DIGITAL TOOLS USED FOR THE DEVELOPMENT OF COMPUTATIONAL THINKING IN PRIMARY EDUCATION: A TEN YEAR SYSTEMATIC LITERATURE REVIEW

Maria Kordaki¹, Panagiotis Kakavas²

¹ Educational Technology in the Dept. of Cultural Technology and Communication of University of the Aegean (GREECE)
² Dept. of Cultural Technology and Communication of University of the Aegean (GREECE)

Abstract

This study presents a 10-year (2006-2016) systematic literature review related to the digital tools which have been used into K-6 education in order to develop Computational Thinking (CT) skills which are of great interest in the research community over the last years. In fact, many attempts have been conducted by various researchers for the development of CT skills in all educational levels using diverse ways. One of these ways is through the use of various digital tools which have been used in students of all ages and across different subjects. Thus, the aim of this paper is to review the CT-literature of the last decade related to primary education with the intention to: (a) identify the digital tools which have been used for the development of CT skills, (b) describe their characteristics and their potentials as educational tools in terms of CT-skill development, (c) demonstrate the ways these tools have been integrated in diverse educational contexts. The analysis of the data shows that for students’ CT skills’ cultivation: (a) twenty-six different digital tools have been used by primary school students, (b) most of these tools are graphical environments which use visual programming languages which were utilized to involve students in learning programming, construct games, develop simulations/models, and program robots, and (c) the most common educational use of these digital tools was for learning programming and in a less extend for game programming and game activities, for collaboration and pair programming, as well as for simulation, modeling and robotic education.

It is hoped that, the systematic review that is presented in this study would help other researchers and primary education level educators to devise new digital tools which could be used to develop students’ CT skills.

Keywords: Computational thinking, digital tools, primary education, systematic literature review.

1 INTRODUCTION

Computational Thinking has received a great interest over the last years as a fundamental set of skills, which promotes new ways of thinking to K-12 students across all scientific fields. Wing (2006) defines that CT is a term which involves “solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science” ([1]) (p. 33). She argued that CT consisted of a set of essential thinking skills such as abstraction, algorithmic thinking, data representation, logical thinking, problem decomposition, data analysis, recursion, debugging, simulation, automation and parallelization ([1], [2]).

After Wing’s call (Wing, 2006) for CT –skill acquisition by all literate population -not only by computer scientists- there were many initiatives and attempts to define the term CT and its core set of skills as well as to integrate these skills across K-12 education. Consequently, increase attention has been paid to help all students cultivate CT skills during their learning of various subjects, not only STEM. Nevertheless, there is still no consensus on what CT is and the best pedagogy for its promotion ([3]).

In fact, the International Society for Technology (ISTE) and the Computer Science Teachers Association realized an essential initiative which provided an operational definition of CT for K-12 with the following characteristics: formulating problems in a way that enables us to use a computer and other tools to help solve them, logically organizing and analysing data, representing data through abstractions such as models and simulations, automating solutions through algorithmic thinking, identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources, and generalizing and transferring this problem-solving process to a wide variety of problems ([4]).
Barr & Stephenson (2011) also focused on CT as an approach for solving problems in a way that can be implemented with a computer. However, they emphasized the students’ role as tool builders using a set of concepts to solve problems and recognize the multiple solutions of a problem ([5]). Furthermore, National Research Council ([6]) defined CT as the method, language, and systems of computer science to understand a variety of topics, ranging from creating computational models of scientific phenomena to create algorithms in order to plan one’s day more efficiently.

Many contemporary efforts have been also conducted to introduce children to CT concepts using various teaching approaches in both; formal educational settings and research programs. One of these ways is focused on the use of digital tools and environments that can easily be used by K-12 students across different subjects. In fact, it was concluded that, there is an impact on the ways CT can be integrated at K-12 education and the development of educational software and vice versa. To this end, many graphical programming environments (e.g. Scratch, Alice, Kodu) and digital tools (e.g. CTArcade, GrACE) have been used/developed by researchers all over the world.

Therefore, it would have a great interest to examine the various digital tools that have been used at each level of education and especially in the primary level—as it comprises the earlier ages of CT nurturing— and the ways they have been used in order to help students cultivate or assess various CT aspects. Thus, the aim of this study is to review the CT-literature of the last decade (2006-2016) related to K-6 education with the intention to: (a) identify the digital tools which have been used for the development of computational thinking skills, (b) describe their characteristics and their potentials as educational tools in terms of CT skills development, (c) demonstrate the ways these tools have been integrated in different educational contexts. This is the contribution of this paper.

Although we could not find studies with the same goals, some reviews with similarities have been reported. Heintz et al. (2016) reviewed how 10 different countries have approached introducing computer science into their K–12 education, finding that the studied countries either emphasize digital competencies together with programming or the broader subject of computer science or computing ([7]). Dasgupta and Purzer (2016) also conducted a systematic review of literature on pattern recognition to define pattern recognition as an aspect of CT ([8]). Moreover, Araujo et al. (2016) identified and classified approaches to promote CT and the different ways of assessing CT abilities ([9]). In addition, Moreno-Leon and Robles (2016) summarized the results of recent research using programming with Scratch in non Computing and communications related subjects, as well as studies analyzing the kind of skills students develop while learning to code in this environment ([10]). Lye and Koh (2014) also presented the current trends of empirical research in the development of CT in K-12 education through programming and suggested possible research and instructional implications ([11]). Despite the above, a review study focusing on the use of digital tools for the development of CT skills by primary level education students have not yet been reported.

It is hoped that, this systematic review would help other researchers to devise new digital tools which could be used to develop students’ CT skills, as well as researchers and primary education level educators to implement the existing tools in different frameworks even in disciplines not related to STEM fields. The rest of this paper is organized as follows: In Section 2, the research methodology is reported followed by the presentation of the results (Section 3) while in Section 4, these results are discussed and finally, in Section 5, conclusions and future research plans are drawn.

2 RESEARCH METHODOLOGY

2.1 Research Questions

This systematic review is guided by the following research questions:

a) What are the digital tools which have been used for the development of CT skills by primary school students?

b) Which are their main characteristics and their potentials as educational tools in terms of CT skills development?

c) In what ways these tools have been integrated in diverse educational contexts for CT development?
2.2 Data collection

2.2.1 Databases searched

This study reviews papers published in scientific journals, book chapters, and proceedings of international conferences for the last decade (2006-2016). To this end, ten electronic databases were searched, namely: SpringerLink, ACM, Bio-Medical Library, ERIC, IEEE Xplore Digital Library, Taylor & Francis Online, Wiley, LearnTechLib, Ingenta Connect and ScienceDirect.

2.2.2 Search term

The search was conducted using the keywords “computational thinking” and was limited to the period from January 2006 to December 2016. As a result, 1,873 papers were identified.

2.2.3 Selection of papers for inclusion in the review

The defined inclusion criteria applied in this systematic literature review demanded that the papers should: (a) explicitly refer the term “computational thinking” in title or abstract or keywords of each paper, (b) be written in English language, (c) use of digital tools for CT-skill development/assessment, (d) have a typical form of a scientific paper (not secondary studies or short papers), (e) focus on CT and primary level education students, and (f) present empirical data for K-6 educational level (less than 12 years old). In fact, studies providing weak information about the research methodology or had just some informal/preliminary observations -without data- after application of a teaching intervention or reported pilot studies, pilot tests and preliminary results were excluded. Papers that were reported in more than one database or were referred to the same research were kept only once. After excluding the papers that did not meet the inclusion criteria, 47 papers were found to be relevant to our inclusion criteria. All the reviewed papers are reported in a specific sub-section of the ‘References’ section.

2.2.4 Coding of papers

The 47 papers meeting the inclusion criteria were coded and categorized according to the research questions posed. The papers were individually analysed by the two researchers/authors and discussed in order to arrive at a mutual agreement whenever there was disagreement.

3 RESULTS

In this section, the results are presented in three Sections. In the first sub-section (Section, 3.1), the digital tools and the educational level they implemented is presented while in the next sub-section (Section, 3.2), the characteristics of these tools and their potentials as educational tools in terms of CT skills development are demonstrated. In the final sub-section (Section 3.3) the ways these tools have been integrated in different educational contexts for CT development are reported.

3.1 Digital tools for CT in K-6 education

The digital tools used in the conducted studies and the level of educational level they implemented are summarized in Table 1.

<table>
<thead>
<tr>
<th>Digital Tools</th>
<th>Educational Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch: 12</td>
<td>Primary: 26</td>
</tr>
<tr>
<td>Alice: 6</td>
<td>T-Maze: 1</td>
</tr>
<tr>
<td>AgentSheets: 4</td>
<td>FormulaT Racing (FTR): 1</td>
</tr>
<tr>
<td>Mindstorms software: 3</td>
<td>Categorizer: 1</td>
</tr>
<tr>
<td>Kodu: 2</td>
<td>Blockly: 1</td>
</tr>
<tr>
<td>ViMAP: 2</td>
<td>Not mentioned: 1</td>
</tr>
<tr>
<td>Logo microworld: 2</td>
<td></td>
</tr>
<tr>
<td>Simulation Creation Toolkit: 1</td>
<td></td>
</tr>
<tr>
<td>CT-IM: 1</td>
<td></td>
</tr>
<tr>
<td>CHERP: 1</td>
<td></td>
</tr>
<tr>
<td>ScratchJr: 1</td>
<td></td>
</tr>
<tr>
<td>TuneBlocks: 1</td>
<td></td>
</tr>
<tr>
<td>TurtleArt: 1</td>
<td></td>
</tr>
<tr>
<td>GrACE: 1</td>
<td></td>
</tr>
<tr>
<td>AgentCubes: 1</td>
<td></td>
</tr>
<tr>
<td>GameMaker: 1</td>
<td></td>
</tr>
<tr>
<td>CT-Arcade: 1</td>
<td></td>
</tr>
<tr>
<td>CTP Video-Prompt Survey: 1</td>
<td></td>
</tr>
<tr>
<td>ENGAGE: 1</td>
<td></td>
</tr>
<tr>
<td>VEnvi: 1</td>
<td></td>
</tr>
<tr>
<td>Software Lego Education</td>
<td></td>
</tr>
<tr>
<td>WeDo: 1</td>
<td></td>
</tr>
<tr>
<td>Ardublock: 1</td>
<td></td>
</tr>
<tr>
<td>T-Maze: 1</td>
<td></td>
</tr>
<tr>
<td>FormulaT Racing (FTR): 1</td>
<td></td>
</tr>
<tr>
<td>Categorizer: 1</td>
<td></td>
</tr>
<tr>
<td>Blockly: 1</td>
<td></td>
</tr>
<tr>
<td>Not mentioned: 1</td>
<td></td>
</tr>
</tbody>
</table>

As it is shown in Table 1, from the 26 different digital tools the Scratch environment was mostly used (12 papers). To a lesser extent the following tools were used: Alice (6 papers), AgentSheets (4
papers), Mindstorm software (3 papers), Kodu (2 papers), Vimap (2 papers) and Logo microworlds (2 papers) while the rest digital tools had been used once. Almost half of the studies (26 studies) have been conducted in grades 1-6 (primary school), while the majority of the remaining studies carried out in both; primary and secondary education. Finally, only one paper reported the development of CT in Kindergarten students.

Table 2 depicts the codes of the reviewed papers (columns 1, 4, 7) along with the digital tools used (columns 2, 5, 8), and the educational level (P=Primary, M= Middle school, H= High school) they used (columns 3, 6, 9).

3.2 Characteristics of the digital tools’ use for CT development in K-6 education

The digital tools used in primary education in terms of CT development fall in the following categories: (a) Visual Programming environments, (b) Modeling-Object-oriented programming environments, (c) Game construction tools, (d) Robotics, (e) Computer-based musical tools,(f) No programming digital tools. The tools that fall in each category and their characteristics are briefly presented below.

- **Visual Programming environments**: The following tools were used: Scratch- is considered the most suitable visual programming language to develop CT skills through programming in K-12 education ([43]). ScratchJr- is also a visual programming language for young children to help them learn to solve problems, design projects, and express themselves creatively on the computer. TurtleArt- that also uses an interface close to the Scratch interface and it has designed to provide children with introductory computer experiences in programming and animation ([25]). Alice- which is an interactive graphics programming environment allowing students to control characters in a 3D environment which is a useful tool for teaching and learning CT, problem solving, and computer programming across a spectrum of ages and grade levels. Ardublock- it is a visual programming environment which is a plug-in that is added to the IDE of Arduino that allows the user to program in blocks of functions. CHERP- it is a hybrid
tangible and graphical computer language that has been designed to provide an engaging introduction to computer programming for young children. Finally, Blockly – that is also a visual programming environment that facilitate the cultivation of CT skills through programming which can also export blocks to many languages, such as JavaScript, Python, PHP etc.

b) **Modeling object-oriented programming environments.** Here the following platforms were used: ViMAP - that is an agent-based visual programming language and a modeling platform that provides a drag-and-drop interface for constructing programs ([45]). CTSim - which is an agent-based modeling environment, especially in the field of physics, where a user can control an individual computational object ([13]). Logo microworlds - which are proposed ([24], [46]) as adequate programming languages for beginners; particularly for primary school students which can cultivate CT skills through programming.

c) **Game construction tools.** The tools which were used include: Kodu – it allows children to create games via a simple visual programming language and can be used to teach creativity, problem solving, storytelling, as well as programming. T-Maze - that is a tangible programming tool for children of younger ages (5-9) to build programs to play multilevel maze-escape games and create their own mazes, by connecting wooden blocks to control a virtual avatar in a grid world on the screen to escape from a maze ([49]). Grace - it is a puzzle game with a vegetable-collecting narrative designed to encourage algorithmic thinking through puzzles analogous to finding the minimum spanning tree (MST) of a graph ([27]). GameMaker - it is designed to allow the users to easily develop video-games ([29]). CTArcade - it is a web-based educational gaming environment designed to help students improve and internalize CT skills while converting their natural game-play into a formal strategy ([32]). Engage - that is an immersive game-based learning environment designed to expose students to problems that encourage the development of CT ([35]). VEnvl - it is a programming software that allows students learn CT concepts through the process of choreographing movement of a virtual character ([28]). AgentSheets (2D) and AgentCubes (3D) - that are object-based programming environments which are tailor for middle-school students to develop computer games and simulations, as well as Simulation Creation Toolkit - that is integrated into AgentCubes software and constitutes an initial attempt at the idea of using high-level CT Patterns directly ([12]).

d) **Robotics.** The robotic toolkits were used are: Lego Mindstorms kits (Ev3 or NXT-G) and Lego WeDO kits - which provide several parts (e.g. brick-controller, motor, sensors etc.) and a visual programming software for building and programming the various robots that can be constructed ([16]).

e) **Computer-based musical tools.** Here, the TuneBlocks was used – that is one of Impromptu’s five “PlayRooms”, a computer-based musical programming environment that allows students to compose melodies using their musical intuitions by arranging small blocks that represent musical notes ([22]).

f) **No programming digital tools.** The following no programming platforms were used: FormulaT Racing (FTR) - it is a racing game where players direct a racecar by painting the track with different colours; so that each colour indicate a different velocity ([52]). CTP Video-Prompt Survey - which is an on-line assessment tool that utilizes video-based prompts to evaluate the degree of transfer of understanding of CTPs to real-world situations ([34]). Categorizer – it is an interactive gallery that allows members of a learning community to contribute and classify computational artifacts ([54]).

3.3 Ways of digital tool-integration in teaching practices for CT development

The diverse ways that digital tools have been integrated in teaching practices for CT-development fall in the following categories: (a) CT development and assessment via programming activities, (b) Simulations and modeling activities, (c) Game programming – game activities, (d) Collaboration and pair programming, (e) Robotics, (f) No computer science education, (g) non programming CT’s evaluation. The tools that fall in each category and their ways of integration in various teaching practices are shortly presented below.

a) **CT development and assessment via programming activities.** The diverse digital tools and the ways they have been used include: Scratch - for the development and assessment of students’ CT abilities via programming activities ([20], [42], [43], [44]). Alice - for teaching programming ([18]), and for students’ assessment of CT development through semi-finished programming
tasks ([57]). LOGO microworlds- for teaching programming via programming tasks ([46], [24]).
 Ardublock- for programming an Arduino ([41]). Scratch- for students’ learning of computer science concepts and attitudes towards computing ([37]), and for school-girls’ engagement in computing activities ([51]). Moreover, Scratch has been used to examine several aspects of learning such as self-efficacy, self-interest, and self-assessment of CT development ([30]), as well as students’ algorithm constructs and how well they transfer this learning from Scratch to text-based languages ([23]). Kodu- for teaching programming moving between diverse frameworks such as Kodu, Alice and Mindstorms ([48]) and to introduce reasoning about lawful behavior as an important CT-skill ([47]).

VEnvI- to facilitate engagement, excitement and interest of students in CT and programming via edutainment ([40]). ViMAP as well TuneBlocks tool were used for learning computational science as an aesthetic experience ([22]). Blockly, finally, has been used as a graphical input in examining how different input and output methods in coding affected the problem-solving process and class dynamics ([58]).

b) Simulations – modelling activities. Here, the tools and the ways they were used are as follows: Scratch- it has been used for children to make simulations ([26]). Simulation Creation Toolkit- it was used for simulation creation while enabling higher-level implementation of agent behaviors ([12]). CTsIM for simulation and modeling activities in the science domain ([13]). AgentSheets has been also used for game design and simulations [31], [50].

c) Game programming - game activities. The tools and the ways they were used are: Alice- for programming games as a strategy for measuring students’ computational learning ([53]). GameMaker- for game-construction by the students ([29]). CTArcade- for teaching CT-skills without the use of programming, but by building on children’s existing game-play interest and skills ([32]). FormulaT Racing- for a video-game construction by the students, to facilitate the relationship between velocity and acceleration ([52]). T-Maze for building computer programs in maze games by manipulating wooden blocks ([49]). AgentCubes ([28]) and AgentSheets for game design ([33]).

d) Collaboration and pair programming. For this aim, the digital tools have been used as follows: Alice, to measure the influence and effectiveness of collaboration and pair programming in diverse programming activities (e.g. gender, partnership) ([19], [56]), ViMAP- to help students collaboratively invent “mathematical machines” for generating geometric shapes ([45]), ScratchJr- in order pairs of students be able to create a range of basic shapes and letters via programming ([21]). TurtleArt- to investigate how students learn, by mapping their ideas in a collaborative programming environment ([25]). AgentSheets to allow students master and retain CT patterns and applying them in formal summative assessments through collaboration ([55]).

e) Robotics. For robotics’ integration in diverse teaching practices the tools and the ways they have been used are: Mindstorms software and WeDo software in order students to design, construct and program their robots ([16], [33], [40]). CHERP programming language with Lego robotic kits were also used in order students to learn core concepts of robotics and programming ([14]).

f) No computer science education. The tools and the ways they have been used are: Scratch, to integrate CT concepts into a science curriculum ([39]), to improve students’ mathematical capabilities via programming ([15]), as well as in English classes to test students’ learning outcome using programming tasks ([36]). Categorizer, also, has been used in order students to construct and categorize fractals in Mathematics ([54]).

g) Non programming CT’s evaluation. The tools and the ways they have been used for CT-skill assessment without the use of programming include: CTP Video-Prompt Survey that has been used for evaluating students’ understanding and application of CT patterns to novel situations ([34]), EngAGE for students’ stealth assessment while they were playing with it ([35]), as well as GRACE for students’ strategy analysis of an educational game ([27]).

4 DISCUSSION AND CONCLUDING REMARKS

This study is a ten-year review (2006-2016) on the digital tools used for CT’s development in primary education. It was based on forty-seven (47) papers, found relevant for inclusion, after searching ten (10) databases using specific keywords and a total of 1,873 papers. This study attempts to answer the research questions posed by trying to: (a) identify the digital tools which have been used for the
development of CT skills, (b) describe their characteristics and their potentials as educational tools, and (c) demonstrate the ways these tools have been integrated in diverse educational contexts.

**Digital tools used for CT skills' development:** The analysis of the data has shown that from the twenty-six different digital tools have been used in K-6 education. Scratch is the software used more in diverse teaching practices for students’ cultivation of CT skills. To a lesser extent were used Alice, AgentSheets, Mindstorms’ software, Kodu, Vimap and Logo microworlds. The rest digital tools have been used once. They constitute alternative software, most of which are programming environments that have been used in various studies. It is also important to be mentioned that the majority of the studies have been conducted to older students of the elementary school, while several studies have been carried out in both; primary and secondary education. Finally, only one paper reported the development of CT in Kindergarten students with the use of a digital tool.

**Characteristics of the digital tools:** The analysis of the data revealed, that, most of the digital tools -that have been used to develop students’ CT skills- use graphical interfaces utilizing visual programming in various activities. Most of these tools were used for programming activities, creating simulations and models, constructing various educational games, as well as programming various robots which were constructed by the students. In a less extend computer-based musical tools and non programming digital tools were used.

**Ways of the digital tools integration in teaching practices:** The analysis of the data shows that there were diverse ways that digital tools have been integrated in various teaching practices for CT development. The most common constitute the use of various programming activities with the intent to teach programming to elementary students aiming to develop or assess their CT abilities. For this purpose, a number of studies also reported that diverse digital tools have been used for game programming and game activities, for collaboration and pair programming between students, as well as for simulations/modeling tasks and robotic education. On the contrary, a small number of studies reported studies attempting to cultivate CT skills via programming in a different framework/curriculum to those of computing and computer science education, such as mathematics or English language. Finally, few studies reported the use of no programming digital tools for students’ CT evaluation.

5 CONCLUSIONS AND FUTURE WORK

This paper has presented a decade (2006-2016) review study focusing on the digital tools that have been used for the development of CT skills in primary education. The findings revealed that for the development of CT skills by primary level education students: (a) twenty-six different digital tools have been used while most of the review studies are referred to students attending upper elementary school classes, (b) most of these tools are graphical environments which use visual programming languages and implemented in teaching practices in order to enable students to program, construct games, develop simulations/models, and program robots, and (c) the most common educational use of these digital tools was to teach programming and in a less extend for game programming and game activities, for collaboration and pair programming between students, as well as for simulation, modeling and robotic education. However, the following issues related to the use of digital tools remain still under-researched: (a) the difference in efficiency -in terms of student CT development- of the tools used, (b) the use of other available digital tools in students’ CT cultivation, (c) the use of existing digital tools that do not utilize a programming language and could contribute to the CT development by the students, (e) the use of digital tools in lower primary school and in Kindergarten, as well as (f) the use of various digital tools in non-STEM subjects. These issues could form future research directions.

The current review has a number of limitations, since it was limited by the search terms used, the scientific databases searched and the period in which the papers were published. However, the papers discussed in this literature review provide a snapshot of research on the digital tools used in K-6 education for students’ CT development, which is representative of the state of the art at the time.

REFERENCES


**Reviewed papers**


