Computational Thinking: Videogames, Educational Robotics, and other Powerful Ideas to Think with

Rosa Bottino, Augusto Chioccariello
Istituto per le Tecnologie Didattiche
Consiglio Nazionale delle Ricerche
via De Marini 6
16149 Genova, Italy
{bottino, augusto}@itd.cnr.it

Abstract: Digital technology has radically changed the way people work in industry, finance, services, media and commerce. Informatics has contributed to the scientific and technological development of our society in general and to the digital revolution in particular. Computational thinking is the term indicating the key ideas of this discipline that might be included in the key competencies underlying the curriculum of compulsory education. The educational potential of informatics has a history dating back to the sixties. In this article, we briefly revisit this history looking for lessons learned. In particular, we focus on experiences of teaching and learning programming. However, computational thinking is more than coding. It is a way of thinking and practicing interactive dynamic modeling with computers. We advocate that learners can practice computational thinking in playful contexts where they can develop personal projects, for example building videogames and/or robots, share and discuss their construction with others. In our view, this approach allows an integration of computational thinking in the K-12 curriculum across disciplines.

Keywords: Computational thinking, programming in context, informatics education

1 Background

The educational potential of informatics was underlined from the beginning in educational technology research studies. Even if the relationship between computers and teaching to support pupils' thinking in schools has been variously conceptualised (Wegerif, 2002), one of the first identified objectives of the drive towards "computer innovation" in education was to develop new skills in students to allow their integration in a society that is deeply changed by information technologies.

In particular, in early years, an important line of study in educational technology was related to the teaching of programming. The advent of the microcomputer in the early 1980s and the development of general purpose programming languages opened up the possibility of using computers to a wide range of users and stressed the necessity of learning how to interact with them. In different contexts, the introduction to programming main ideas was carried out, through different modalities and approaches, not only in professional courses but also as part of school basic education (Olimpo, Persico, Sarti, Tavella, 1985). The declared aim was to provide some elementary notions on topics such as programming languages and methods, algorithm development, modelling of situations, use of correct and not ambiguous language. These notions were considered important both for a basic knowledge of the discipline and for the possibilities they offer in the teaching of other more traditional disciplines like mathematics (Ralston, 1981).

The evolution of hardware and software, that made computer interaction ever easier, and the parallel evolution of cognitive and pedagogical frameworks led to a gradual shift of interest, apart from specialized education, from the integration of informatics elements in school curricula to the implementation of new ICT-based educational applications and to the development and use of computer-linked methods, contents and tools for transforming and improving teaching and learning processes (Bottino, 2004). Technology design and use was progressively considered in relation to the whole teaching and learning process where a crucial role is assigned not only to the tool but also to the definition of meaningful practices through which technology can be used effectively to reach specific learning goals (Bottino, Ott, Tavella, 2011).

This line of evolution does not mean that research studies in the educational value of programming completely disappeared. They remain in the framework of constructivist approaches and mainly imply the development and use of educational specifically targeted languages, microworlds and programmable construction kits in the Logo tradition. Programming is often associated

with writing code; for Papert it is a way of thinking. Papert (1980) stressed the importance of supporting active thinking on the part of learners by means of programming concrete objects that can provide immediate feedback and concepts reification. Knowledge emerges as a result of an active engagement with the world through the creation and manipulation of artefacts that are seen as objects to think with. The Logo turtle, both the virtual screen version and the robotic one, is the most famous example of a computational tool to "think with" applied to differential geometry, a powerful mathematical idea that Logo makes concrete and accessible to children.

In the course of time the interest in the educational value of informatics never disappears, currently such interest has expanded in the presence of a wide debate on the foundations of informatics and on the definition of the informatics skills to be included in basic education. "Computational thinking" emerges as the main keyword which is now broadly considered to underline informatics core skills.

2 Computational Thinking

"Computational Thinking" is the title of a "viewpoint" published in the Communications of the ACM in March 2006 by Jeanette Wing (2006). The article argues that "computational thinking is a fundamental skill for everyone, not just for computer scientists" and "we should add computational thinking to every child's analytical ability" (p. 33). The article has stimulated a lively international debate and reflections of prestigious institutions. For example, the USA National Research Council has organized two workshops and published the related reports: the first on computational thinking (National Research Council, 2010) and the second on its pedagogical implications (National Research Council, 2011). The reports, however, also document a failure: the participants did not reach a consensus on the definition of computational thinking. Subsequently, Aho (2012) and Wing (2011) did propose slightly different, but equivalent, definitions. Wing's version is: "the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms".

The debate around these concepts is active in European countries as well: the Royal Society (2012), for example, published the report "Shut down or restart? The way forward for computing in UK"; the Académie des Sciences intervened on this subject with the report (2013) "L'enseignement de l'informatique en France – Il est urgent de ne plus attendre". Moreover, Informatics Europe and the ACM Europe Working Group on Informatics Education

(2013) urged Europe "not to miss the boat" on this subject. All reports call for a change in the curricula to make room for informatics.

The history of the relationship between informatics and education points out that attempt to add a new discipline to an already crowded K-12 curriculum presents problems and furthermore it is not necessarily a warrant for computational thinking to be really mastered in way suitable to be universally applied. Introducing computational thinking into schools requires special attention to contents and levels. In terms of content, subjects like science, technology, engineering and math (STEM) seem to offer a direct match with informatics key ideas. However, other subjects can be considered as well. For example, constructing video games and interactive storytelling are activities related to media art that can offer interesting connections with computational thinking. In the following, we will consider educational robotics and digital games as two exemplary contexts that are actively contributing to introduce computational thinking in education.

Robotic construction kits, such as the LEGO Mindstorms or Lilypad Arduino (Buechley, Qiu, Goldfein, de Boer, 2013) have revitalized the idea of end-user programming. Computational textile kits, like Lilypad Arduino, address the gender gap by contextualizing computational thinking into digital versions of arts and crafts activities such as sewing. The construction of an autonomous robot or an interactive garment encompasses both the physical assembly of the artefact and programming the rules that control the artefact behaviour. Nowadays, defining robot behaviour is facilitated by the design of the kit: sensor and actuators are "smart", i.e. their hardware contains intelligent components that reduce the complexity of the programming tasks. Furthermore, modern end-user programming environments featuring visual and tangible interfaces support the learner in writing a program by providing a structured context and powerful primitives while taking care of the syntactical aspects of coding. Robotic construction kits endowed with rule based visual and tangible programming environments can enable four-six years old children to program the behaviour of their robots (Bers, Flannery, Kazakoff, Sullivan, 2014 and Chioccariello, Manca, Sarti, 2004). These kits provide opportunities for children to explore issues of control and enable them to build and play with things that act as if they had a will of their own. Robotic construction kits are microworlds that well exemplify some important concepts that are usually mentioned when reflecting on the educational value of informatics, for example, to get in touch with powerful ideas such as feedback and emergent behaviours.

Digital game-based learning is a novel approach in the area of education and lifelong learning, displaying great potential as an active form of knowledge creation. Since games are now available in different platforms and devices, such as mobile devices, they offer the possibility to take learning outside the classroom and can provide a fun and interesting way of learning anytime, anywhere. Learning by playing is probably the ideal condition of education. One can play by the rules or with the rules, in the sense of building a new game. Video games are largely an example of playing by the rules, but games creation too can became a very valuable educational activity, able to trigger students' transversal skills, such as reasoning abilities, creative attitudes and digital competences. Specific environments to support games building activities have began to appear on the market and there is an increasing interest in their educational use. Kodu is an innovative environment for the creation of video games inspired by robot behaviour programming (Coy, 2013). Scratch, a visual programming environment where the instructions are assembled like LEGO building blocks (Resnick et al., 2009), is another popular environment for building games and interactive stories.

The construction of artefacts, robots or video games alone does not guarantee for learning. Computational activities should be embedded into an environment fostering collaboration, discussion, and reflection. Traditionally collaboration and discussion activities take place in classrooms where the teacher plays the role of mediator. How-ever, learning activities are increasingly developing also outside the school. Internet provides the condition for the birth of online social communities that involve learners of all ages. On-line communities of practice involving video games and robot con-structions are part of this movement (Kafai and Burke, 2013). They take advantage of national and international competitions where the practitioners meet (e.g. FIRST Lego League, National STEM Video Game Challenge, Robocup junior).

3 Conclusions

We advocate that learners should practice computational thinking in playful contexts where they can develop personal projects, for example building videogames and/or robots, share and discuss their construction with others. In our view, this approach allows, in principle, the integration of computational thinking in compulsory school curriculum. Let us outline a possible progression of playing, creating and exploring with programmable play kits leading to learning the mathematic and physic of motion. The math and physics of motion are usually included in secondary school programs. Computer simulations and

computer-based laboratory are often used in science classes. Developing models of phenomena is advocated, but difficult to pursue due to the complexity of dealing with the required math and physics concepts. Computational model, unlike the corresponding math representations, are executable model that can be more easily tested, debugged and refined. Research in the use of computational modeling in science education provides evidence that this approach is more learnable. However, familiarity with computational thinking skills and concepts is required to pursue this approach.

Objects falling, colliding, in equilibrium are phenomena that interest children from an early age; they develop sophisticated way of thinking trough experimenting with the physical world. Learning to interpret the same phenomena according to physical laws, however, requires a conceptual change mediated by education. Computers provide children with the opportunity to play with moving objects, but also to create their own moving objects, and eventually computational models of motion. At an early age, for example, they might construct cartoon like animations of a flying bird in a visual programming environment like Scratch. Later on they might add a jumping behavior to the character of a video game they are building. Computational models of projectile motion, like the ones in the popular Angry Birds video game, are within the reach of lower secondary school learners (DiSessa, 2000).

Learners should engage in minds-on and hands-on activities to understand science. Robotic kits allow the design and construction of tangible autonomous artifacts moving and interacting with the environment. Robotic projects might start, at primary school, with a simple reactive construction and eventually, at secondary school, reach the complexity of constructing a robot that plays soccer and it is capable of kicking the ball in the goal of the opposing team, as in the Robocup junior competition. A longitudinal progression of the use of programmable play kit in formal and informal education might provide the necessary familiarity and competencies in computational thinking skills to innovate education across the curriculum.

References

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55 (7), 832–835. doi: 10.1093/comjnl/bxs074

Académie des sciences (2013). *L'enseignement de l'informatique en France - Il est urgent de ne plus attendre*. Retrieved from http://www.academie-sciences.fr/activite/rapport/rads_0513.pdf

Bers, M. U., Flannery, L., Kazakoff, E. R., Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157. doi: 10.1016/j.compedu.2013.10.020

Bottino, R. M. (2004). The evolution of ICT-based learning environments: which perspectives for the school of the future?, *British Journal of Educational Technology*, 35 (5), 553–567. doi: 10.1111/j.0007-1013.2004.00413.x

Bottino, R. M., Ott, M., Tavella, M. (2011). Scaffolding Pedagogical Planning and the Design of Learning Activities: An On-Line System. *International Journal of Knowledge Society Research* (IJKSR), 2(1), 84–97. doi:10.4018/jksr.2011010107

Buechley, L., Qiu, K., Goldfein, J., de Boer, S. (2013). *Sew Electric: A Collection of DIY Projects That Combine Fabric, Electronics, and Sewing*. HLT Press.

Chioccariello, A., Manca, S., Sarti, L. (2004). Children's playful learning with a robotic construction kit. In Siraj-Blatchford, J. (Ed), *Developing New Technologies for Young Children* (93–112). Trentham Books.

Coy S. (2013). Kodu game lab, a few lessons learned. XRDS: *Crossroads. The ACM Magazine for Students*, 19 (4), 44–47. doi: 10.1145/2460436.2460450

DiSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Mit Press.

Kafai Y. B., Burke Q. (2013, March). The social turn in K-12 programming: moving from com-putational thinking to computational participation. In *Proceeding of the 44th ACM technical symposium on computer science education* (603–608). ACM. doi: 10.1145/2445196.2445373

Informatics Europe & ACM Europe (2013). *Informatics education: Europe cannot afford to miss the boat*. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education. Retrieved from http://europe.acm.org/iereport/ACMandIEreport.pdf

National Research Council (2010). Committee for the Workshops on Computational Thinking: *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.

National Research Council (2011). Committee for the Workshops on Computational Thinking: *Report of a workshop of pedagogical aspects of computational thinking*. Washington, DC: National Academies Press.

Olimpo, G., Persico, D., Sarti, L., Tavella, M. (1985). An experiment in introducing the basic concepts of informatics. In *Proceedings of the Fourth World Conference on Computers in Education*, WCCE'85, 31–38.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Ralston, A. (1981). Computer science, mathematics and the undergraduate curricula in both. *American Mathematical Monthly*, 88(7), 472–485.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52 (11), 60–67. doi: 10.1145/1592761.1592779

The Royal Society (2012). *Shut down or restart? The way forward for computing in UK schools*. Retrieved from http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf

Wegerif, R. (2002). *Literature Review in Thinking Skills, Technology and Learning*. Bristol Futurelab. Retrieved from http://archive.futurelab.org.uk/resources/documents/lit_reviews/Thinking_Skills_Review.pdf

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. doi: 10.1145/1118178.1118215

Wing, J. (2011). Research notebook: Computational thinking – What and why? *The Link Magazine*, Spring. Carnegie Mellon University. Pittsburgh. Retrieved from http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Biographies

Rosa Bottino is director of the Istituto per le Tecnologie Didattiche at the Consiglio Nazionale delle Ricerche. She promoted and chaired both national and European projects and Networks of Exellence in Technology Enhanced Learning. She is member of international research associations and journals editorial boards. Dr. Bottino received international awards like IFIP Silver Core Award and IFIP Outstanding Services Award.

Augusto Chioccariello is a researcher of the Istituto per le Tecnologie Didattiche at the Consiglio Nazionale delle Ricerche since 1986. Dr. Chioccariello completed his Physics Degree (in 1980 at the University of Naples. From 1982 to 1986, he worked in physics education at the Educational Technology Centre, UC Irvine. In 1996, he was awarded a NATO senior researcher fellowship at MIT Media Lab with Prof. Sey-mour Papert.

Copyright

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/3.0/