DePryck, K. (2016). From computational thinking to coding and back. In F. J. García-Peñalvo (Ed.), *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016)* (pp. 27-29). New York, NY, USA: ACM. doi: 10.1145/3012430.3012492.

From computational thinking to coding and back

Koen DePryck
Vrije Universiteit Brussel
Dept of Educational Sciences
&
Superintendent of education
GO! school district Antwerpen
koen.depryck@vub.ac.be

ABSTRACT

Introducing coding in the curriculum at an early age is considered a long term investment in bridging the skills gap between the technology demands of the labour market and the availability of people to fill them. The keys to success include moving from mere literacy to active control – not only at the level of learners but also at the level of teachers. However, given the fast development of the field, one might wonder whether acquiring specific coding skills really is the essence of introducing coding early in the curriculum. We argue that the reach of ICT –including coding skills-- is much broader than STEM alone and a background in STEM is no longer a requirement for successful coding. The complex link between coding and computational thinking is the real critical success factor. We refer to TACCLE3 (an EU Erasmus+ project) as a successful approach to the implementation and valorisation of computational thinking across the curriculum.

CCS Concepts

• Social and professional topics→Computational thinking

• Social and professional topics→K-12 education

Keywords

Computational Thinking; Coding; STEM; TACCLE3

1. INTRODUCTION

The need for programming professionals is still growing. The European Commission estimates there are 700,000 unfilled vacancies for IT practitioners, of which programmers outnumber other IT professionals by ratio 5 to 1. In the Netherlands, development jobs account for 51,6% of all IT jobs. The limited response from the different European education systems is worrisome, although there are signs that awareness is on the rise. Estonia has introduced programming across the curriculum in 2012, Denmark is following suit, and, so are the UK, Finland and Ireland. In Germany some regions are more on the forefront than others. It should be noted that schools are not the only answer to this challenge. The likes of after-school coding clubs as well as initiatives from the IT industry itself are good and practical examples, and are all key stakeholders.

Introducing coding in the curriculum at an early age is a long term investment in bridging the skills gap between the technology demands of the labour market and the availability of people to fill them. The key seems to be moving from mere literacy to active control. However, given the rapid development of the field, one might wonder whether acquiring *specific* coding skills really is the essence of introducing coding in the curriculum.

2. CODING VS COMPUTATIONAL SKILLS

While serious coding is essentially a text-based operation (even when the functionality of programming environments now resembles that of a very sophisticated word processor with auto correction) learning to program in schools was not all that long ago equated with learning the vocabulary and syntax of programming. In other words, "getting the program to run correctly is the overarching objective of today's computer programming classroom" [16].

Today we have the option to work in programming environments in which one can drag visual blocks to write programs. Both, text as well as visual elements, are, of course, in essence additional layers above what is, ultimately, a structure of electric pulses, O's and I's, well defined (binary) changes in a structure.

Definitions of computational thinking come in many flavours, but "the thought process involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" nicely catches the core of the concept [7]. While the intuitive connotation to "information-processing agent" might be "computer', not humans themselves but an incredible range –perhaps all—organisms and objects process information in the sense that they react to stimuli and internal and external changes.

DePryck, K. (2016). From computational thinking to coding and back. In F. J. García-Peñalvo (Ed.), *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016)* (pp. 27-29). New York, NY, USA: ACM. doi: 10.1145/3012430.3012492.

Against that background, the very conceptual and later physical design of "computers" is the result of quite impressive computational thinking.

The assumption underlying the importance given to coding, including high level visual coding, is that a "specific type of learning" takes place when young people are engaged in programming, activities. For instance it is believed that programming interactive media (for example Logo and Scratch) support the development of computational thinking among young people" [3].

Realistically, programming is based on an analysis of the metacognitive abilities required for computational thinking, including connecting new information to former knowledge, deliberately selecting thinking strategies, planning, monitoring and evaluating thinking processes, breaking down complex actions into a conditional sequence [2].

These metacognitive abilities in turn rely on executive functions, a construct uniting working memory, attention, and inhibitory control for the purpose of planning and executing a goal directed activity [4; 15; 19]. More specifically, executive functions as mentioned above are necessary but not sufficient antecedents to metacognitive skills, which are then themselves basis of self-regulated learning [6].

Relevant findings in light of this include and concluded that (younger children's) immature executive functions may limit their metacognitive skill use [4]; that students having programming experience scored higher on various cognitive ability test (reflectivity, divergent thinking…) [8]

However, if teachers believe that the only computing experiences available to students occur through learning text based programming languages such as C++ they may never consider introducing computing at the earlier grades [14].

High levels of motivation and self-regulation are clearly associated with academic achievement, independent of measured intelligence [1; 12; 17].

So, the question remains if coding and its associated problem solving capability, at the level of K-12 and in a context and format suitable for age and cognitive level may contribute to the development of executive functions, metacognition and self-regulation.

Problem solving is a necessary but not a sufficient criterion for programming [13]. This research makes the case for the value of cognitive development during coding as opposed to the focus on the value of delivering a functional program.

The authors of the JKarelRobot software for learning programming mentioned that their mismatching the lesson to the level of cognitive development of their students resulted in a disaster. The conclusion reached by Buck and Stucki [5] after studying the JKarelRobot software was quite interesting, in their words: "We have recognized that the natural tendency to teach according to the structure of one's own understanding runs contrary to established models of cognitive development".

Programming sets the stage for and encourages children to think about their thinking, which is a capacity important to developing as a self-regulated learner. Computational thinking focuses on the process of thinking and learning, moving beyond "what are you learning" to "how are you learning" [3].

3. STEM

The link with STEM may be less crucial than one might expect. As mentioned earlier, the development of interactive coding environment catering to a specific segment (business, science, arts…) may open up opportunities for coding in very different domains, without necessarily requiring an interest in STEM. While the need for more students to become interested in STEM is needed in its own right, strongly connecting that need to coding might not only scare away learners without that explicit interest in STEM but might also hinder the pervasive inclusion of coding skills in a broad array of domains. A STEM mindset can be developed and integrated with programming skills, but that is by no means a requirement or a desired outcome. The reach of ICT is much broader than STEM alone and a background in STEM is no longer a requirement for successful coding.

4. TACCLE3

TACCLE 3 (see taccle3.eu; funded under Erasmus+, succeeding TACCLE which focused on developing teacher's competences in developing digital content and TACCLE 2 which focused on learners using programs and apps to create their own content) [9-11; 18] is an EU funded strategic partnership (2015-2017) for school education focusing on introducing coding in schools in light of a competitive strategy after 2020. In order to do so, the partnership identified the following needs:

- Professionalization and upskilling of teachers;
- Enhanced digital literacy for teachers and learners; and
- A positive mindset of young children towards coding and STEM.

From a critical perspective and in light of the previous sections, we would like to make the following comments. Metacognitive strategies must be an integral part of such a project. The pre-service and in-service professionalization of teachers should therefore not only include the use and mastery of coding environment but also insights in metacognitive strategies and tools to teach them. Learning how to guide students in the use of metacognitive strategies may be essential.

5. CONCLUSIONS

Introducing coding in the curriculum is really not about coding itself. It is, rather, about introducing a culture of algorithmic thinking, breaking down more complex actions into a sequence of instructions, and computational thinking, focusing on problems and their

DePryck, K. (2016). From computational thinking to coding and back. In F. J. García-Peñalvo (Ed.), *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016)* (pp. 27-29). New York, NY, USA: ACM. doi: 10.1145/3012430.3012492.

solutions. This computational thinking is itself based on a set of metacognitive strategies in a wide array of domains, applicable beyond coding.

6. ACKNOWLEDGMENTS

With the support of the EU Erasmus+ Programme. KA2 project "TACCLE 3 – Coding" (2015-1-BE02-KA201-012307).

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

7. REFERENCES

[1] Blair, C., 2002. School readiness: Integrating cognition and emotion in a neurobiological conceptualization of children's functioning at school entry. *American Psychologist 57*, 2, 111-127. DOI=http://dx.doi.org/http://dx.doi.org/10.1037/0003-066X.57.2.111.

[2] Blakey, E. and Spence, S., 1990. *Developing Metacognition.* ERIC.

[3] Brennan, K. and Resnick, M., 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the Annual American Educational Research Association meeting* (Vancouver, BC, Canada2012).

[4] Bryce, D., Whitebread, D., and Szucs, D., 2015. The relationships among executive functions, metacognitive skills and educational achievement in 5 and 7 year-old children. *Metacognition & Learning 10*, 2, 181-198.

[5] Buck, D. and Stucki, D.J., 2001. JKarelRobot: a case study in supporting levels of cognitive development in the computer science curriculum. *SIGCSE Bull. 33*, 1, 16-20. DOI= http://dx.doi.org/10.1145/366413.364529.

[6] Chetty, J. and Van Der Westhuizen, D., 2014. Implementing Metacognition Skills for Learners Studying Computer Programming. In *Proceedings of EdMedia: World Conference on Educational Media and Technology 2014. Jun 23, 2014, Tampere, Finland*, J. Viteli and M. Leikomaa Eds. Association for the Advancement of Computing in Education (AACE), Waynesville, NC, USA, 726-731.

[7] Cuny, J., Snyder, L., and Wing, J.M., 2010. Demystifying computational thinking for noncomputer scientists.

[8] Fessakis, G., Gouli, E., and Mavroudi, E., 2013. Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education 63*(4//), 87-97. DOI= http://dx.doi.org/http://dx.doi.org/10.1016/j.compedu.2012.11.016.

[9] García-Peñalvo, F.J., 2016. A brief introduction to TACCLE 3 – Coding European Project. In *2016 International Symposium on Computers in Education (SIIE)*, F.J. García-Peñalvo and J.A. Mendes Eds. IEEE, USA.

[10] García-Peñalvo, F.J., 2016. Presentation of the TACCLE3 Coding European Project. http://repositorio.grial.eu/handle/grial/654.

[11] García-Peñalvo, F.J., 2016. Proyecto TACCLE3 – Coding. In *XVIII Simposio Internacional de Informática Educativa, SIIE 2016*, F.J. García-Peñalvo and J.A. Mendes Eds. Ediciones Universidad de Salamanca, Salamanca, España, 187-189.

[12] Gottfried, A.E., 1990. Academic intrinsic motivation in young elementary school children. *Journal of Educational Psychology 82*, 525-538.

[13] Hwang, W.-Y., Wang, C., Hwang, G.-J., Huang, Y.-M., and Huang, S., 2008. A web-based programming learning environment to support cognitive development. *Interacting with Computers 20*, 524-534. DOI= http://dx.doi.org/http://dx.doi.org/10.1016/j.intcom.2008.07.002.

[14] Israel, M., Pearson, J.N., Tapia, T., Wherfel, Q.M., and Reese, G., 2015. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education 82*(Mar), 263-279. DOI= http://dx.doi.org/10.1016/j.compedu.2014.11.022.

[15] Lyon, G.R. and Krasnegor, N.A., 1996. *Attention, memory, and executive function*. Paul H Brookes Publishing Company, Baltimore, USA.

[16] Pea, R. and Kurland, D., 1984. On the cognitive effects of learning computer programming. *New Ideas in Psychology 2*, 137–168.

[17] Skinner, E.A., Zimmer-Gembeck, M.J., Connell, J.P., Eccles, J.S., and Wellborn, J.G., 1998. Individual Differences and the Development of Perceived Control. *Monographs of the Society for Research in Child Development 63*, 2/3, i+iii+v-vi+1-231.

[18] Taccle 3 Consortium, 2016. TACCLE 3: Coding Erasmus + Project website. http://www.taccle3.eu/.

[19] Zelazo, P.D., Carter, A., Reznick, J.S., and Frye, D., 1997. Early development of executive function: A problem-solving framework. *Review of General Psychology 1*, 198-226.