OBSERVABLE COMPUTATIONAL THINKING SKILLS IN PRIMARY SCHOOL CHILDREN: HOW AND WHEN TEACHERS CAN DISCERN ABSTRACTION, DECOMPOSITION AND USE OF ALGORITHMS

M. Moschella

Faculty of Education, Free University of Bolzano (ITALY)

Abstract

This article aims to explore the elements which compose the Computational Thinking skills: algorithmic thinking, abstraction and decomposition, and their correlation with primary school learners and with the adopted tools. In order to answer the research questions, a literature review was pursued. Forty-eight contributions were selected, analysed and thematised. Findings show that many valid tools exist and could be used to observe and improve pupils' abilities in this field and that pupils can use abstract and algorithmic thinking to solve problems. Practical guidelines are needed for implementing this 21st-century skill with a cross-curricular approach, which could include all subjects and all children.

Keywords: Computational Thinking, primary school, cognitive processes.

1 INTRODUCTION

Computational Thinking is deemed one of the fundamental skills of the 21st century. It is defined as a set of cognitive skills which involves problem-solving, abstraction, using algorithms, restructuring processes, reformulating problems, implementing solutions. Seymour Papert hypothesised this way of thinking when reflecting on the relationship between computers and children [1]. Then, in the last decade, Wing [2, p.33] proposed that CT entails "solving problems, designing systems, and understanding human behaviour, by drawing on the concept, fundamental to computer science". While specifying this concept, considerable attention was gained by the educational environment, because of the primary role for the education to active citizenship and the conduction to the labour market. The debate on several aspects of CT is still under the magnifying glass and, in the last ten years, the number of publications has been growing exponentially [3][4][5][6]. Despite a growing effort to implement computational thinking (CT) skills in primary schools, only a few studies reported which CT skills could be learned at a certain age.

Generally, computer science and programming courses are not included in the study plan of university courses in education, but, once at school, teachers should deal with this subject. For example, they are expected to plan cross-curricular computer science activities, starting from the upper primary school classes, or even earlier. They are also expected to diversify the activities, to consider the individual differences and the special education needs. In many countries, a compulsory computer science (CS) curriculum has been introduced for years [5], and if the theoretical issues are much investigated, empirical research is lacking, above all related to the primary school.

This article aims to develop a framework for the teacher, who are planning computer science activities and want to teach it effectively. Many aspects have been investigated until now, like assessment criteria, curriculum features, age-related evaluation parameters and teacher training problems. However, among them, three aspects related to the implementation of CT at the primary school still seem to be not enough considered: abstraction, algorithmic thinking and decomposition. Therefore, the focus of the present paper is on these three aspects that are deemed as fundamental at that age [6].

The paper is structured as follows. Section 2 presents the definition for each one of the three abilities and related literature; then, the correlation with developmental psychology is outlined, in order to motivate the research questions. In section 3, the methodology is presented, followed by the results in section 4. Finally, in section 5 conclusions and recommendations are suggested.

2 THEORETICAL BACKGROUND

According to the constructivism hypothesis, children could achieve a different level of abilities at a different stage during their growth. Early primary school children are generally considered to be between the preoperational stage and the concrete operational stage, while their thoughts move from being symbolic to being concrete, and finally, abstract (in the last formal operational stage) [7]. With his studies, Seymour Papert developed the constructionism approach, when theorising the need for learners to construct an object-to-think-with and to experience situated learning. According to Piaget's writings, a child is more likely to be able to manage abstract thoughts after the age of ten. At the same time, Papert developed an approach based on abstracting and algorithmic thinking for younger children. Successively in [8] and [9] the authors differentiated the elements of CT in Abstraction, Generalization, Decomposition, Algorithms (Sequencing, Flow of control), Debugging. In order to develop a K-6 curriculum, they then described each skill, from the age of six to eight and from grade K-2 to K-6. In addition to [1] and [2], other pivotal works focused on primary school teacher training [10] [11], on the effect of programming on children's motivation [12] and on the cognitive transfer effect [13], on international assessment program (Bebras) [14] [15] and on the cross-curricular implementation challenge [16] [17].

Of the five basic elements mentioned above three were taken into consideration and they could be defined as follows:

a) Algorithmic thinking could be defined as the use of a set of rules that precisely define a sequence of operations [18]. Other formal definitions of this concept are accepted and used by the scientific community, mostly when specifying different types of algorithms (sorting algorithms, procedural algorithms).
b) The definition of abstraction is not simple, because this process entails to many fields: art and music, language, mathematics. It is hard to find a standard definition that could be exhaustive. Kramer outlined two related aspects of abstraction in the Informatics field. The first it is suggested that it could be noticed when withdrawing or removing something or ignoring one or more properties of a complex object to follow others. Moreover, it could also involve the process of formulating general concepts by abstracting common features of examples or instances [19].
c) Decomposition is defined as the ability to divide a problem, too big or too complicated, in many subproblems, so that they could be solves effectively [20]. Decomposition and use of algorithms are partially related because, for instance, an effective step-by-step solution for subproblems could be an algorithm itself.

This article outlines the state of the empirical research of these three essential elements of CT (algorithmic thinking abstraction, decomposition) through a systematic review of the literature relevant to the primary school, regarding the aspects of age of first contact, tools and assessed/improved ability.

3 METHODOLOGY

3.1 Research questions

The research questions aim to explore three aspects, namely, whether abstraction, decomposition and using algorithms can be observed in primary school students (RQ1), at what age these cognitive skills could be observed (RQ2), and which tools can be used to develop these skills (RQ3).

3.2 Procedure

In order to investigate didactic and pedagogical aspects of abstraction, algorithmic thinking and decomposition, and to outline a practical framework for planning activities and developing CT skills, a systematic literature review was performed.

This method is used to explore a research area by classifying and counting the contributions based on an adopted taxonomy. Articles and book chapters were found and categorised, using Scopus and ProQuest as databases. Scopus was used as academic search engine three times, with the following search strings:

a. "computational thinking" AND abstraction AND ("elementary school" OR "primary school")

b. "computational thinking" AND "algorithm" AND ("elementary school" OR "primary school")

c. "computational thinking" AND decomposition AND ("elementary school" OR "primary school")

The database returned for (a) 55 results; for (b) 84 results and for (c) 12 results. Any other filter was not applied.

ProQuest was also investigated, but using the following strings:

d. Abstraction AND "CT"

e. Algorithm AND "CT"

f. Decomposition AND "CT".

The database returned in this case for (d) 109 results, for (e) 159 results and for (f) 44 results. The two databases were last accessed on the 12th of January 2019. Filters were applied for language (English) and peer-reviewed sources. A first screening of the papers was made to evaluate the relevance of this work, and 99 works were found out. The following inclusion criteria (IC) were then used to sort them:

IC1: papers and contribution related to primary school

IC2: Relevant papers for the topic

Exclusion criteria (EC) were also used to screen the papers after the first reading:

EC1: duplicate

EC2: papers focused on teacher training CT

EC3: systematic literature reviews or non-empirical studies

Table 1: Remaining Studies by Selection Criterion

Criterion applied	Number of studies
Returned by search engines	463
IC1 – Papers related to primary school	252
IC2- Relevant to the specific topic	189
EC1- Non-duplicate studies	126
EC2- Focused on students' skill	93
EC3 – Empirical studies	48

As a result of the screening, 48 studies were selected to be analysed for the present paper.

By the 48 studies, the three research questions were analysed. The following classification scheme guided the analysis: abilities and skills that were improved/evaluated/enhanced during the study; the age of students/grade level of the primary or elementary school; a classification of the most common and useful tools.

All 48 studies were analysed and classified by the author. Besides, to answer the research questions defined previously, data about the number of children involved and the countries where the study took place were extracted. The number of participants varied considerably, varying from 2 to 1600. The studies were mostly applied only in primary school (27 studies), but middle school was considered as well (10 studies), when: a) the topic and the research was present in both, and b) the activities might be interchangeable or involved longitudinal groups. Eleven studies involved pre-school/kindergarten children and early grade of primary school pupils. In one case, the school grade was not given, but the study was designed for students with twice-exceptional students. USA, Italy and UK are the most represented countries.

4 RESULTS AND DISCUSSION

As aforementioned, algorithmic thinking, abstraction and decomposition are the three elements we took into consideration to formulate the research questions. Studies were thematically classified in Table 2.

Table 2: Thematic classification of contributions by school level

	Kindergarten and early primary	Primary school (only)	Primary and middle/secondary school
Algorithmic thinking and sub-elements	[21] [22] [23]	[24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40]	[25] [26] [28] [31] [34] [38]
Abstraction and sub-elements	[41] [42] [43] [44] [45] [46] [47] [48]	[38] [39] [41] [42] [43] [47] [48] [49] [50] [51] [52[53[54[55[56] [57] [58] [59] [60] [61] [62] [63] [64] [65]	[43] [54] [14] [44] [55] [59]
Decomposition	[42] [43]	[49] [50] [56]	[31] [38]

Articles based exclusively on decomposition were not found because this element was always related to algorithmic thinking or abstraction.

In Table 3, then, an overview of the most commonly used tools concerning algorithmic thinking is presented. In kindergarten and early primary levels, no study reported data on the algorithmic thinking. Consequently, no tools were found to improve algorithmic thinking for this school grade. In the second column, the instruments used at primary school are listed, being numerous and different from each other; the third column, contains the tools that could be suitable even for older learners, as reported in the articles.

Table 3. Tool classified by level and related to algorithmic thinking.

	Kindergarten + Early Primary school	Primary school	Primary school + Middle school
Algorithmic thinking	/	Run Marco (1) Game Based Learning (1) Code Baymax - Kodable -Code studio- Lego Bits and Bricks- Lightbot (1) Turstle Graphic Tutorial System (TGTS) (1) Metaphors (1) Analytic Hierarchy Process (1) Tangibles for graph (2) CS unplugged activities (6) Bebras tasks (Little Beavers - 6-10 years) (2)	Metaphors Tangibles for graph Puzzle based game Nimrod (1) Beebot (1) Question – Problems (2)

		Weather Forecast Games (1) Scratch (10) and Phyton Puzzle based game (2) Xlogo4schools (1) SOLO taxonomy (1) TurtleArt (Drawing Shapes)-LaPlaya (Digital Story) (1) Nimrod (1) Beebot (1) Mega Math (1)	

Table 4. Tool classified by the school level related to abstraction

	Kindergarten + Early Primary school	Primary school	Primary school + Middle school
Abstraction	Lego Music Notation (1) Creative Hybrid Environment for Robotic Programming (2) Lego WeDo (1) Questions (2) MecWilly Robot (1) iPad (1) - pencil paper (1) ScratchJr (1) CTRL_SPACE (1) Embodied activities (1)	Barefoot Computing project activities (1) Pirate Plunder (1) Kodu Game Lab (4) Physical manipulatives- tiles and flashcards (1) LÜK (1) Tangram (1) Unplugged mathematical tools (1) Bebras (2) CodeSpells (1) Scratch (10) Logo (2) Phyton (2) Minecraft (1) MIT App inventor (2) Alice (1)	Scratch Bebras tasks Barefoot Computing project activities Kodu Game Lab

Table 5. Tool classified by school level related to decomposition.

	Kindergarten + Early Primary school	Primary school	Primary school + middle school
Decomposition	iPad – pencil paper activities	CS Unplugged Unplugged mathematic tool	Scratch Nimrod CS unplugged

		Tangram Barefoot Computing project activities	Beebot

The results obtained from the systematic literature review are summarised in tables 3 to 5 and show the relationship between the three essential elements of CT, school grade and tools. In this literature, 44 different programming tools have been listed, and most of them are easy to use and do not require preparation course. Various tools are based on visual programming languages, aiming at promoting programming skills (Barefoot Computing project activities, Pirate Plunder, Kodu Game Lab, Physical manipulatives- tiles and flashcards, LÜK, Tangram, Unplugged mathematical tools, Bebras, App Inventor, CodeSpells, Scratch, Logo, Phyton, Minecraft, MIT App Inventor, Alice). Although adopted to develop computing skills, other tools have as their first aim to improve mathematical and logic skills (Physical manipulatives- tiles and flashcards, LÜK, Tangram, Unplugged mathematical tools). These results suggest that it is possible for primary school teachers to enhance and develop algorithmic thinking, abstraction and decomposition that go along with CT. Together these findings provide more insights into the possibility to introduce CS activities, not only with a computer or tablet but also with paper and pencil, flashcards, tiles and embodied activities. One of the advantages of adopting unplugged activities is that also young learners could experience them avoiding the exposition to the screen. Another strength is that such projects could be implemented in schools that do not have a tablet or a computer for each student. Besides, some studies were longitudinal and involved students of different school levels, demonstrating that such projects could be executed also in special schools with a different organization (Progressive education, Montessori school). According to the data reported in Table 3, currently there are no data on using and promoting algorithmic thinking in kindergarten and early primary school, even if children at that age start to follow and apply procedures and sequence in the diverse contexts of their life. Further research should be undertaken to investigate these characteristics and other elements (Sequencing, Debugging) related to Computational Thinking in young learners.

5 CONCLUSIONS

After reviewing over 48 studies on CT it emerged that the systematic adoption of new approaches and tools create an active and enjoyable learning experience. Although the study was limited to only two databases, it was possible to identify trends and find out which aspects should be explored in order to suggest tools for teachers who have to teach computer science from primary school. Findings showed that also unplugged activities could be effective in improving different abilities, even with pre-literate children. Indeed, the variety of activities, tools and methods can represent an issue, if educators and teachers do not have a practical framework for each school grade. This would be a fruitful area for a further literature review considering the other two elements composing CT, like debugging and generalization in K-6 education [12], in order to outline broader and comprehensive guidelines for each ability. More widely, research is also needed to determine long- term effect on the learning process and on other cognitive processes, like attention, memory and visuo-spatial abilities. A couple of study reported also data about pair programming or cooperative learning: investigations on the correlation with computing and Computational Thinking and social skills are also needed.

In conclusion, the present study has the merit to demonstrate that the implementation of computational thinking is possible since the early childhood with specific tools and activities.

REFERENCES

[1] S. Papert, '"*Mindstorms: children, computers, and powerful ideas*,", ed. 2., 1993.

[2] J M.Wing, Computational thinking. *Communications of the ACM,* vol.*49*, n.3, 33-35, 2006.

[3] K. Rich, C. Strickland and D. Franklin, '"A literature review through the lens of computer science learning goals theorized and explored in research," *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, pp. 495-500, 2017.

[4] De Araujo, W.L. Andrade and D.D. Serey Guerrero, '"A systematic mapping study on assessing computational thinking abilities," *Proceedings - Frontiers in Education Conference, FIE*, vol.2016, November 2016.

[5] V. Garneli, M.N. Giannakos and K. Chorianopoulos, '"Computing education in K-12 schools: A review of the literature," *IEEE Global Engineering Education Conference, EDUCON*, vol. 2015-April, pp. 543-551, 2015.

[6] J. Voogt, P. Fisser, J. Good, P. Mishra and A. Yadav, '"Computational thinking in compulsory education: Towards an agenda for research and practice," *Education and Information Technologies*, vol. 20, no. 4, Dec 2015, pp. 715-728, 2015.

[7] J. Piaget and P. Cook, *The origin of intelligence in the child*. (Translated by Margaret Cook.). Routledge & Kegan Paul, 1953.

[8] A. Fluck, M. Webb, M. Cox, C. Angeli, J. Malyn-Smith, J. Voogt and J. Zagami, "Arguing for computer science in the school curriculum," *Educational Technology and Society*, vol. 19, pp. 38-46, 2016.

[9] C. Angeli, J. Voogt, A. Fluck, M. Webb, M. Cox, J. Malyn-Smith and J. Zagami, '"A K-6 Computational Thinking curriculum framework: Implications for teacher knowledge," *Educational Technology & Society*, vol. 19, no. 3, pp. 47-57, 2016.

[10] P.J. Rich, S.F. Browning, M. Perkins, T. Shoop, E. Yoshikawa and O.M. Belikov, '"Coding in K-8: International trends in teaching elementary/primary computing," *Tech Trends*, Jun 2018, pp. 1-19, 2018.

[11] E.M. Saari, P. Blanchfield and G. Hopkins, '"Learning computational thinking through the use of flash action scripts: Preparing trainee elementary school teachers for teaching computer programming," *Proceedings CSEDU 2015 - 7th International Conference on Computer Supported Education*, vol. 2, pp. 75-84, 2015.

[12] Y. Tran, '"Computer programming effects in elementary: perceptions and career aspirations in STEM," *Technology, Knowledge and Learning*, vol. 23, no. 2, pp. 273-299, 2018.

[13] R. Scherer, F. Siddiq and B. Sánchez Viveros, '"The cognitive benefits of learning computer programming: A meta-analysis of transfer effects," *Journal of Educational Psychology.*, Oct 25, 2018.

[14] C. Izu, C. Mirolo, A. Settle, L. Mannila and G. Stupuriene, '"Exploring Bebras tasks content and performance: A multinational study," *Informatics in Education*, vol. 16, no. 1, pp. 39-59, 2017.

[15] V. Dagiene and G. Stupuriene, '"Bebras - A custainable ommunity building model for the Concept Based Learning of informatics and computational thinking," *Informatics in Education*, vol. 15, no. 1, pp. 25-44, 2016.

[16] D. Lim, J.-. Lee, D. Moon and G. Um, '"Prioritizing learning topics of coding curriculum for elementary students using the analytic hierarchy process," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 3, pp. 46-50, 2018.

[17] J. Sáez-López, M. Román-González and E. Vázquez-Cano, "Visual programming languages integrated across the curriculum in elementary school: A two-year case study using "Scratch" in five schools," *Computers and Education*., vol. 97, pp. 129-141, 2016.

[18] J. Mezak and P. Pejic Papak, '"Learning scenarios and encouraging algorithmic thinking," 2018, *Proceedings 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018*, pp. 760-765, 2018

[19] J. Kramer, '"Is abstraction the key to computing?" *Communications of the ACM*, vol. 50, no. 4, pp. 36-42, 2007.

[20] W.J. Rijke, L. Bollen, T.H.S. Eysink and J.L.J. Tolboom, '"Computational thinking in primary school: An examination of abstraction and decomposition in different age groups," *Informatics in Education*, vol. 17, no. 1, pp. 77-92, 2018.

[21] L.P. Flannery and M.U. Bers, "Let's Dance the "Robot Hokey-Pokey!": Children's programming approaches and achievement throughout early cognitive development," *Journal of Research on Technology in Education*, vol. 46, pp. 81-101, Fall 2013. 2013

[22] D. Messer, L. Thomas, A. Holliman and N. Kucirkova, "Evaluating the effectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory," *Education and Information Technologies*, vol. 23, pp. 2879-2888, Nov 2018. 2018.

[23] D.J. Portelance, A.L. Strawhacker and M.U. Bers, "Constructing the ScratchJr programming language in the early childhood classroom," *International Journal of Technology and Design Education*, vol. 26, pp. 489-504, Nov 2016.

[24] K.M. Rich, C. Strickland, T. Andrew Binkowski, C. Moran and D. Franklin, "John henry AWARD k–8 learning trajectories derived from research literature: Sequence, repetition, conditionals," *ACM Inroads*, vol. 9, pp. 46-55, 2018.

[25] D. Pérez-Marín, R. Hijón-Neira and M. Martín-Lope, "A methodology proposal based on metaphors to teach programming to children," *Revista Iberoamericana De Tecnologias Del Aprendizaje*, vol. 13, pp. 46-53, 2018.

[26] A. Bonani, V. Del Fatto, G. Dodero, R. Gennari and G. Raimato, "Participatory design of tangibles for graphs: A small-scale field study with children, in" *Smart Innovation, Systems and Technologies*, vol. 80, pp. 161-168, 2018.

[27] T. Bell and J. Vahrenhold, "CS unplugged—How is it used, and does it work?" *Lecture Notes in Computer Science* vol. 11011 LNCS, pp. 497-521, 2018.

[28] L. Budinská and K. Mayerová, "Graph tasks in bebras contest - What does it have to do with gender?" in *ACM International Conference Proceeding Series*, pp. 83-90, 2017.

[29] H. Gürbüz, B. Evlioğlu, Ç.S. Erol, H. Gülseçen and S. Gülseçen, "“What's the Weather Like Today?”: A computer game to develop algorithmic thinking and problem-solving skills of primary school pupils," *Education and Information Technologies*, vol. 22, pp. 1133-1147, 2017.

[30] P. Mozelius and L.-. Öberg, "Play-based learning for programming education in primary school: The Östersund model," *in Proceedings of the European Conference on e-Learning, ECEL*, pp. 375-383, 2017.

[31] D. Isayama, M. Ishiyama, R. Relator and K. Yamazaki, "Computer science education for primary and lower secondary school students: Teaching the concept of automata," *ACM Transactions on Computing Education*, vol. 17, 2016.

[32] J. Hromkovič, T. Kohn, D. Komm and G. Serafini, "Combining the power of python with the simplicity of logo for a sustainable computer science education," *Lecture Notes in Computer Science*), vol. 9973 LNCS, pp. 155-166, 2016.

[33] L. Seiter, "Using SOLO to classify the programming responses of primary grade students," in SIGCSE 2015 - *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pp. 540-545, 2015.

[34] D.B. Harlow, H. Dwyer, A.K. Hansen, C. Hill, A. Iveland, A.E. Leak and D.M. Franklin, "Computer programming in elementary and middle school: Connections across content," in *Improving K-12 STEM Education Outcomes through Technological Integration*, pp. 337-361, 2015.

[35] H. Dwyerz, C. Hilly, S. Carpenterz, D. Harlowz and D. Frankliny, "Identifying elementary students' pre-instructional ability to develop algorithms and step-by-step instructions," in *SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, pp. 511-516, 2014.

[36] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson and A. Settle, "Computational thinking in K-9 education," in *ITiCSE-WGR 2014 - Working Group Reports of the 2014 Innovation and Technology in Computer Science Education Conference*, pp. 1-29, 2014.

[37] T. Bell, F. Rosamond and N. Casey, "Computer science unplugged and related projects in math and computer science popularization," *Lecture Notes in Computer Science*, vol. 7370, pp. 398-456, 2012.

[38] L. Benton, I. Kalas, P. Saunders, C. Hoyles and R. Noss, "Beyond jam sandwiches and cups of tea: An exploration of primary pupils' algorithm-evaluation strategies," Journal of Computer Assisted Learning., vol. 34, pp. 590-601, 2018.

[39] H. Ehsan, T.M. Dandridge, I.H. Yeter and M.E. Cardella, "K-2 students' computational thinking engagement in formal and informal learning settings: A case study (fundamental)," *Proceedings of ASEE Annual Conference and Exposition, Conference,* 2018.

[40] C., Hsu and T.-. Wang, "Applying game mechanics and student-generated questions to an online puzzle-based game learning system to promote algorithmic thinking skills," *Computers and Education*, vol. 121, pp. 73-88, 2018.

[41] L.A. Ludovico, D. Malchiodi and L. Zecca, "A multimodal LEGO®-based learning activity mixing musical notation and computer programming," in MIE 2017 - *Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education*, Co-located with ICMI 2017, pp. 44-48, 2017.

[42] E.R. Kazakoff and M.U. Bers, "Put your robot in, put your robot out: Sequencing through programming robots in early childhood," *Journal of Educational Computing Research*, vol. 50, pp. 553-573, 2014.

[43] E.R. Kazakoff, A. Sullivan and M.U. Bers, "The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in *Early Childhood Education Journal*, vol. 41, pp. 245-255, 2013.

[44] J.P. Gibson, "Teaching graph algorithms to children of all ages," in *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, pp. 34-39, 2012.

[45] . Mazzoni and M. Benvenuti, "A robot-partner for preschool children learning english using Socio-Cognitive Conflict," *Journal of Educational Technology & Society*, vol. 18, pp. 474-485, 2015.

[46] A. Sempere, "Animatronics, children and computation," *Journal of Educational Technology & Society*, vol. 8, pp. 11-21, 2005.

[47] A. Strawhacker and M.U. Bers, ""I want my robot to look for food": Comparing kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces," *International Journal of Technology and Design Education*, vol. 25, pp. 293-319, Aug 2015. 2015

[48] W. Sung, J. Ahn and J.B. Black, "Introducing Computational Thinking to young learners: practicing computational perspectives through embodiment in mathematics education," *Technology, Knowledge and Learning*, vol. 22, pp. 443-463, Oct 2017. 2017

[49] A. Aggarwal, C. Gardner-Mccune and D.S. Touretzky, "Evaluating the effect of using physical manipulatives to foster computational thinking in elementary school," in *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, pp. 9-14, 2017

[50] A. Djurdjevic, C. Pahl, I. Fronza and N. El Ioini, "A pathway into computational thinking in primary schools," *Lecture Notes in Computer Science*, vol. 10108 LNCS, pp. 165-175, 2017.

[51] Y.-. Kim and J.-. Kim, "Application of a software education program developed to improve computational thinking in elementary school girls," *Indian Journal of Science and Technology*, vol. 9, 2016.

[52] S. Esper, S.R. Foster, W.G. Griswold, C. Herrera and W. Snyder, "CodeSpells: Bridging educational language features with industry-standard languages," in *ACM International Conference Proceeding Series*, pp. 51-60, 2014.

[53] Giannakoulas and S. Xinogalos, "A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students," *Education and Information Technologies*, vol. 23, pp. 2029-2052, 2018.

[54] W.J. Rijke, L. Bollen, T.H.S. Eysink and J.L.J. Tolboom, "Computational thinking in primary school: An examination of abstraction and decomposition in different age groups," *Informatics in Education*, vol. 17, pp. 77-92, 2018.

[55] M. Akcaoglu, "Design and implementation of the game-design and learning program," *Tech Trends*, vol. 60, pp. 114-123, Mar 2016. 2016.

[56] S. Rose, J. Habgood and T. Jay, "Pirate plunder: Game-based computational thinking using scratch blocks," in *Proceedings of the European Conference on Games-based Learning*, pp. 556-564, 2018.

[57] F. Kalelioglu and Y. Gülbahar, "The effects of teaching programming via Scratch on problem solving skills: A Discussion from Learners' Perspective," *Informatics in Education*, vol. 13, pp. 33-50, 2014.

[58] G. Chiazzese, G. Fulantelli, V. Pipitone and D. Taibi, "Promoting computational thinking and creativeness in primary school children," in *ACM International Conference Proceeding Series*, 2017.

[59] M. Mladenović, I. Boljat and Ž. Žanko, "Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level," *Education and Information Technologies*, vol. 23, pp. 1483-1500, Jul 2018. 2018.

[60] A. Oluk, "Comparing students' Scratch skills with their Computational Thinking skills in terms of different variables," *International Journal of Modern Education and Computer Science*, vol. 8, pp. 1-7, Nov 2016. 2016.

[61] M. O'Sullivan, N. Robb, S. Howell, K. Marshall and L. Goodman, "Designing inclusive learning for twice exceptional students in Minecraft," *Journal of Distance Education (Online)*, vol. 32, pp. 1-25, 2017.

[62] J. Raiyn, "Developing a mathematics lesson plan based on visual learning technology," *International Journal of Education and Management Engineering*, vol. 4, pp. 1-9, Jul 2016. 2016.

[63] B. Zhong, Q. Wang, J. Chen and Y. Li, "Investigating the period of switching roles in pair programming in a primary school," *Journal of Educational Technology & Society*, vol. 20, pp. 220-233, 2017.

[64] D. Basso, I. Fronza, A. Colombi, and C. Pahl. Improving assessment of computational thinking through a comprehensive framework. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research (Koli Calling '18).* ACM,2018.

[65] A., Colombi, I., Fronza, C., Pahl, and D., Basso, COCONATS: Combining Computational Thinking didactics and software engineering in K-12. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education (SIGITE '18*). ACM, 2018.