposing possible directions for future work.

6 Conclusion and Future Work

This discussion aims to highlight the open question of Computational Thinking assessment at lower school levels, a context in which the variables are so many that it is challenging to allow assessments other than outcome assessments.

We believe that the evaluation of the finished product is the most reliable; however, it is essential that the researchers, together with the teacher, observe and compare the results. Indeed, school teachers can assess whether the concepts expressed in the activities are in line with the plan for the specific class and whether the objectives correspond to them, to avoid overwhelming children with requests for performance that go beyond their real capabilities.

Future perspectives encourage us to continue to think about possible evaluation of the implementation of CT even at lower levels of education, and in those ages where cognitive development such as abstraction has not yet unfolded.

We, therefore, intend to propose similar activities in other schools in order to broaden our sample, and possibly create and test an evaluation framework that can be disseminated to other interested parties.

References

[A$^+$16] Computer Science Teachers Association et al. Operational definition of computational thinking, 2016.

[BFCP18] Demis Basso, Ilenia Fronza, Alessandro Colombi, and Claus Pahl. Improving assessment of computational thinking through a comprehensive framework. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, page 15. ACM, 2018.

[BR12] Karen Brennan and Mitchel Resnick. New frameworks for studying and assessing the development of computational thinking. In *2012 Annual Meeting of the American Educational Research Association (AERA'12), Vancouver, Canada*, pages 1–25, Vancouver, Canada, 2012. AERA.

[Bur12] Quinn Burke. The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, 4(2):121–135, 2012.

[CFPB18] Alessandro Colombi, Ilenia Fronza, Claus Pahl, and Demis Basso. Coconats: Combining computational thinking didactics and software engineering in k-12. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, pages 162–162. International World Wide Web Conferences Steering Committee, 2018.

[CLN17] Isabella Corradini, Michael Lodi, and Enrico Nardelli. Conceptions and misconceptions about computational thinking among italian primary school teachers. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 136–144. ACM, 2017.

[Cro14] Dan Crow. Why every child should learn to code, feb 2014.

[Den09] Peter J Denning. Beyond computational thinking. *Communications of the ACM*, 52(6):28–30, 2009.

[EA17] David W Eccles and Güler Arsal. The think aloud method: what is it and how do i use it? *Qualitative Research in Sport, Exercise and Health*, 9(4):514–531, 2017.

[FGM13] Georgios Fessakis, Evangelia Gouli, and E Mavroudi. Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63:87–97, 2013.

[FIC17] Ilenia Fronza, Nabil El Ioini, and Luis Corral. Teaching computational thinking using agile software engineering methods: A framework for middle schools. *ACM Transactions on Computing Education (TOCE)*, 17(4):19, 2017.

[FP18] Ilenia Fronza and Claus Pahl. Envisioning a computational thinking assessment tool. In *CC-TEL/TACKLE@ EC-TEL, CEUR Workshop Proceeding 2190*, 2018.

[Gon15] Marcos Román González. Computational thinking test: Design guidelines and content validation. In *Proceedings of EDULEARN15 conference*, pages 2436–2444, 2015.

[Gro15] Shuchi Grover. "systems of assessments" for deeper learning of computational thinking in k-12. In *Annual Meeting of the American Educational Research Association*, pages 1–9, Washington, DC, 2015. AERA.

[Gro17] Shuchi Grover. Assessing algorithmic and computational thinking in k-12: Lessons from a middle school classroom. In *Emerging research, practice, and policy on computational thinking*, pages 269–288. Springer, 2017.

[HL15] Roxana Hadad and Kimberly A Lawless. Assessing computational thinking. In *Encyclopedia of Information Science and Technology, Third Edition*, pages 1568–1578. IGI Global, 2015.

[ICD+95] Barbara A Israel, K Michael Cummings, Mark B Dignan, Catherine A Heaney, Daniel P Perales, Bruce G Simons-Morton, and Marc A Zimmerman. Evaluation of health education programs: current assessment and future directions. *Health education quarterly*, 22(3):364–389, 1995.

[KA19] Siu-Cheung Kong and Harold Abelson. *Computational Thinking Education.* Springer, 2019.

[KCÖ17] Özgen Korkmaz, Recep Cakir, and M. Yasar Özden. A validity and reliability study of the computational thinking scales (cts). *Computers in Human Behavior*, 72:558 – 569, 2017.

[Kon19] Siu-Cheung Kong. Components and methods of evaluating computational thinking for fostering creative problem-solvers in senior primary school education. In *Computational Thinking Education*, pages 119–141. Springer, 2019.

[MRH15] Andreas Mühling, Alexander Ruf, and Peter Hubwieser. Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the workshop in primary and secondary computing education*, pages 2–10. ACM, 2015.

[Net13] Warren Nettleford. Primary school children learn to write computer code, jul 2013.

[O+94] World Health Organization et al. Life skills education for children and adolescents in schools. pt. 3, training workshops for the development and implementation of life skills programmes. Technical report, Geneva: World Health Organization, 1994.

[PM19] David Parsons and Kathryn MacCallum. *Agile and lean concepts for teaching and learning: Bringing methodologies from industry to the classroom.* Springer, 2019.

[PNS03] Jean E Pretz, Adam J Naples, and Robert J Sternberg. Recognizing, defining, and representing problems. *The psychology of problem solving*, 30(3), 2003.

[RGMLR17] Marcos Román-González, Jesús Moreno-León, and Gregorio Robles. Complementary tools for computational thinking assessment. In *Proceedings of International Conference on Computational Thinking Education (CTE 2017), S. C Kong, J Sheldon, and K. Y Li (Eds.). The Education University of Hong Kong*, pages 154–159, 2017.

[RGPGJF17] Marcos Román-González, Juan-Carlos Pérez-González, and Carmen Jiménez-Fernández. Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, 72:678–691, 2017.

[RGPGMLR18] Marcos Román-González, Juan-Carlos Pérez-González, Jesús Moreno-León, and Gregorio Robles. Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80:441–459, 2018.

[RH14] David Riley and Kenny A Hunt. *Computational thinking for the modern problem solver*. Chapman and Hall/CRC, 2014.

[Win06] Jeannette M. Wing. Computational thinking. *Comm. ACM*, 49(3), 2006.

[Win14] Jeannette M. Wing. Computational thinking benefits society, January 2014.

[WW15] David Weintrop and Uri Wilensky. Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In *ICER*, volume 15, pages 101–110, 2015.