What Computational Thinking Is

Francisco José García-Peñalvo

Computer Science Department
Research Institute for Educational Sciences
GRIAL research group
University of Salamanca
JITR Editor-in-Chief

fgarcia@usal.es

Introduction

Current software-driven Society demands skilled professionals for ICT (Information and Communication Technologies) business sector. A very common situation in countries with a high rate of unemployment is they have unfilled positions for engineers and technicians for the industry and digital services. This has caused an increasing approach for introduce digital or information technology (IT) literacy from the early beginning of the individual development till the high school courses (Allan, Barr, Brylow, & Hambrusch, 2010), even in post-secondary institutions (Astrachan, Hambrusch, Peckham, & Settle, 2009), combining it with other key competences such as reading, writing and math skills.

The most frequent approach to teaching digital literacy has been to gradually encourage the learning of programming, and the term code-literacy (diSessa, 2000; Hockly, 2012; Prensky, 2008; Rushkoff, 2012; Vee, 2013) has been coined to referer the process of teaching children programming tasks, from the simplest and most entertaining to the most complex, this way the student’s progress is centred on the difficulty of the tasks and in their motivating characteristic.

Consequently, at the same time that children learn human languages, both for speaking and writing, natural languages, encompassing all matters related with the experimental sciences (physics, chemistry, biology, etc.), and humanity languages, involving social sciences and humanities, it is also necessary they learn digital languages, in which ones the competences to be success in the digital world are included, using coding as the way to solve problems and computational thinking as working paradigm (Llorens-Largo, 2015).

With the awareness of the importance of digital skills and related information technology (eSkills), there are several proposals worldwide about the need to include coding from the curriculum of non-university levels, starting since primary education (or sooner) (Balanskat & Engelhardt, 2015; Brown et al.,
2013), because of the code-literacy skills are becoming understood as a core element for STEM (Science, Technology, Engineering, & Mathematics) subjects (Weintrop et al., 2016) and imaginative programming is the most crucial element of computing because it closely aligns mathematics with computing and in this way brings mathematics to life (Felleisen & Krishnamurthi, 2009). According to this different projects have been develop in different world regions such as Taccle 3 Coding European Project (García-Peñalvo, 2016a; TACCLE 3 Consortium, 2016) that provides practical ideas that teachers can use immediately together with suggestions on how these can be adapted for introducing computing or coding in their classrooms; the 985 project in China that explicitly pointed out that “prominent computational thinking ability must be the fundamental skill for innovative talents in any discipline” (Long, Zhang, & Li, 2013), or Mobile Computing project in EEUU (Turbak, Pokress, & Sherman, 2014), which is devoted to teach the big ideas of computer science to undergraduate students using App Inventor (Pokress & Domínguez Veiga, 2013). These projects are a great example of the connection of the university with the society (García-Peñalvo, 2011, 2016b).

A code-literate person means that can read and write in programming languages (Román-González, 2014), computational thinking is referred to the underlying problem-solving cognitive process that allows it. Thus, coding is a key way to enable computational thinking (Lye & Koh, 2014) and computational thinking may be applied to various kinds of problems that do not directly involve coding tasks (Wing, 2008).

The term computational thinking was made popular by Jeannette M. Wing (2006), however this increasing interest about introducing coding, code-literacy or computational thinking in the pre-university studies, there still exist a lack of consensus on a formal definition of these terms (Barr & Stephenson, 2011; Gouws, Bradshaw, & Wentworth, 2013; Grover & Pea, 2013).

Although coding is so interesting, it is more important to emphasize in the idea of computational thinking as the application of high level of abstraction and an algorithmic approach to solve any kind of problems.

From JITR, we encourage to promote coding and special computational thinking in pre-university studies and also in other university disciplines different from Computer Science.

Contents of the issue
Current JITR issue comprises five papers.

The paper “Virtual Learning Communities in Google Plus, implications and sustainability in MOOCs” (Martínez-Núñez, Borrás-Gene, & Fidalgo-Blanco, 2016) proposes a new model to allocate informal learning and collective intelligence in Massive Open Online Courses (MOOCs) (García-Peña1vo, 2015) using external virtual learning communities through social networks, based on Google+. The main aim of this article is to assess the virtual learning community performance and analyze the interactions and the kinds of learning that take place inside the community and over time.

The paper entitled “Investment Location Selection based on Economic Intelligence and Macbeth Decision Aid Model” (Mostafa, Khaled, Jamila, & Hanoune, 2016) presents a case study that aims to apply some sound MCDM techniques in the case of Economic Intelligence (EI) and show how the use of strategic information may help deciders to choose among geographic locations in which they could settle their investments.

Salem et al. (2016) propose a system that would evaluate answers using Natural Language Processing and lastly compared the results obtain by human expert graders and proposed system.

In the last paper Winley & Singhapong (2016) examine the importance assigned by Human Resource personnel to the personality traits of Information Technology officers.

References


