Assessment of Computational Thinking in K-12 Context: Educational Practices, Limits and Possibilities - A Systematic Mapping Study

Lúcia Helena Martins-Pacheco, Christiane Annelise Gresse von Wangenheim and Nathalia da Cruz Alves
Department of Informatics and Statistics, Federal University of Santa Catarina, Campus Universitário s/n. Trindade, Florianópolis, Brazil

Keywords: Computational Thinking, Assessment, K-12, CT Approaches, CT Instruments.

Abstract: The computational thinking (CT) concept has been the basis for several studies in the K-12 educational context. However, there are many questions that need to be deepened to attend K-12 educational demands. One great challenge is concerning assessment. Aiming to contribute to understanding this issue we present a systematic mapping study. We found 46 articles that approach assessment in this context, and we extract this information. The vast majority are recent publications, there is no consensus in CT characteristics, block-based languages are the most commonly used tool, instruments for assessment that are more used are pre and post-tests/questionnaires/surveys; samples sizes are usually small, and there is some psychometric rigor in just a few studies. Generally, the CT approaches were an isolated course or application, and their length of time was very different. Pedagogical foundations concerning the cognitive development stages and principles of knowledge structuration were rare. In addition, questions as "what has to be taught to the youngster?" and "how to teach and to assess in alignment with K-12 goals?" were not appropriately answered. Therefore, there are many research opportunities for the further development of this field.

1 INTRODUCTION

According to Wing (2006)'s seminal article, "*Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science*" (p.33). This primordial idea has impacted academic and educational groups (NRC, 2010); (NRC, 2011); (Brennan and Resnick, 2012); (CSTA, 2012); (CSTA, 2016) that have tried to comprehend how to make it feasible in educational practices. This effort is justified due to the widespread use of electronic devices such as desktops, notebooks, tablets and cell phones all around the world. Also, this is a booming job market that can potentially offer opportunities for brilliant careers.

Moreover, technological developments in computer science interactive programming environments allowed for other ways to use computers. Several user-friendly visual languages were created, which constitute ludic environments that are easily programmable. The formal syntax that traditionally needed to be programmed can now be replaced for graphical tools, such as block-based language, which is very intuitive and can quickly show results. As examples of those environments, we can cite Scratch, Blockly, AppInventor, and Snap! (von Wangenheim et al. 2017A and 2017B; Alves et al., 2018). Therefore, programming no longer requires exhaustive high cognitive reasoning efforts and, consequently, it makes it possible to focus on logic instead of strict mechanical writing of computer commands (Lye and Koh, 2014).

Despite the boost that academic groups have given to CT, there are still many challenges in terms of pedagogical and psychological educational practices (Grover and Pea, 2013; Shute et al., 2017; Seiter and Foreman, 2013). For example, questions as "how to match teaching and learning with the cognitive development of children?", "how to train teachers to motivate students to engage in learning CT?", "what kind of contents must be taught?", "how to assess what was learned?" urge to be answered.

Aligned with this concern, in this study we present

Martins-Pacheco, L., von Wangenheim, C. and Alves, N.
Assessment of Computational Thinking in K-12 Context: Educational Practices, Limits and Possibilities - A Systematic Mapping Study.
DOI: 10.5220/0007738102920303
In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)*, pages 292-303
ISBN: 978-989-758-367-4

some important issues related to educational assessment for CT in the K-12 context. The aim is to gather data about what has been done to assess CT and further the discussion on the topic. Firstly, we present some relevant aspects of educational assessment followed by a systematic mapping review, data analysis, discussion, and the conclusion.

2 CT ASSESSMENT

According to Brookhart and Nitko (2015), "*assessment provides information for decisions about students; schools, curricula, and programs; and educational policy*," aiming to improve the teaching-learning process. A great variety of assessment methods can be "*used to gather information: formal and informal observations of a student; paper-and-pencil tests; a student's performance on homework, lab work, research papers, projects, and during oral questioning; and analyses of a student's records*." In addition, this information may be used to polish the entire educational system.

According to the same authors, evaluation **"***is the process of making a value judgment about the worth of a student's product or performance"* and *"may or may not be based on measurements or test results*." Evaluation can be influenced by bias, subjectivity, and inconsistency. In contrast, assessments are based on tests and measurements, which tend to be standardized and objective, consequently reducing the influence of subjectivity.

It is important to highlight two kinds of assessment: formative assessment and summative assessment. According to Dixson and Worrel (2016), formative assessment aims to improve teaching and learning, to diagnose student difficulties (ongoing before and during instruction), and to ask, "*what is working*" and "*what needs to be improved*." As for summative assessment, it focuses on the evaluation of learning, placement and promotion decisions. It is usually formal, cumulative, after instruction, and asks "*does student understand the material*" and "*Is the student prepared for the next level of activity*." The psychometric rigor in summative assessment is higher than in formative assessment.

On the topic of assessing CT, there are five relevant studies. Araújo et al., (2016) made a systematic mapping study about assessing computational thinking abilities, that analyzed 27 studies. Alves et al. (2018) present a systematic mapping study looking for approaches to assess CT competencies in K-12 education based on code analysis. They identified 12 approaches that mostly focused on the assessment of the Scratch program use. Kalelioglu et al., (2016) analyzed 125 papers about CT aiming to define a framework for CT. According to them "*CT literature is at an early stage of maturity, and is far from either explaining what CT is, or how to teach and assess this skill*." Grover and Pea (2013) framed discourses on CT in K-12 education, identified gaps in research, and articulated priorities for future inquiries. Finally, Shute et al. (2017) found a variety of definitions, interventions, assessments, and models for CT. They proposed a definition and a model of CT to inform instructions and assessments that can be used across disciplines and educational settings.

3 EXECUTION OF SMS

In order to discover the state of the art on CT assessment in K-12 education, we conducted a systematic mapping study (SMS) according to Petersen et al., (2008) definition.

3.1 Definition of the Mapping Protocol

3.1.1 Research Question

Which approaches exist for the assessment of computational thinking (CT) in the context of K-12 education? We unfold this research question into the following analysis questions.

3.1.2 Pedagogical Approaches

AQ1: Which approaches exist and what are their characteristics?
AQ2: Which theoretical, pedagogical foundations are used?

3.1.3 Assessment Approaches

AQ3: Which concepts of CT are assessed and how they are assessed?
AQ4: Which assessment methodology is used, and which instruments are used?
AQ5: Are there instructional assessments and feedbacks?

3.1.4 Measurement Approaches

AQ6: How does the instrument assign weights in the assessment?
AQ7: Are there psychometric bases in the assessment?

3.1.5 Data Source

We examined all published English-language articles that were available on Scopus, Web of Science, Wiley Online Library, ACM Digital Library, IEEE Xplore, APA Psycnet, Science Direct, with access through the CAPES Portal[1] and free-access. To increase publication coverage including grey literature, we also used Google Scholar, which indexes a large set of data across several different sources as suggested by Haddaway et al., (2015).

3.1.6 Inclusion/Exclusion Criteria

We considered only English-language articles that presented an approach about the assessment of CT in K-12. We considered articles that were published after 2005, because the concept of "computational thinking" was only proposed by Wing in March 2006 (Wing, 2006). In our searches, we established that "computational thinking" must be in the title of the article. We excluded approaches that act out of K-12 context or approaches focusing on other educational contexts, such as higher education or teacher training, given that they are out of the scope of our research objective.

3.1.7 Quality Criteria

We considered only articles that present substantial information on the presented approach, to enable the extraction of relevant information for the analysis questions. Articles that provided, for example, only a summary of a proposal and for which no further information could be found, were excluded.

3.1.8 Definition of Search String

According to our research objective, we defined the search string by identifying that "computational thinking" must be in the title of the article. The terms "assess," "assessment," "assessing" were searched in the title and other fields. We highlight that in some bases as, for example, IEEEXplore using "assess" also returns "assessing" and "assessment" but in other bases just the exact word. Also, we search for psychometrics studies in APA Psycnet base using in title "computational thinking" and terms as "psychometrics," "validity" and "reliability." We do not use "evaluation" because we consider the before mentioned definition that distinguishes "assessment" and "evaluation" (Brookhart and Nitko, 2015). Using these keywords, the search string was calibrated and adapted in conformance with the specific syntax of each of the databases.

3.2 Execution of the Search

The search has been executed in February 2018 by the first author and revised by the co-authors. The definition of the search string was handled together by all authors. The first author carried out the initial search that resulted in the selection of 310 articles but, as it expected, some of them appeared in several databases. Then we proceeded to analyze the title and the abstract, excluding those that were not related to K-12 context, some poster presentations or when only the abstract was available. In the first analysis stage, we reviewed titles, abstracts, and keywords to identify the articles that matched the inclusion criteria, resulting in 58 potentially relevant articles based on the results from all databases. Secondly, considering that we are interested in the article that deepens the assessment subject, we analyzed those that emphasize it. For this reason, 12 more articles were excluded, which resulted in the final selection of 46 articles.

4 DATA ANALYSIS

In this section, we present the distribution of the studies per year, and according to their focus, as well as discuss the analysis questions. The distribution of studies according to their year of publication is shown in Figure 1. In 2018 we found three articles since the search was done at the beginning of the year (February). More than 50% were published in 2016 and 2017, showing the increase in publications in this subject in the last years.

Considering the types of studies found, we classified them into eight categories, according to their focus. Figure 2 shows the distribution of the articles into these categories.

The most frequent category was "Implementation," followed by "Framework and Implementation." By "implementation" we mean the studies that dealt with practical approaches such as a course or a test application. By "framework" we mean the ones that present a theoretical conceptual structure to model computational thinking. Some of them are testing an

[1] Portal with access to scientific databases worldwide sponsored by Brazilian Education Ministry, only available for research institutions.

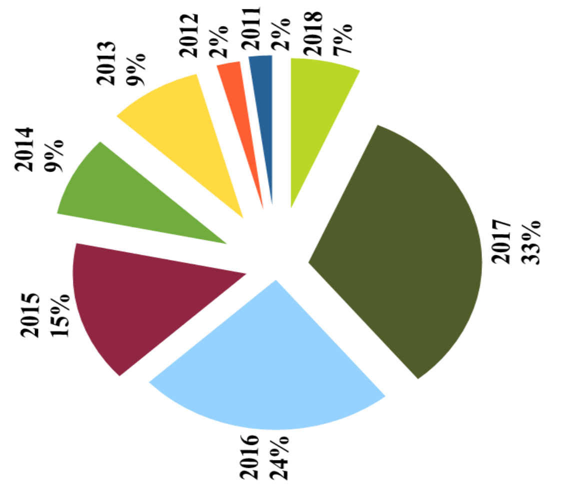

Figure 1: Amount of publications include in defined criteria per year.

instrument in comparison to a formal instrument, that already had acknowledged validity and reliability, such as in Moreno-León et al. (2017) and Jiang and Wong (2017). Finally, "dataset comparison" refers to the analysis of standard examination databases in comparison to CT tests (Rodrigues et al., 2016).

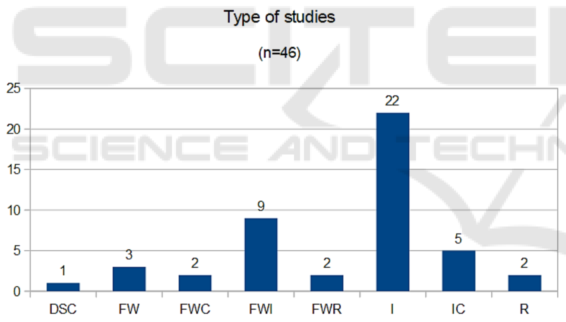

Figure 2: Articles distribution according to our classification. I-Implementation; FW-Framework; FWC-Framework with Comparison; FWI- Framework with Implementation (usually pilot implementation); FWR-Framework and Literature Review or Mapping; IC-Implementation with Comparison; R- Literature Review or Mapping; and DSC – Dataset Comparison.

AQ1: Which approaches exist and what are their characteristics?

To analyze the approaches and the theoretical pedagogical foundations we considered just the categories that involved implementation or framework. Table 1 summarizes the most frequent approaches found and Table 2 summarizes the most frequently used tools in the studies. It is important to consider that some articles show more than one approach or more than one tool. Most implementation studies apply CT principles in non-computer science curricular courses (e.g., Aiken et al., 2013; Werner et al., 2012). Also, unplugged activities were usual (e.g., Brackmann et al., 2017; Feldhausen et al., 2018; Jiang and Wong, 2017; Rodriguez et al., 2017), as well as block-based programming, such as Scratch.

We classified 17 approaches. Most of them (12) just appear once in our search (age-appropriate computational activities; agile software engineering; blended learning; interest-driven creator (IDC) theory; modeling and simulations; mind maps; SGD (Scalable Game Design); scaffold learning / a set of hypertext resources and formative assessment quizzes in the system; storyboard; Student-driven instruction tutorial; virtual robot; Zombinis puzzles). Therefore, there was a great variety of approaches, that was only used once in our literature search.

Table 1: Most frequent CT approaches used in the studies.

Approach	**Fq.**
CT context of non-computing - curriculum	**19**
Unplugged activities	**4**
Agent-based modelling	**3**
Robot	**3**
Drag-and-drop programming tools	**2**

As for the tools used, we found ten different options. Seven of them only appear once in our search: Arduino, Game Maker Studio, Lego EV3 Robotic Kit, Lego Mindstorms NTX 2.0, LOGO, Mighty Micro Controller, NetLogo. The other three tools that are cited more than once are shown in Table 2. The most cited tool is Scratch.

Table 2: Most frequent tools used in the studies.

Tool	**Fq.**
Scratch	**9**
Alice, Storytelling Alice	**5**
Python, VPython	**2**

Regarding the implementation length of time, there were significant variations. The range varied from some hours (e.g., Jenson and Droumeva (2016), that took about 20 hours) to years, in an incremental teaching-learning process (e.g., Grgurina et al. (2015) and Feldhausen et al. (2018) that took three years). Therefore, it is difficult to compare its impact in the processes. Shute et al. (2017) had similar findings.

AQ2: Which theoretical pedagogical foundations are used?

Regarding theoretical pedagogical foundations, many articles do not explicit the chosen principles. Some of them are based on CT practices and interaction with a system or a group. Other authors follow some assumptions concerning the following questions: "how someone learns?", "how contents have to be taught?", "how to align human development in the teaching-learning process, especially in childhood?", "how to insert CT practices in the educational context?", and so on. Grover and Pea (2013) asked in their study what "*can we expect children to know or do better once they've been participating in a curriculum designed to develop CT and how can this be evaluated?*".

We classified 18 approaches. Most of them (10) just appear once in our search: EDM/LA-BBPE (Grover et al., 2017); Gamma et al.,(1995) in (Seiter and Foreman, 2013); Comer et al., (1989) in (Grgurina et al., 2015); Computing Progression Pathways - Dorling; Walker, (2014) in (Bilbao et al., 2017); PSL (problem-solving learning); Salomon and Perkins, (1987) in (Witherspoon et al., 2017); Zone of Proximal Development (ZPD); Webb, (2010); Linn, (1985) in (Werner et al,, 2012); 5E Instructional Model[2] in (Ouyang et al., 2018), and ISTE Standard. Table 3 shows the chosen foundation of each study that deals with implementation or framework that appeared more than once in our search.

The most usual foundation was Constructivism and Constructionism, which are traditional in pedagogy and educational psychology approaches. Constructivism defends that the learner has an active role in creating, and in changing the knowledge representation. Constructionism (Papert 1980; Papert 1991) considers that knowledge construction is related to concrete and practical action, resulting in a real product. LOGO language is the main tool for this approach.

Bloom's taxonomy, it is a hierarchical organization regarding the educational goals. It is very popular, especially in the USA. Scaffolding approaches consider that, in the beginning, learners have to be supported to facilitate understanding and making it possible to consolidate knowledge representation process (Lye and Koh, 2014).

Top-down and bottom-up approaches are used in several levels of knowledge structures. For example, Basogain et al. (2018) took explicitly top-down and bottom-down approaches in their study. Problem based learning and Game based learning are also top-down approaches. Usually, it is applied in a real-world problem. It induces the learner to established strategies such as decomposition, modeling, reuse and so on to solve the problem. CSTA provides curriculum guidelines for concepts and practices of computing, including computational thinking

Table 3: Pedagogical foundations used in the studies.

Foundation	Fq.
Constructivism / constructionism	7
Game-based learning	7
Bloom's taxonomy	3
Learner-centered	3
Peer collaboration	3
Brennan and Resnick (2012)	2
CSTA (2011)	2
Adaptive scaffolding	2

AQ3: Which concepts of CT are assessed and how they are assessed?

CT concepts were classified: abstraction; algorithm; data representation/collection/analysis; debugging; decomposition; events; flow control; loops, sequences and conditionals; modeling; modularity; parallelism; problem solving; reuse; synchronization; variables; and user interactivity. Also, some authors refer to CTSA or Brennan and Resnick (2012) concepts, but only take some aspects of them (e.g., Román-González et al. (2017)). Similarly, we consider that CTSA, CTt, and even Scratch as "umbrellas" for approaching CT. We do not address these concepts in depth here due to the size limitation of this paper, but for details, we suggest to refer to the following papers: Brennan and Resnick (2012), Grover and Pea (2013), Shute et al. (2017) and Alves et al. (2018).

Figure 3 shows the frequency of concepts in articles.

[2] Bybee, R. (1997). Achieving Scientific Literacy, Portsmouth, NH: Heinemann..

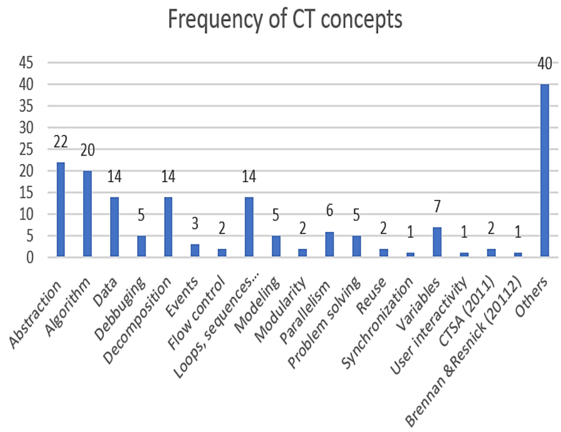

Figure 3: Frequency of CT concepts.

The way to assess CT depends on the approach. Some implementation studies use formative assessment during all process, others in some stages, others just in the end. There are several instruments for assessment, and they are analyzed in the next question.

AQ4: Which assessment methodology is used, and which instruments are used?

At first, we analyze the articles that proposed a model or those that have some psychometric rigor or potential for that.

REACT (Real Time Evaluation and Assessment of Computational Thinking) proposed by Koh et al. (2014) "*enables formative assessment of game design projects*" and "*teacher summative assessment of student game design projects*" also it "*can be used by the teacher for effective in-class management through intervention*" and it "*can lead student self-assessment and peer interaction, and teacher/student 2 way validation*". The authors' foundation is Vygotsky's Zone of Proximal Development (ZPD).

Computational Thinking using Simulation and Modeling (CTSiM) (Basu et al., 2014; Basu et al., 2015; Basu et al., 2017) is an open-ended learning environment for middle school students. They can choose different tools offered in the environment and construct their models. Also, the environment provides feedback and clues that make it easier for student to reach their learning goals. Some formative assessment is given during the teaching-learning process. The environment was tested and has demonstrated some psychometric properties.

The final model of CTt (CT test) is shown in Román-González et al., (2017). The test was applied to 1251 individuals and compared with Primary Mental Abilities (PMA) battery, and RP30 problem-solving test. As the authors assert "*we have provided evidence of reliability and criterion validity of a new instrument for the assessment of CT, and additionally we expanded our understanding of the CT nature through the theory-driven exploration of its associations with other established psychological constructs in the cognitive sphere*." Therefore, it is an interesting instrument with psychometric properties, but it is independent of the educational context.

DISSECT (DIScover SciEnce through Computational Thinking) is a project aimed at introducing students to computer science principles by establishing computational thinking (CT) as a problem-solving technique within middle school and high school Science, Technology, Engineering, and Mathematics (STEM) courses (Nesiba et al., 2015; Burgett et al., 2015). This project applies four assessments: (1) CT term recognition, (2) CT term definitions, (3) Likert job questions, such as "Getting a job in computing would allow me to...", (4) Likert interest questions: "How much interest do you have in the following?". Then the assessment is not just CT skills, but also other important skills for student's development. They used pre and post assessment and experimental groups and control groups to know about student's performance and to assess their process.

SDARE uses an instrument that contains 23 items, organized into 6 item sets. Among these items are 15 multiple-choice questions, and eight open-ended questions. Everyday scenarios and robotics programming are assessed by the instrument. It shows some psychometric properties.

Doleck et al., (2017) developed a CT scale that comprises 29 items and is divided into five dimensions: algorithmic thinking, cooperativity, creativity, critical thinking, and problem-solving. These items were scored on a 5-point Likert scale. Academic performance is also self-reported by students. Demographics and prior achievements (age, gender, high school GPA[3]) are used as the control variables in the model. The study aimed to investigate the relationship between CT skills and academic performance empirically. They did not find a strong correlation between the variables; however, the instrument has shown some psychometric properties.

We classified 13 approaches that did not present a specific model. Most of them (7) just appear once in our search: P2P assessment (Basogain et al., 2018); Paper and pencil test (Worrell et al., 2015) ; Online interactive assessment (Weintrop et al., 2014); Test (Basogain et al., 2018); Video analysis (Rowe et al.,

[3] Grade Point Average

2017); Written essay (Aiken et al., 2013) , and Data set (Doleck et al., 2017). The most frequent implementation instruments are shown in Table 4, just those appear more than once.

Table 4: Others implementation instruments.

Instrument	Fq.
Pre-post test/survey/ questionnaire	9
Interview	7
Survey/ questionnaire	5
Project/Design/Artifact resulted	4
Matched pair or paired groups	3
Self-assessment/ report	2

The more frequent instruments used are Pre- post-test/ survey/ questionnaire followed by interview instruments. The items could be scored by a number or qualitatively depending on the statistical methodology. It is common to do just one survey, questionnaire or test, showing the final results or state. Besides, interviews are also used allowing to understand the students point of view. It is interesting to highlight that the results of the assignments, such as project/design/artifact, are important inside the educational environment as assessment instruments. This kind of assignments could be associated with formative assessment, with constructionism or with scaffolding procedures, allowing students to engage in learning attitudes. Problem based learning and Game based learning approaches are usually chosen for this kind of assessment. In addition, achievement of goals tends to motivate students (Jiang and Wong, 2017). Another way to assess methods is by using matched pair or paired groups experiments, that are very traditional statistical methods.

Dr. Scratch (Moreno-León et al., 2015) analyzes concepts such as abstraction, logic, and parallelism scoring each concept based on a rubric. Open-ended ill-structured problems are checked by static code analysis. For each programming exercise, a set of concepts are analyzed. Therefore, Dr. Scratch provides an automatic assessment of the student's program.

The PECT approach presents a rubric to perform manual analysis for open-ended ill-structured problems (Seiter and Foreman, 2013). Based on Gamma et al., (1995), the model provides foundations for age-appropriate CT curriculum. The concept of design patterns categorizes the level of skill utilized in the student's design, "calculated by measurable evidence from programs written in Scratch." It is interesting to address the question related to "age-appropriateness" since in terms of psychological development is fundamental care. The scoring accounts for three levels: Basic; Developing and Proficient. The model was tested in 25 projects. Fairy Assessment approach (Werner et al., 2012) makes use of a rubric to assess the code for the open-ended well-structured problem. Fairy Assessment, being aimed for Alice programs, works with CT concepts of thinking algorithmically, and making effective use of abstraction and modeling. Students are engaged with CT in a three-stage progression called Use-Modify-Create.

Three-Dimensional Integrated Assessment (TDIA) framework, proposed by Zhong et al. (2016), aims to integrate three dimensions (directionality, openness, and process) into the design of effective assessment tasks. It uses three pairs of tasks: closed forward tasks and closed reverse tasks; semi-open forward tasks and semi-open reverse tasks; and open tasks with a creative design report and open tasks without a creative design report. This framework diversified assessment tasks and extended the theoretical basis for designing assessment tasks.

From this analysis, it is evident that there is a wide variety of implementation instruments for the assessment CT.

AQ5: Are there instructional assessments and feedbacks?

Only a few articles show the details about formative assessment, summative assessment and about feedbacks explicitly. Some authors are looking for an instrument with psychometrics properties; others are more interested in the process of learning and teaching CT, in a way to keep students motivated with the technological practices.

REACT (Koh et al., 2014) uses an embedded assessment for helping teachers to give a formative assessment and communicate students' progress. CTSiM (Basu et al., 2014; Basu et al., 2015; Basu et al., 2017) makes use of a mentor agent to give feedback to students during their interaction with the system. DISSECT (Nesiba et al., 2015; Burgett et al., 2015) applies four tests during the teaching-learning process, making adjustments possible in the student's performance. Fairy Assessment (Werner et al., 2012) uses survey, attendances and four tasks during the process of teaching-learning, to follow a student's performance and to give them feedback. TDIA (Zhong et al., 2016) uses scaffolding methods and three tests to follow the student's performance. Then, in fact, a formative assessment takes place in several practices, even though without being explicitly declared.

In this sense, even P2P (peer to peer) practices could facilitate formative assessment. Also, automatic assessment by code analysis could be performed in some kinds of formative assessments, quickly giving some clues about the decisions made by students to solve problems.

AQ6: How does the instrument use weights in the assessment?

The most usual approach to weight assessments is to score each item in a test or questionnaire, e.g., CTt (Román-González et al., 2017). CTt has a length of 28 items, and it addresses the following CT concepts: conditionals; defined/fixed loops; undefined/unfixed loops; simple functions; functions with parameters/variables. The score is calculated as the sum of correct answers along the 28 items of the test (minimum 0 and maximum 28). Werner et al., (2012) graded each task on a scale from zero to ten, with partial credit possible, resulting in a maximum score of 30.

Fronza et al., (2017) calculated *cyclomatic complexity* for each project and classified it as low, medium, high. CTSiM (Basu et al., 2014; Basu et al., 2015; Basu et al., 2017) calculates "*vector distance model accuracy metric*" to evaluate the difference between a reference of correctness and result presented. Doleck et al., (2017) use a Likert scale. Rodriguez et al., (2017) classified results as Proficient, Partially Proficient, and Unsatisfactory. Seiter and Foreman (2013) classify the assessment as Basic, Developing, Proficient.

Therefore, there several ways to weight or ponder the assessment depending on the objective of the study or implementation. Surveys and interviews also give feedback and are usually used for qualitative evaluation.

AQ7: Are there psychometric bases in the assessment?

Only four studies present psychometrics properties — three of them only partially, and one (CTt) more complete. The former uses as reference consolidated psychometrics instruments (PMA battery and RP30). Psychometrics properties are related to validity and reliability and depend on the theoretical construct. This construct must be reliable in modeling or in representing the psychological reality. And the statistical methodology allows generalization, according to the size of the sample (n).

Among the searched studies, 30 present an application of instruments to individuals. Except for Román-González et al., (2017) work (n=1251), the studies analyzed small samples. We calculated the distribution of the sizes of the samples by box-plot parameters: minimum=5; first quartile= 26; median=88,5; third quartile= 149; maximum=441.

Therefore, just 15 studies (50%) have "n" greater than 88 individuals, and eight studies less than 26 individuals. So, the statistical representativity is not strong for psychometric properties.

5 DISCUSSION

Considering our research question, "*Which approaches exist for assessment computational thinking (CT) in the context of K-12 education?*" we gathered publications that the use of Wings' concept of CT and assessment in K-12 are recent. The oldest article dates from 2011, and therefore it is a very new field. Sixteen articles (35% of total) have proposed frameworks for CT, which indicates the need for theoretical support to cope with this issue.

Approaches for CT teaching-learning in K-12 were classified into 17 categories, but some work together. The most common approach is "*CT within the context of non-computing disciplines*" followed by "*unplugged activities*," "*agent-based modeling*" and use of "robots." These findings differ from Araújo et al., (2016), they found that "*programming courses are the most common pedagogical approaches to promote CT for K-12 students*". But Kalelioglu et al., (2016) found that "*the main topics covered in the papers composed of activities (computerised or unplugged) that promote CT in the curriculum*."

The most commonly used tool is "Scratch" followed by "Alice, Storytelling Alice." Therefore, teaching and assessing CT using block-based language probably is the most interesting approach, because these environments are usually free, easy to use, with a graphic appeal, some of them have automatic code analysis, making quick feedback possible.

Regarding pedagogical, theoretical foundations, we can highlight the choice of constructivism and constructionism, followed by Game based learning. This last could be understood encompassed by constructionism principles. These findings agree with Kalelioglu et al., (2016) position that "*Gamed-based learning and constructivism were the main theories covered as the basis for CT papers*." The construction of games or game playing could be an interesting way to associate higher cognitive process, such as abstraction, with concrete results, besides allowing for some fun. However, in general, the articles analyzed did not deepen the pedagogical approaches, and many of them did not even show any concern about this issue.

CT aspects or concepts are represented by a great variety of concepts, and some of them are synonyms. Several authors (e.g., Alves et al., (2018), Shute et al., (2017)) point out the lack of consensus among concepts, and they show some concerns about this. In computer science, ill-defined concepts are more difficult to deal with and constrain standardization. They make comparing and repeating experiments difficult, as well as impact educational practices. Meanwhile, it brings up a vast amount of possibilities to solve problems, allowing for more creative solutions.

Excluding "*others*" (see Figure 3), the most frequent CT concept was "*abstraction*" followed by "*algorithm*," "*data representation/ collection/ analysis*," "*decomposition*" and "*loops, sequences, and conditionals*." Araújo et al., (2016) found that the abilities more assessed are solving problem, algorithms, and abstraction. In our analysis inside "*others*" (see Figure 3) there are several concepts that cannot be framed within those that we presented. Some are very specific, and others make use of high-level structures or top-down approaches.

Regarding "assessment methodologies" we found that the majority deals with isolated experiences. That is, they are not framed within the whole educational context, and, therefore, do not provide *"information for decisions about students; schools, curricula, and programs; and educational policy*" (Brookhart and Nitko, 2015). Some are long-term projects and have good theoretical support, others are looking for a standardized test with psychometric rigor, while some are just practices in a computational environment. The assessments take place in a teaching-learning context, and the methodology and results depend on the length time of courses and the goals of each approach.

The more usual assessment instruments are Pre or post-test/ survey/ questionnaire followed by interviews, surveys, and questionnaires (just one stage). These are traditional ways to measure a student's performance, by means of a numerical score. It is an interesting option, which makes statistical and numerical data analysis possible. This gives clues about the student's performance and the effectiveness of the used processes. It is also possible to consider qualitative variables, using Likert scales, for example. Araújo et al., (2016) found that "*codes and multi-choice questionnaires are the most common artifacts for assessing CT abilities*." And Shute et al., (2017) found that "*Questionnaires and surveys are the* most commonly used measure for knowledge of and/or attitudes towards CT"

Some approaches are concerned with formative and summative assessments, making use of educational intervention during all their processes. This kind of feedback tends to be more efficient and generally is based on more than just the student's cognitive aspects. Alves et al., (2018) noticed a lack of consensus on the assessment criteria and also in the instructional feedback. They point to the need to promote a more comprehensive feedback process. Shute et al., (2017) affirm that: "*Because of the variety of CT definitions and conceptualizations, it's not surprising that accurately assessing CT remains a major weakness in this area. There is currently no widely-accepted assessment of CT. This makes it difficult to measure the effectiveness of interventions in a reliable and valid way*". Also, Kalelioglu et al., (2016) pointed out that a personal view about CT is very common in the papers.

Regarding psychometric rigor, most of the studies deal with small samples, which does not assure statistical representativity for the generalization of the results, which concerns the validity and reliability of the assessment instruments.

6 CONCLUSION

CT in K-12 context associated with new tools of programming is becoming an interesting possibility to teach the principles of Computer Science for a younger audience. Thus, the conceptual "umbrella" of computational thinking – CT - is important and has been evoking a great number of researches until now. So far, Wing (2006) article has been cited over 4,200 times. Perhaps its greatest contribution was in the assertion that reasoning for solving a problem in computer science could be useful in several contexts and does not necessarily need to be formal, strict and logically complicated.

Nonetheless, there are programming approaches in K-12 that are not based on Wing's article. For example, Mühling et al., (2015) present a preliminary version for a psychometric test, for measurement of basic programming abilities. It already has been experimentally applied in a secondary school in Germany. They did not use Wing's CT definition as a reference. Therefore, researches in computer science in K-12 context should consider not only the "CT umbrella," but also keep synergy with educational principles, as well as include other approaches of teaching computer science for youngsters, and non-majors.

Studies do not usually approach pedagogical foundations concerning the cognitive development stages and principles of knowledge structuration. Grover and Pea (2013) consider: "*much remains to be done to help develop a more lucid theoretical and practical understanding of computational competencies in children. What, for example, can we expect children to know or do better once they've been participating in a curriculum designed to develop CT and how can this be evaluated? These are perhaps among the most important questions that need answering before any serious attempt can be made to introduce curricula for CT development in schools at scale*". In addition, questions as "what has to be taught to the youngster?" and "how to teach and to assess in alignment with K-12 goals?" were not yet appropriately answered.

Due to several new technologies, there are a lot of different possibilities that challenge educators to explore new ways of learning and teaching. In this sense, the present study intends to contribute to understand what is CT assessment in educational K-12 context, showing that there are many research opportunities for the further development of this field.

Finally, we find that there is a need to expand the conceptual foundations that underlie teaching CT in K-12. The conceptual gaps might fuel innovative ideas for new researches, producing more scientific knowledge, enlarging the possibilities for everyone.

ACKNOWLEDGMENTS

The authors would like to thank Renata Martins Pacheco for her help with formatting and reviewing the English version of the final text.

REFERENCES

Aggarwal, A., Gardner-McCune, C. and Touretzky, D. S., 2017. *Evaluating the Effect of Using Physical Manipulatives to Foster Computational Thinking in Elementary School.* Seattle, ACM.

Aiken, J. M., et al., 2013. *Understanding student computational thinking with computational modeling.* Sidney, AIP, pp. 46-49.

Alves, N. D. C., von Wangenheim, C. G. and Hauck, J. C. R., 2018. Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. *Informatics in Education,* p. Accepted for publication in 2018..

Araujo, A. L. S. O., Andrade, W. L. and Guerrero, D. D. S., 2016. *A systematic mapping study on assessing computational thinking abilities.* Erie, IEEE.

Atmatzidou, S. and Demetriadis, S., 2016. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems,* Volume 75, pp. 661-670.

Basogain, X., Olabe, M. Á., Olabe, J. C. and Rico, M. J., 2018. Computational Thinking in pre-university Blended Learning classrooms. *Computers in Human Behavior,* Volume 80, pp. 412-419.

Basu, S., Biswas, G., Kinnebrew, J. and Rafi, T., 2015. *Relations between modeling behavior and learning in a Computational Thinking based science learning environment.* Hangzhou, ICCE, pp. 184-189.

Basu, S., Kinnebrew, J.S. and Biswas, G., 2014. *Assessing student performance in a computational-thinking based science learning environment.* Honolulu, Springer, pp. 476-481.

Basu, S., Biswas, G. and Kinnebrew, J., 2017. Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and User - Adapted Interaction,* 27(1), pp. 5-53.

Bennett, V. E., Koh, K. and Repenning, A., 2013. *Computing creativity: divergence in computational thinking.* Denver, ACM.

Bilbao, J. et al., 2017. Assessment of Computational Thinking Notions in Secondary School. *Baltic Journal of Modern Computing,* 5(4), pp. 391-397.

Brackmann, C. P. et al., 2017. *Development of Computational Thinking Skills through Unplugged Activities in Primary School.* Nijmegen, ACM Workshop on Primary and Secondary Computing Education.

Brennan, K. and Resnick, M., 2012. *New frameworks for studying and assessing the development of computational thinking.* Vancouver, AERA.

Brookhart, S. M. and Nitko, A. J., 2015. *Educational Assessment of Students.* 7th ed. Des Moines: Pearson.

Burgett, T., et al., 2015. *DISSECT: Analysis of pedagogical techniques to integrate computational thinking into K-12 curricula.* El Paso, IEEE.

Chen, G. et al., 2017. Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education,* Volume 109, pp. 162-175.

Comer, D. E., et al., 1989. Computing as a Discipline. *Communinications of ACM,* 32(1), pp. 9-23.

CSTA and ISTE, 2011. *Operational definition of computational thinking for K-12 education..* [Online] Available at: http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf [Acesso em 10 November 2017].

CSTA K–12, 2016. *Computer Science Framework.* [Online] Available at: http://www.k12cs.org [Acesso em 10 November 2017].

CSTA, 2011. *K-12 computer science standards.* [Online] Available at: http://csta.acm.org/Curriculum/sub/Curr

Files/CSTA_K-12_CSS.pdf [Acesso em 10 November 2017].

Dixson, D. D. and Worrell, F. C., 2016. Formative and Summative Assessment in the Classroom. *Theory Into Practice,* 55(2), p. Theory Into Practice.

Djambong, T. and Freiman, V., 2016. *Task-based assessment of students' computational thinking skills developed through visual programming or tangible coding environments.* Mannheim, CELDA, pp. 41-51.

Doleck, T., et al., 2017. Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education,* 4(4), pp. 355-369.

Feldhausen, R., Weese, J. L. and Bean, N. H., 2018. *Increasing Student Self-Efficacy in Computational Thinking via STEM Outreach Programs.* Baltimore, ACM.

Fronza, I., Ioini, N. and Corral, L., 2017. Teaching Computational Thinking Using Agile Software Engineering Methods: A Framework for Middle Schools. *ACM Transactions on Computing Education (TOCE),* 17(4), pp. 1-28.

Gamma, J., Helm, R., Johnson, R. and Vlissides, J., 1995. *Design patterns: elements of reusable object oriented software.* Boston: Addison-Wesley Longman Publishing Co.

Grgurina, N., et al., 2015. *Exploring students' Computational thinking skills in modeling and simulation projects: A pilot study.* London, ACM, pp. 65-68.

Grover, S. et al., 2017. Framework for Using Hypothesis-Driven Approaches to Support Data-Driven Learning Analytics in Measuring Computational Thinking in Block-Based Programming Environments. *ACM Transactions on Computing Education (TOCE),* 17(3), p. 14.

Grover, S., Cooper, S. and Pea, R., 2014. *Assessing Computational Learning in K-12.* Uppsala, ACM.

Haddaway, N. R., Collins, A. M., Coughlin, D. and Kirk, S., 2015. The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching. *PloS one,* 10(9), pp. 1-17.

Hoover, A. K. et al., 2016. *Assessing Computational Thinking in Students' Game Designs.* Austin, ACM.

Hubwieser, P. and Mühling, A., 2014. *Playing PISA with Bebras.* New York, ACM, pp. 128-129.

Jenson, J. and Droumeva, M., 2016. Exploring Media Literacy and Computational Thinking: A Game Maker Curriculum Study. *Electronic Journal of E-Learning,* 14(2), pp. 111-121.

Jiang, S. and Wong, G. K. W., 2017. *Assessing Primary School Students' Intrinsic Motivation of Computational Thinking.* Tai Po, IEEE, pp. 469-474.

Kalelioglu, F., 2015. A new way of teaching programming skills to K-12 students: Code.org.. *Computers in Human Behavior,* Volume 52, pp. 200-210.

Kalelioglu, F., Yasemin, G. and Kukul, V., 2016. A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing,* 4(3), pp. 583-596.

Koh, K.H., Basawapatna, A., Nickerson, H. and Repenning, A., 2014. *Real time assessment of computational thinking.* Melbourne, IEEE, pp. 49-52.

Kong, S.-C., 2016. A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education,* 3(4), pp. 277-394.

Lee, E. and Park, J., 2016. Challenges and Perspectives of CS Education for Enhancing ICT Literacy and Computational Thinking in Korea. *Indian Journal of Science and Technology,* 9(46), pp. 1-13.

Linn, M. C., 1985. The Cognitive Consequences of Programming Instruction in Classrooms. *Educational Researcher,* 14(5), pp. 14-16+25-29.

Lye S.Y. and Koh, J. H. L., 2014. Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior,* Volume 41, p. 51–61.

Moreno-León, J., Robles, G. and Román-González, M., 2016. *Comparing computational thinking development assessment scores with software complexity metrics.* Abu Dhabi, IEEE, pp. 1040-1045.

Moreno-León, J., Robles, G. and Román, M., 2015. Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED. Revista de Educación a Distancia,* Volume 46, pp. 1-23.

Moreno-León, J., Román-González, M., Harteveld, C. and Robles, G., 2017. *On the Automatic Assessment of Computational Thinking Skills: A Comparison with Human Experts.* Denver, Proceedings of the 2017 CHI Conference Extended, pp. 2788-2795.

Mühling, A., Ruf, A. and Hubwieser, P., 2015. *Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities.* London, ACM.

Nesiba, N., Pontelli, E. and Staley, T., 2015. *DISSECT: Exploring the relationship between computational thinking and English literature in K-12 curricula.* El Paso, IEEE.

NRC, 2010. *Report of a Workshop on the Scope and Nature of Computational Thinking,* USA: The National Academies Press.

NRC, 2011. *Report of a Workshop of Pedagogical Aspects of Computational Thinking,* USA: The National Academies Press.

Ouyang, Y., Hayden, K. L. and Remold, J., 2018. *Introducing Computational Thinking through Non-Programming Science Activities.* Baltimore, ACM.

Papert, S., 1980. *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Papert, S., 1991. *Situating constructionism. In I. Harel & S. Papert, Constructionism..* Norwood: Ablex.

Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M., 2008. *Systematic mapping studies in software engineering.* Swindon, BCS Learning & Development Ltd..

Rodrigues, R. S., Andrade, W. L. and Campos, L. M. S., 2016. *Can Computational Thinking help me? A* *quantitative study of its effects on education.* Erie, IEEE.

Rodriguez, B., Kennicutt, S., Rader, C. and Camp, T., 2017. *Assessing computational thinking in CS unplugged activities.* Seattle, SIGCSE '17.

Román-González, M., 2015. *Computational Thinking Test: Design Guidelines and Content Validation.* Barcelona, EDULEARN Proceedings.

Román-González, M., Pérez-González, Juan-Carlos and Jiménez-Fernández, C., 2017. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, Volume 72, pp. 678-691.

Rowe, E., Asbell-Clarke, J., Gasca, S. and Cunningham, K., 2017. *Assessing implicit computational thinking in zoombinis gameplay.* Hyannis, ACM.

Salomon, G. and Perkins, D. N., 1987. Transfer of cognitive skills from programming: When and how?. *Journal of Educational Computing Research,* Volume 3, pp. 149-170.

Seiter, L. and Foreman, B., 2013. *Modeling the learning progressions of computational thinking of primary grade students.* La Jolla, ACM.

Shute, V. J., Sun, C. and Asbell-Clarke, J., 2017. Demystifying computational thinking. *Educational Research Review,* Volume 22, pp. 142-158.

von Wangenheim, C.G., Alves, N.C., Rodrigues, P.E. and Hauck; J.C., 2017. Teaching Computing in a Multidisciplinary Way in Social Studies Classes in School – A Case Study. *International Journal of Computer Science Education in Schools,* 1(2), pp. 1-14.

von Wangenheim, C.G., et al., 2017. Teaching Physical Computing in Family Workshops. *ACM Inroads,* 8(1), pp. 48-51.

Webb, D. C., 2010. Troubleshooting assessment: an authentic problem solving activity for it education. *Procedia-Social and Behavioral Sciences,* Volume 9, pp. 903-907.

Weintrop, D., et al., 2014. *Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms.* Utrecht, Springer.

Werner, L., Denner, J., Campe, S. and Kawamoto, D. C., 2012. *The fairy performance assessment: measuring computational thinking in middle school.* Raleigh, ACM.

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

Witherspoon, E. et al., 2017. Developing Computational Thinking through a Virtual Robotics Programming Curriculum. *ACM Transactions on Computing Education (TOCE),* 18(1), pp. 1-20.

Wolz, U. et al., 2011. Computational Thinking and Expository Writing in the Middle School. *ACM Transactions on Computing Education,* 11(2), pp. 1-22.

Worrell, B., Brand, C. and Repenning, A., 2015. *Collaboration and Computational Thinking: A classroom structure.* Atlanta, IEEE, pp. 183-187.

Zhong, B., Wang, Q., Chen, J. and Li, Y., 2016. An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking. *Journal of Educational Computing Research,* 53(4), pp. 562-590.