Computational Thinking in Innovative Computational Environments and Coding

Alberto Ferrari
University of Parma, Italy

Agostino Poggi
University of Parma, Italy

Michele Tomaiuolo
University of Parma., Italy

INTRODUCTION

The concept of Computational Thinking has been discussed for several decades and in recent years has been brought to the attention of the scientific community by Jeanette Wing. Her article presents Computational Thinking as "a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science." (Wing, 2006).

CT is a cognitive process involving logical reasoning that embraces the ability to think algorithmically using abstraction, decomposition and generalization.

The importance of CT places it among the basic skills for 21st century, together with reading, writing and calculation, that every person will have to master, so it is important to teach it already in primary school. As the invention of printing facilitated the spread of the three Rs (reading, writing and arithmetic) technology must lead to the spread of CT.

In our high-technological society teaching CT concepts in all level of instruction allows individuals to participate more equitably in society overcoming the differences now present in mastering these skills. Everyone should be able to apply computational strategies in each domain and understand what problems may be treated automatically.

CT is drawing fundamentally on concepts from computer science, it is not programming, but programming, meant as analysis and solution of problems, allows to highlight all CT features. Easy-to-use computational environments foster students in their first approach to coding and their use are becoming more common in school and university.

This chapter will introduce the research on CT and, in particular, the works on innovative computational environments, and will describe the situation of the education to CT in high school and in academic courses.

BACKGROUND

There is no single definition of CT and the following is the work of several authors who point out different aspects. A common feature is that CT represents a set of skills that are part of a cognitive process related to deal and seek solutions to problems. Abstraction, decomposition and generalization are common features of most of these works.

The term "Computational Thinking" was adopted for the first time by Seymour Papert, with reference to the LOGO programming language (Papert, 1996). According to his theory of "constructionism", programming is a valu-

DOI: 10.4018/978-1-5225-2255-3.ch208

able educational tool that provides the cognitive artifacts necessary to the human mind to build a representation of the world with which it interacts.

"Think like a computer scientist" means being able to deal with a problem, to design a system and understand human behavior using the fundamental concepts and tools of Computer Science (CS). The power of our "mental" tools is amplified by the power of our "metal" tools (Wing, 2008).

There is still confusion over an acceptable definition for the term, in its essence, CT can be viewed as a re-foundation of "algorithmic thinking". In the 1950s and 1960s, algorithmic thinking was understood as a mental attitude, enabling the description of a generic problem as a translation of some input data to output data, and the formalization of the required translation as an algorithm. CT is built on the same basis, with the inclusion of the ability to think about a complex problem at different levels of abstractions, to use mathematical methods, and to analyze data and the complexity of solutions, i.e., to study how a certain algorithm scales when the size of the problem grows.

CT can be defined as the ability to solve complex problems by applying the logic of computing paradigm; it is a set of cognitive skills, concepts and techniques of computer science related to problem solving.

In the 2010 workshop on "The Scope and Nature of Computational Thinking" participants developed a vivid discussion on what "CT for everyone" might mean. Wing describes CT as a bridge between science and engineering. Since it deals with thinking processes and abilities that can be applied to different disciplines, it represents a sort of meta-science. Moursund et al. (National Research Council, 2010) suggest a close relation between CT and Procedural Thinking, as developed by Seymour Papert in Mindstorms (Papert, 1980).

CT revolution goes much deeper than the use of computer in everyday work; it is changing the way we think. Among the features of the CT we find the ability to formulate a problem and represent its solution writing an algorithm and compare the solution with others in order to assess its efficiency.

Another important feature is the ability to represent and organize data logically, using abstraction, generalization and modeling concepts, and identifying patterns within these.

Data abstraction, identification of common features and functionality comes into play both when we have to decide what to abstract and when we have to define the level of abstraction. The process of abstraction must then be followed by a process of analysis to verify the correctness of the assumptions and the quality of the result obtained.

Abstraction is probably the most significant process in CT: abstraction is used to define patterns, to generalize, to allow an object to represent many; abstraction allows you to extract common properties, to scale and then to deal with complexity; abstractions are the mental tools of computing.

Decomposition, the ability to decompose a problem into sub-problems of smaller size, is another important aspect. Decomposing a problem usually leads to the recognition of patterns and generalization, and therefore it also leads to the ability to design algorithms. Pattern recognition, the ability to identify similarities or common differences, is the basis of problem solving and algorithm design.

Generalization of pattern and abstraction allows to represent an idea or a process in general terms and then to be able to use this idea to solve other problems of the same nature. Algorithm design is finally the ability to develop a step-by-step solution for a given problem.

Bundy (2007) focuses, among other things, on the ability to process large amounts of data, Big Data, and describe a new model of science that defines e-science.

As we have seen, there are various definitions of CT and each one highlights certain features. A common factor among the various interpretations is the ability to solve problems using a systematic approach: problem solving.