A Framework for Computational Thinking Based on a Systematic Research Review

Article · May 2016

3 authors:

Filiz Kalelioglu
Baskent University

Yasemin Gulbahar
Ankara University

Volkan Kukul
Gazi University

Baltic J. Modern Computing, Vol. 4 (2016), No. 3, 583-596

A Framework for Computational Thinking Based on a Systematic Research Review

Filiz KALELİOĞLU[1], Yasemin GÜLBAHAR[2], Volkan KUKUL[3]

[1] Computer Education and Instructional Technology Department.
Faculty of Education, Baskent University, Ankara, Turkey
[2] Department of Informatics. Ankara University, Ankara, Turkey
[3] Distance Education Research and Application Centre.
Gazi University, Ankara, Turkey

filizk@baskent.edu.tr, ysmnglbhr@gmail.com, kukulvolkan@gmail.com

Abstract. Computational Thinking (CT) has become popular in recent years and has been recognised as an essential skill for all, as members of the digital age. Many researchers have tried to define CT and have conducted studies about this topic. However, 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. In the light of this state of affairs, the purpose of this study is to examine the purpose, target population, theoretical basis, definition, scope, type and employed research design of selected papers in the literature that have focused on computational thinking, and to provide a framework about the notion, scope and elements of CT. In order to reveal the literature and create the framework for computational thinking, an inductive qualitative content analysis was conducted on 125 papers about CT, selected according to pre-defined criteria from six different databases and digital libraries. According to the results, the main topics covered in the papers composed of activities (computerised or unplugged) that promote CT in the curriculum. The targeted population of the papers was mainly K-12. Gamed-based learning and constructivism were the main theories covered as the basis for CT papers. Most of the papers were written for academic conferences and mainly composed of personal views about CT. The study also identified the most commonly used words in the definitions and scope of CT, which in turn formed the framework of CT. The findings obtained in this study may not only be useful in the exploration of research topics in CT and the identification of CT in the literature, but also support those who need guidance for developing tasks or programs about computational thinking and informatics.

Keywords: computational thinking, content analysis, literature review, CT framework

1 Introduction

"Computational Thinking" (CT) as a concept has become popular in recent years; especially after being defined by Wing in 2006. Until recently, computing was considered a specialist skill possessed by computer scientists, engineers, mathematicians and those from similar disciplines. However, nowadays almost everybody, irrespective of age, is expected to have some basic computing skills in parallel with the developments in technology. Hence, being a digital citizen requires students to possess

CT skills as defined by ISTE (2007), and also indicated in the "Framework for K-12 Science Education" (NRC, 2011).

Many researchers started to write about CT in addition to undertaking research studies. However, 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. For the purposes of this paper, the authors created a 'Wordle' (see Figure 1), based on the well-known definitions of CT by (Wing, 2006; Lee *et al.*, 2011; Chang, 2011; Zhenrong, Wenming and Rongsheng, 2009; Liu and He, 2014; Soh *et al.*, 2009; Miller *et al*., 2013; Wang and Zhou, 2011; Wentworth, 2010; Yevseyeva and Towhidnejad, 2012; Carnegie Melon University Centre for Computational Thinking, 2015; Barr, Harrison and Conery, 2011; CSTA and ISTE, 2011; Gouws, Bradshaw, and Wentworth, 2013; Lu and Fletcher, 2009; NRC, 2010; Aho, 2012; Bers, 2010; Denning, 2009; Hudkins, 2013; Ioannidou, Bennett, Repenning, Koh and Basawapatna, 2011; Lye and Koh 2014; Marshall, 2011; Wolz, Stone, Pearson, Pulimood, and Switzer, 2011; Lee, Martin, and Apone, 2014). This wordle is based on only the definitions provided by these researchers and may not include some core concepts of CT. Actually, many researchers can have different insights about this concept; however our scientific approach for visualisation revealed the following wordle.

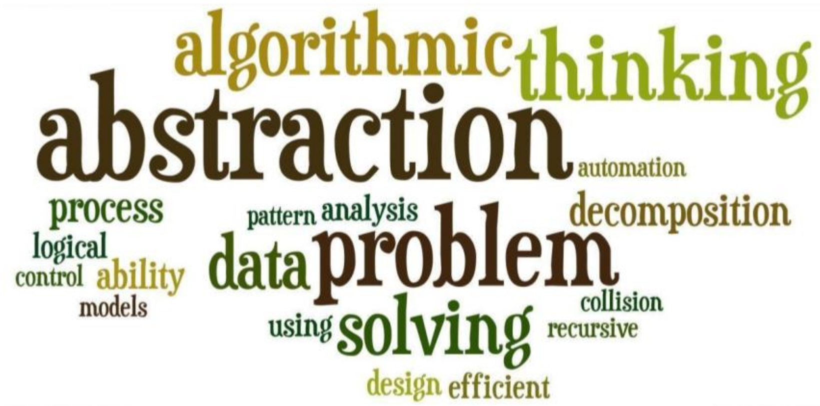

Fig. 1. Most commonly used words in the definitions of CT

Based on the Wordle, the most frequently mentioned words are abstraction, problem, solving, algorithmic and thinking. In the following sections, the authors will try to reveal both the definition and scope of the concept of CT in more detail.

1.1 Definitions of CT in the Literature

CT is a concept that has been growing for the past few years, having been first used by Papert in 1996. In his article, Papert didn't clearly define CT, but in 2006, Jeannette Wing presented CT and defined it as a skill for everyone, not just for computer scientists. Wing (2006) outlined the basic definition of CT as a way of "solving problems, designing systems and understanding human behaviour by drawing on the concepts of computer science". This definition seems general and abstract in terms of integrating CT into curricula and how to observe students' CT ability (Zhenrong, Wenming, and Rongsheng, 2009). Companies like Google and Microsoft supported this idea and several programs and projects have emerged to disseminate CT across different curricula. On the other hand, the International Society for Technology in Education (ISTE) and Computer Science Teacher Association (CSTA) published an operational definition about CT;

CT is a problem-solving process that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them;
- Logically organising and analysing data;
- Representing data through abstractions such as models and simulations;
- Automating solutions through algorithmic thinking (a series of ordered steps);
- Identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources;
- Generalising and transferring this problem solving process to a wide variety of problems (CSTA and ISTE, 2011).

In addition to this definition, Mannila and her colleagues (2014) asserted that CT is a term covering a set of concepts and thinking processes from computer science that help in formulating problems and their solutions in different disciplines. Similarly Riley and Hunt (2014) addressed the cognitive strategies of thinking as “the best way to characterize Computational Thinking is as the way that computer scientists think, the manner in which they reason” (p.4). In depth, Sysło and Kwiatkowska (2013) also underlined that CT is a set of thinking skills that may not result in computer programming. Moreover, CT should “focus on the principles of computing rather than on computer programming skills (p. 50)”.

When examining the definitions in the literature, most of them dwell on problem solving, understanding problems, and formulating problems (Wing, 2006; Zhenrong, Wenming and Rongsheng, 2009; Liu and He, 2014; Barr, Harrison and Conery, 2011). To improve this ability, words such as algorithm and precondition must be a part of everyone’s vocabulary (Zagami, 2013). More words can be ordered for understanding CT and improving CT ability. In line with this purpose, we examined the scope of CT found in the literature. Finally, CSTA and ISTE (2011) defined the vocabulary for CT as: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, simulation and parallelisation.

Based on the categorisation proposed by Brennan and Resnick (2012), Lye and Koh (2014) proposed dimensions of CT in terms of concepts, practices and perspectives. With the dimension of computational concepts, they referred to concepts that programmers use such as variables. For the second dimension, computational practices, they meant problem solving practices that occur in programming processes such as loops and recursion. As the last dimension, namely computational perspectives, the authors referred to self-realisation of students in terms of the technological world around them such as abstracting, questioning and debugging.

1.2 Purpose of the Study

The purpose of this study is to examine the definition, scope and theoretical basis of existing articles and conference papers in the selected databases that focus on CT. It is thought that this research will add benefit to the literature, because it will compose a general framework about the notion of CT. In addition to this purpose, suggestions for teaching and the assessing of CT skills will be discussed in this paper. Specifically, this study is guided by the following questions:
What is reported in terms of the following in CT literature, between 2006 and 2014?

1. purpose of the papers,
2. targeted population,
3. emphasised theoretical/conceptual backgrounds,
4. suggested definitions,
5. chosen framework/scope, and
6. paper types and employed research design

2 Method

The procedures of systematic text analysis were performed as qualitative content analysis. The main idea of the inductive development of categories is to articulate a criterion of characterisation that emerges from theoretical basis and research question. Following this criterion, the text analysis is examined in detail and where categories are uncertain, they are step-by-step reasoned. “Within a feedback loop those categories are revised, eventually reduced to main categories and checked in respect to their reliability. If the research question suggests, quantitative aspects (e.g. frequencies of coded categories) can be analysed” (Mayring, 2000, p. 12).

2.1 Sample

For this research, six different databases and digital libraries were selected. These databases and digital libraries are;

- Ebscohost,
- ScienceDirect,
- Web of Science,
- Springer,
- IEEE Digital Library,
- ACM Digital Library.

These resources were accessed online and “Computational Thinking” searched for in the keywords, title and abstract parts of these resources. Initially, 274 papers were accessed. Intentionally, parallel with the aim of this article, papers which did not mention Computational Thinking conceptually, or did not include the domains and definitions of Computational Thinking, were removed because they were not suitable for the purposes of this study. The remaining 125 papers formed the sample of this study.

2.2 Data Analysis

The papers were divided equally and reviewed by three researchers. Then the researchers started to individually qualitatively analyse the papers according to predefined criteria. The predefined criteria were: purpose, target, theoretical basis, suggested definitions, scope and elements of CT, type of paper, and research methods. During this process, the researchers met weekly to discuss their findings and progress. According to the criteria, papers were coded by each researcher and frequency tables were formed. The tables and codes of each researcher were rechecked and combined into one file in order to see a general picture of the results from across the three researchers. After the creation of a results table, a qualitative data analysis program Nvivo 7.0 was used to create visuals such as Wordle, graphics that depict the most commonly mentioned words in the definition and scope of CT.

3 Results

The results of the inductive analysis were presented according to the research questions below. The papers were analysed according to their purposes, theoretical basis, CT definitions, CT Scope and CT elements.

3.1 Analysis of Papers according to the Purposes

Table 1 depicts the purposes of the 125 examined papers (note: papers may or may not have multiple purposes). 43 papers were about integration and discussion of courses or activities and CT in the curriculum. 34 papers discussed unplugged methods or computational activities (rabbit escape, Simulation Creation Toolkit, Light-Bot, Math on a Sphere, LEGO® Mindstorms NXT, SCRATCH, App Inventor, and CS Unplugged, digital storytelling, Algo.Rhythm, Kodu, Alice, Lego NXT-G…) in order to promote and teach the learning of CT. 26 papers defined and criticised computation or CT in order to understand the notion of CT. 24 papers were found to describe an innovative educational system, design or module designed to engage students with CT concepts. Finally, 13 papers were about presenting a pedagogical framework and four were about CT pattern analysis.

When the purposes of the papers were analysed in depth, the main topics covered in the papers composed of the activities (computer based or unplugged) that promote CT in the curriculum. With many activities, games or use of programming languages, it is suggested that these activities support the teaching of CT skills (such as Apostolellis, Stewart, Frisina, and Kafura, 2014; Basawapatna, Repenning, Koh, and Savignano, 2014; Lee, Martin, and Apone, 2014; Prater and Mazur, 2014; Boechler, Artym, Dejong, Carbonaro, and Stroulia, 2014; Cross, Bartley, Hamner, and Nourbakhsh, 2013).

Table 1. Purpose of the Papers

Purposes of the Papers (note: papers may or may not have multiple purposes)	**# of indicators**
Integration/discussion of courses/activities or CT in the curriculum	43
Unplugged methods or Computational activities (rabbit escape, Simulation Creation Toolkit, Light-Bot, Math on a Sphere, LEGO® Mindstorms NXT, SCRATCH, App Inventor, and CS Unplugged, digital storytelling, Algo.Rhythm, Kodu, Alice, Lego NXT-G…) to promote/teach the learning of CT.	34
Defining/criticising computation or CT / Understanding of the notion of CT (comparison with other thinking types).	26
Describe an innovative educational system/design/module designed to engage students with CT concepts.	24
Presenting a pedagogical framework.	13
CT pattern analysis.	4

3.2 Analysis of Papers according to the targeted population

The targeted population of the papers were also analysed. Based on the analysis, 47 of the papers have participants in the K-12 level, whilst 31 of them were in the higher education level (Table 2).

Table 2. Targeted population

Targeted population	**# of indicators**
K-12	47
Higher Education	31
Total	**78**

3.3 Analysis of papers according to the theoretical basis

The analysis of the papers according to theoretical basis is presented in Table 3. Of the 125 papers, six included theoretical foundations about Game-Based Learning. Five papers mentioned about constructionism, three covered the National Research Council's (NRC) framework, two included Positive Technological Development (PTD), another two discussed STEM and lastly two papers discussed Vygotsky's Zone of Proximal Development theory for their studies.

Table 3. Theoretical basis of the Papers

Theoretical basis	# of indicators
Game-Based Learning	6
Constructionism	5
National Research Council's (NRC) framework	3
Positive Technological Development (PTD)	2
STEM	2
Vygotsky's Zone of Proximal Development theory	2

When the theoretical basis of the papers was examined, it is seen that the papers lack theory and don't give sufficient theoretical framework basis for the research or idea. Game-Based Learning and constructivism are the main theories covered as the basis for CT papers.

3.4 CT definition based on Analysis

CT definitions explained in the papers were analysed and are presented in Figure 2 below. Of the 125 papers, the words used to describe the meaning of the word CT was arranged from most to least: problem solving (22%), abstraction (13%), computer (13%), process (9%), science (7%), data (7%), effective (6%), algorithm (6%), concepts (5%), ability (5%), tools (4%) and analysing (4%).

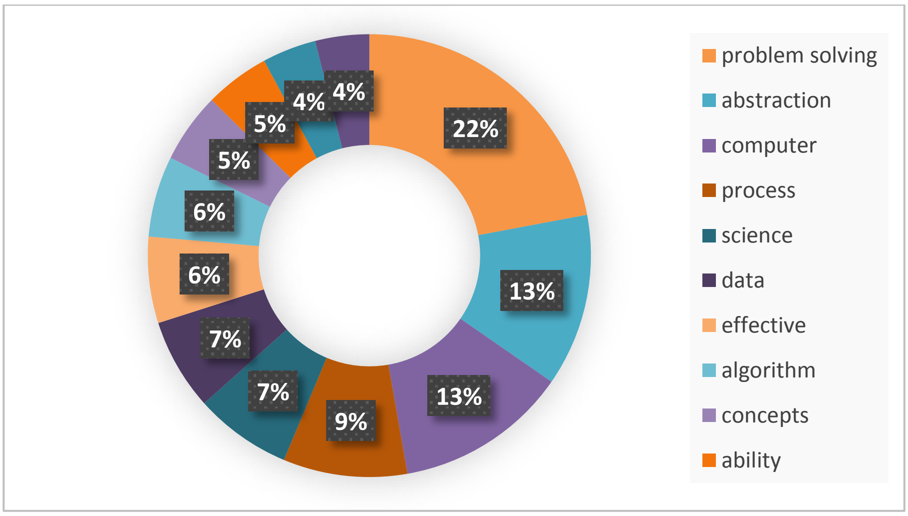

Fig. 2. Mostly used words in CT Definitions

3.5 CT Scope and elements based on Analysis

Based on the literature articles reviewed, the most common defined features that form the scope of CT is abstraction (17), problem solving (12), algorithmic thinking (11), pattern recognition (8), and design-based thinking (6). This finding is consistent with the finding of Ioannidou *et al*., (2011). As underlined by Ioannidou *et al*., (2011), among computer science articles (Lu and Fletcher, 2009; Orr, 2009; Qin, 2009; Qualls and Sherrell, 2010), the most common characteristics of CT are abstraction and problem solving.

Our inductive analysis of reaching themes for defining the scope of CT ended up with 14 aspects (Figure 3).

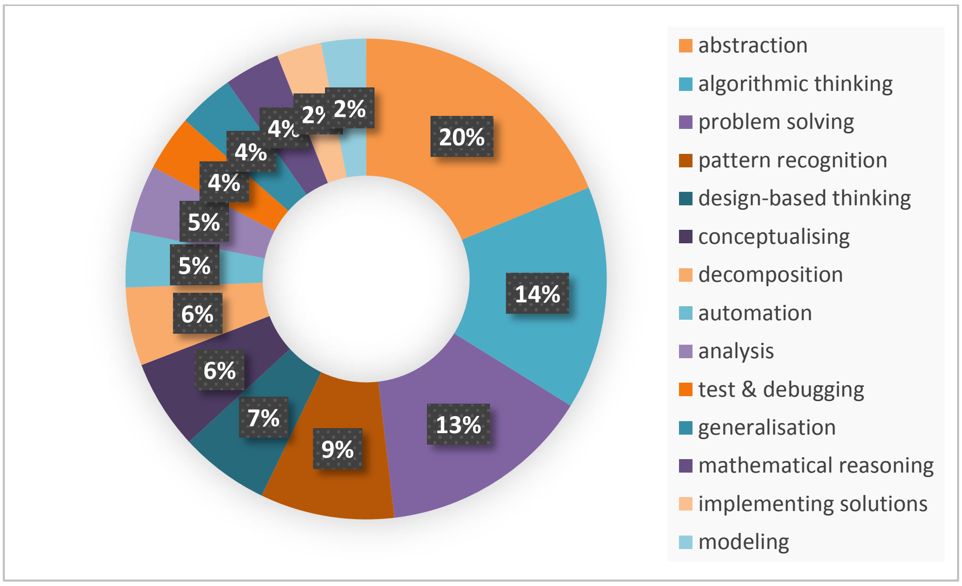

Fig. 3. Mostly used words in CT Scope

In fact, Figure 3 focuses on thinking patterns for design and algorithm, whereas other themes in general form or exist in the steps of problem solving. Thus, CT is a collection of key mental tools and practices originated in computing but addressed to all areas far beyond computer science. In other words, CT is a very important and useful mode of thinking in almost all disciplines and school subjects as an insight into what can and cannot be computed.

3.6 Analysis of papers according to types and research method

Of the 125 papers reviewed for this study, 102 (82%) are conference papers, whereas 23 (18%) papers are journal papers. Of the 125 papers reviewed, 39 were idea papers, as in they were not based on a research method, and were composed mainly of personal views about CT. 33 of the papers were based on quantitative measures, whilst 31 were conducted as qualitative measures, 11 papers were literature reviews about CT, and 11 papers were based on mixed research methods (Table 4).

Table 4. Research methods of the papers

Research method	# of indicators
Idea paper	39
Quantitative	33
Qualitative	31
Literature review	11
Mixed	11
Total	**125**

4 Discussion and conclusion

The purpose of this article was to explore how CT is defined and what forms the scope of this concept. It is found that the literature is at an early stage of maturity and some facts about this concept are still changing. Although teaching of this concept was previously considered special to some disciplines and age groups, it is now expected that all ages and disciplines possess CT skills. The reason may be down to the availability of technology, the rapid growth of computer performance and the subsequent advantages of living in a digital world. On the other hand, it is more advantageous to handle real life problems if you possess CT skills.

In light of these facts, this research study revealed that the origin of CT literature was no more than ten years old and most of the papers are at the stage of ideas. Hence, most of the papers are lacking theoretical or conceptual backgrounds and also they lack research designs. There is no in-depth research about the concept and no discussions about scientific value. There are just a few organisations and researchers who try to define the concept and provide a framework for teachers and students. Since there are no alternatives or grounded work, we have used the available information, although there are no scientific proofs of what is being put forward.

This situation was very much as expected, since there are no more than 500 papers (as of 2015), which is almost saying no research at all when compared to many other disciplines. Furthermore, there is no accepted or well-known definition which has been scientifically proven, and that various researchers seem to perceive the concept in a slightly different way. There are researchers who suggest various instructional approaches to best equip their students with CT skills, but they approach the phenomenon from different perspectives. Hence, they can neither guarantee that the tasks and/or programs they are implementing are really qualified to deliver CT skills to students, nor can they assess if the students are equipped with predefined CT skills after treatment. However, the operational definition of CT could serve as a framework for designing learning activities which aim at comprehending the CT approach in dealing with problems and developing their solutions – to reach this goal one needs to follow the steps of the operational definition of CT with some modifications and variations when needed.

To summarise, it can be beneficial to teach CT by starting discussions on the following;

- how to teach CT skills,
- how to assess if our students really have CT skills, and
- how to assess if our students can adopt CT skills into real-life situations.

In order to achieve this goal, we first have to define CT skills in detail. From analysis of the findings, we found that abstraction, algorithmic thinking, problem solving, pattern recognition, and design-based thinking were the top five skills underlined by researchers, and it was obvious that even the definition of CT also consists of thinking types such as algorithmic and design-based. Moreover, CT includes practices such as problem representation, abstraction, decomposition, simulation, verification, and prediction that are also central to modelling, reasoning, and problem solving in a large number of scientific and mathematical disciplines (NRC, 2010).

Before illustrating the use of the framework for Computational Thinking, it is worthwhile to discuss its intended scope. The framework is proposed to articulate the core elements of Computational Thinking that commonly facilitates coordination among humans or computer systems within a problem-solving process. It is meant to be applicable in either a computerised or unplugged problem-solving process, but it is not intended to be all-inclusive. Drawn as it is from document analysis from the literature, the framework for CT may have limitations in the sense that it may not cover all functions that computational processes can perform.

It may not be necessary to follow all the steps and cover all the core elements, while using this framework for different purposes like task or problem generation or providing solutions, whereas some missing elements (formulating solutions, reusing etc.) should be added if the intent is to create comprehensive course content. It is easy to imagine a setting where only one element is in charge (e.g., abstraction, automation), yet computational thinking occurs. Furthermore, all five steps and core elements could be in place, perhaps poorly implemented, with little resultant computational thinking. Thus the core elements of the framework for CT are, strictly speaking, neither necessary nor sufficient for a complete computational thinking process besides teaching or experiencing its functions.

This framework is as scalable as it sounds. It is for analysing the processes in which humans or computers are leveraged in the hope of bringing about coordination and collaboration. Hence, as researchers, we don't make any claim that this framework for CT will be the best framework for all possible cases; however, we argue that it can be effectively applied to any instance of teaching informatics and computational thinking.

Voskoglou and Buckley (2012) stated that CT was a new problem-solving process due to its use in the computer science field and added that this skill also synthesises critical thinking and knowledge to solve complex problems. Hence, in parallel with this idea and the similarity of the concepts with problem solving steps that emerged from the analysis and literature review, we want to base our results as a practice for CT processes, in parallel with problem-solving; effectively, "Computational Thinking as a Problem Solving Process", as a way to reveal the skills needed for CT. Hence, based on our findings and CT vocabulary suggested by CSTA and ISTE (2011), we developed the following framework to be used as a roadmap for future studies (Table 5).

Table 5. Framework for Computational Thinking as a Problem-Solving Process

Identify the problem	**Gathering, representing and analysing data**	**Generate, select and plan solutions**	**Implement solutions**	**Assessing solutions and continue for improvement**
Abstraction (3) Decomposition (3)	Data collection (2) Data analysis (3) Pattern recognition (1) Conceptualising (1) Data representation (2)	Mathematical reasoning (1) Building algorithms and procedures (3) Parallelisation (2)	Automation (3) Modelling and simulations (3)	Testing (1) Debugging (1) Generalisation (1)

The proposed framework is composed of a combination of both the scope of CT and also problem solving. As it is thought that CT is complex higher-order thinking, skills may require to use the power of human cognitive ability and embrace the support of machines to think and solve problems.

In this framework, where there is a number 1 in the parenthesis, this indicates that it is gathered from analysis of selected papers in this study; number 2 indicates that it is mentioned in the operational definition of CT and lastly, number 3 indicates that the actions in this category are based on our findings and the operational definition of CT. Briefly, this framework depicts the process of CT as a problem-solving process. One can start identifying the problem through abstraction and decomposition. To profoundly understand and solve a problem, the process of gathering, representing and analysing data should be employed. In this process, data collection and analysis, pattern recognition, conceptualising and data representation are the main actions that should be taken into consideration. To provide more accurate solutions, some cognitive processes can be applied such as mathematical reasoning, parallelisation building and algorithms, and procedures. To implement solutions, automation, modelling and simulations can put into action and lastly, to assess the accuracy of the solutions, testing and debugging should be performed. As a final step, for the purposes of continuity, transferring and applying the solution to different type of problems should be realised in order to generalise the solutions.

This framework is within the development phase and is not yet finalised. Moreover, there is no certain distinction among the subcategories and similar actions can be processed in multiple categories. In this framework, we tried to represent CT as a problem-solving process, but in terms of computational tasks. However, CT can be used in much wider context learning environments than simply developing a computerised solution. As such, the next issues in this subject should be:

- to write down learning tasks for each domain, and also for each different age group,

- pedagogically try to teach CT skills to students, and
- to find ways to assess to what extent students are equipped with the required skills.

Meanwhile, CT dispositions are also important in the process of teaching and learning in order to support and enhance CT skills. Tishman and Andrade (1995) stated that "thinking dispositions are tendencies toward particular patterns of intellectual behaviour" (part 7). CSTA and ISTE (2011) also underlined dispositions such as confidence in dealing with complexity, stability to work with difficult problems, tolerance for uncertainty, and the ability to engage with open-ended questions, and communicate and cooperate with others to overcome a common solution.

As a conclusion, it would be easier to teach CT in a planned way by following the steps of problem solving offered in the framework. It is considered that this framework could be used as a roadmap to teach and learn, and to practice CT and informatics concepts within many courses. Supporting this fact, Dagiene and Stupuriene (2016) stated to give "emphasis on developing computational thinking, programming, designing computational systems and other basic concepts of informatics" (p.33). Although this study is only a first attempt to define the learning goals for equipping students with CT skills, more studies are needed in order to validate the framework based on teaching the processes of CT skills. Researchers can focus on teaching and assessment dimensions, either by developing scales and tasks that assesses CT skills, or by concentrating on the dispositions which have a direct effect on learning.

References

Aho, A.V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835.

Apostolellis, P., Stewart, M., Frisina, C., Kafura, D. (2014). RaBit EscApe: A Board Game for Computational Thinking. In: *Proceeding IDC '14: Proceedings of the 2014 conference on Interaction design and children*, Aarhus, Denmark, 349-352.

Barr, D., Harrison, J., Conery, L. (2011). Computational Thinking: A Digital Age. *Learning & Leading with Technology*, March/April, 20-23.

Basawapatna, A., Repenning, A., Koh, K.H., Savignano, M. (2014). The Consume - Create Spectrum: Balancing Convenience and Computational Thinking in STEM Learning. In: *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, Atlanta, USA, 659-664.

Bers, M. U. (2010). The TangibleK Robotics Program: Applied Computational Thinking for Young Children. *Early Childhood Research and Practice,* 12(2). http://ecrp.uiuc.edu/v12n2/bers.html

Boechler, P., Artym, C., Dejong, E., Carbonaro, M., Stroulia, E. (2014). Computational Thinking, Code Complexity, and Prior Experience in a Videogame-Building Assignment. In: *Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies,* Athens, Greece, 396-398.

Brennan, K., Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In: *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada.

Carnegie Melon University Centre for Computational Thinking. (2015). http://www.cs.cmu.edu/~CompThink/index.html.

Chang, C.-K. (2011). Integrate social simulation content with game designing curriculum to foster computational thinking. In: *Proceedings of the 7th International Conference on Digital Content, Multimedia Technology and its Applications,* Busan, South Korea, 115-118.

Cross, J., Bartley, C., Hamner, E., Nourbakhsh, I. (2013). A visual robot-programming environment for multidisciplinary education. In *Proceedings of IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 445-452.

CSTA and ISTE (2011). *Computational Thinking in K–12 Education leadership toolkit*. http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf.

Denning, P. J. (2009). Beyond Computational Thinking. *Communications of the ACM*, 52(6), 28-30.

Dagiene, V., Stupuriene, G. (2016). Bebras - a Sustainable Community Building Model for the Concept based Learning of Informatics and Computational Thinking. *Informatics in Education, 15*(1), 25-44

Gouws, L. A., Bradshaw, K., Wentworth, P. (2013). Computational thinking in educational activities: an evaluation of the educational game light-bot. In: *Proceedings of the 18th ACM Conference on Innovation And Technology In Computer Science Education*, Canterbury, United Kingdom, 10-15.

Hudkins, D. (2013). Why We Must Require Computer Science Education Now. *Independent School*, 72(4), 76-80.

Ioannidou, A., Bennett, V., Repenning, A., Koh, K., Basawapatna, A. (2011). Computational Thinking Patterns. In: *Paper Presented at Annual Meeting of the American Educational Research Association (AERA),* New Orleans, Louisiana.

ISTE. (2007). *ISTE Standards for Students.* http://www.iste.org/docs/pdfs/20-14_ISTE_Standards-s_PDF.pdf.

Lee, I., Martin, F., Apone, K. (2014). Integrating Computational Thinking Across The K–8 Curriculum. *ACM Inroads*, 5(4), 64-71.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32-38.

Liu, B., He, J. (2014). Teaching Mode Reform and Exploration on the University Computer Basic based on Computational Thinking Training in Network Environment. In: *Proceedings of the 9th International Conference on Computer Science & Education (ICCSE 2014),* Vancouver, Canada, 59-62.

Lu, J. J., Fletcher, G.H. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, 41(1), 260-264.

Lye, S. Y., 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*, 41, 51–61.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., Settle, A. (2014). Computational thinking in K-9 education. In: *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, ITiCSE-WGR 2014, 1–29. ACM, New York.

Marshall, K. S. (2011). Was that CT? Assessing Computational Thinking Patterns through Video-Based Prompts. *Paper Presented at 2011 Annual Meeting of the American Educational Research Association*, New Orleans, Louisiana, USA.

Mayring, P. (2000). Qualitative Content Analysis. *Forum: Qualitative Social Research,* 1(2), http://www.qualitative-research.net/index.php/fqs/article/view/1089/2385

Miller, L., Soh, L.- K., Chiriacescu, V., Ingraham, E., Shell, D., Ramsay, S., Hazley, M. P. (2013). Improving Learning of Computational Thinking Using Creative Thinking Exercises in CS-1 Computer Science Courses. In: *Proceedings of the frontiers in education conference (FIE2013).* Oklahoma City, OK.

NRC (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking.* Washington, D.C.: The National Academies Press.

NRC (2011). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas.* Washington, D.C.: The National Academies Press.

Orr, G. (2009). Computational thinking through programming and algorithmic art. In: *Proceedings of the 36 the International conference and Exhibition on Computer Graphics and Interactive Techniques,* New Orleans, LA, USA.

Papert, S. (1996). An Exploration in the Space of Mathematics Educations. *International Journal of Computers for Mathematical Learning,* 95-123.

Prater, L., Mazur, J. M. (2014). Embedded standards-based digital gaming assessments: Pilot study with teachers. In*: Proceedings of the 19th International Computer Games Conference,* Louisville, Kentucky, USA. 1-5.

Qin, H. (2009). Teaching Computational Thinking through Bioinformatics to Biology Students. In: *Proceedings of the 40th ACM technical symposium on Computer science education,* Chattanooga, TN USA , 188-191.

Qualls, J. A., Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, 25(5), 66-71.

Riley, D. D., Hunt, K. A. (2014. *Computational thinking for the modern problem Solver.* Boca Raton, FL : CRC Press.

Soh, L.-K., Samal, A., Scott, S., Ramsay, S., Moriyama, E., Meyer, G., Moore, B., Thomas, W. G., Shell, D. F. (2009). Renaissance computing: an initiative for promoting student participation in computing. *ACM SIGCSE Bulletin,* 41(1), 59-63.

Sysło, M. M., Kwiatkowska, A. B. (2013). Informatics for All High School Students : A Computational Thinking Approach. *In: Diethelm, I., Mittermeir, R.T. (eds.) ISSEP 2013. LNCS*, 7780, 43–56. Springer, Heidelberg

Tishman, S., Andrade, A. (1995). *Thinking dispositions: A review of current theories, practices, and issues.* ACCTION report #1. Washington, DC. ACCTION. http://learnweb.harvard.edu/alps/thinking/gettingready_dispositions.cfm

Voskoglou, M. G., Buckley, S. (2012). Problem Solving and Computers in a Learning Environment. *Egyptian Computer Science Journal*, 36(4). 28-46.

Wang, X., Zhou, Z. (2011). The research of situational teaching mode of programming in high school with Scratch. In: *Proceedings of the 7th IEEE Joint International Information Technology and Artificial Intelligence Conference,* Chongqing, China, 488-492.

Wentworth, P. (2010). Can Computational Thinking Reduce Marginalization In The Future Internet? In: *Proceedings of ITU-T Kaleidoscope 2010: Beyond the Internet? - Innovations for future networks and services,* Pune, India, 1-5.

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

Wolz, U., Stone, M., Pearson, K., Pulimood, S.M., Switzer, M. (2011). Computational thinking and expository writing in the middle school. *Journal of ACM Transactions on Computing Education*, 11(2).

Yevseyeva, K., Towhidnejad, M. (2012). Work in Progress: Teaching Computational Thinking in Middle and High School. In: *Proceedings of Frontiers in Education Conference (FIE),* Seattle, Washington , 1-2.

Zagami, J. (2013). *Computational Thinking.* Brisbane, QLD: EduTechPress.

Zhenrong, D., Wenming, H., Rongsheng, D. (2009). Discussion of ability cultivation of computational thinking in course teaching. In: *Proceedings of International Conference on Education Technology and Computer,* Singapore, 197 – 200.

Received April 20, 2016, accepted May 23, 2016