Behaviors that Computational Thinking involves

JAVIER BILBAO, OLATZ GARCÍA, CAROLINA REBOLLAR, EUGENIO BRAVO,
CONCEPCIÓN VARELA,
Applied Mathematics Department
University of the Basque Country, UPV/EHU
Engineering School, Alda, Urkijo, s/n, 48013 - Bilbao
SPAIN
javier.bilbao@ehu.es olatz.garcia@ehu.es carolina.rebollar@ehu.es eugenio.bravo@ehu.es
concepcion.varela@ehu.es

Abstract: - In this time where the ubiquity of computation is more and more notorious, Computational Thinking is called to play a very important role in Education. Perhaps because its very short history, since we can say that it started its path in education when J. Wing presented her idea in 2006, there are various point of view about what Computational Thinking is and what it involves. In anyway, Computational thinking can be a great help in using new technologies and how to apply them in several fields of the knowledge. It is a new and fundamental way of thinking and problem solving, described as a way for solving problems, designing systems and understanding human behavior by drawing on the concepts fundamental to computer science. Some fundamental concepts of Computational Thinking that are recognized by all different currents of thinking are the abstraction, algorithm design, data collection, decomposition and pattern recognition.

Key-Words: - Computational thinking, computer ubiquity, digital competence, education, learning, teaching, skills.

1 Introduction

Trends of using programmed devices, from computers to small smart watches or just drones or 3D printers, are changing year by year because the fast speed of the evolution of the technology. Since the last decade, the use of electronic devices has increased exponentially and nobody imagine the future without the presence of computers and microprocessors in their lives. We use them for working, studying, sports, social life, etc. It is the ubiquity of computers and microprocessors in our lives. Almost all fields of innovation are related to computing in some way. And the necessity of knowledge in computing is essential in the global economy market, being a basic tool for competition in the majority of jobs. And this implies that education systems have to include some concepts and skills in the curricula.

Engineering educators today are facing a number of challenges [1]. Not only in Engineering, but in all kind of disciplines, from Science to Humanities. But it is more notorious in STEM subjects, that is, Science, Technology, Engineering and Mathematics. Students entering engineering courses are less skilled in STEM subjects [2] despite being brought up in an environment with all forms of computational and electronic devices. A key challenge faced by engineering educators is to prepare students with the necessary skills and knowledge to work in multidisciplinary design teams upon graduation, solving complex problems using computational tools [3].

Nowadays in the majority of countries it is not strange to have a computer that students usually use to, among different other purposes, to help in their studies. Mark Weiser said in the early 1990s that ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives [4, 5]. Technologists had seen a dramatic shift in computing from many-to-one environment of mainframes to the one-to-one relationship of the personal computer. Drawing this trend out, Weiser foresaw the emergence of a world where one person would interact seamlessly with many computers—a development that he believed would lead to the age of ubiquitous computing. Two decades later, some in the ubiquitous computing community point to the pervasiveness of microprocessors as a realization of this dream. Without a doubt, many of the objects we interact with on a daily basis are digitally augmented [6].

But in spite of being microprocessors in all places and for all uses, before computers can be used to solve a problem, the problem itself and the ways in which it could be resolved must be understood. Computational Thinking (CT) techniques help with these tasks. Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. It includes a range of mental tools that reflect the breadth of the field of computer science.

2 Computation is an increasingly essential tool for doing scientific research

Computational thinking can provide some abilities that are not exclusive of people who will work in jobs related to Computer Science, but for ant type of job, even for any type of person: worker, student, unemployed, retired… Computational thinking is strategically important for dealing with many kinds of problems, and can be especially useful in the STEM subjects (science, technology, engineering and mathematics), where models, simulation, experiments are primary learning asset. But the abilities that computational thinking provides are not just for use in these scientific-technological subjects but in anyone, such as music, languages, politics, etc.

Computation is an increasingly essential tool for doing scientific research. It is expected that future generations of engineers will need to engage and understand computing in order to work effectively with computational systems, technologies and methodologies. CT is a type of analytical thinking that employs mathematical and engineering thinking to understand and solve complex problems within the constraints of the real world. The term was first used by S. Papert [7], who is widely known for the development of the Logo software. However, it was brought to the forefront of the computer society by Wing [8] to describe how to think like a computer scientist. She described CT as “solving problems, designing systems and understanding human behavior by drawing on the concepts fundamental to computer science”.

3 What is computational thinking?

According to Liu and Wang [9] computational thinking is a hybrid of other modes of thinking, like abstract thinking, logical thinking, modelling thinking, and constructive thinking:

In order to understand the main body of computer problem, abstract thinking is essential in computer science and technology. In solving an interesting problem, abstraction of thinking is one very general purpose heuristic that can help to attack this problem. Informally, abstraction thinking can be thought of the mapping from a ground representation to a new but simpler representation.

Logical thinking is the process in which one uses reasoning consistency to come to a conclusion. Some computer problems or computer states (situations) involving logical thinking always call for mathematics structure, for relationships between some hypotheses and given statements, and for a sequence of reasoning that makes the conclusion more reasonable.

Modelling thinking, in the technical use of the term, refers to the translation of objects or phenomena from the real world into mathematical equations (mathematical models) or computer relations (simulation models). It is choosing an appropriate representation or modelling the relevant aspects of a problem to make it tractable. Computer modelling is the representation of reality objects on a computer. A problem which will be solved by computer must be modelled by a corresponding software model.

Constructive thinking is any well-defined computational procedure that takes some value, or set of values as input and produces some value, or set of values as output.

The main characteristics of CT include:

- Analyzing and logically organizing data.
- Data modeling, data abstractions, and simulations.
- Formulating problems such that computers may assist.
- Identifying, testing, and implementing possible solutions.
- Automating solutions via algorithmic thinking.
- Generalizing and applying this process to other problems.

4 Inclusion of computational thinking in education

Computer Science and ICT (Information and Communication Technology) are generally recognized as very important issues at all levels of Education. Digital Agenda for Europe (European

Commission, 2010a) includes them as Pillar VII “ICT-enabled benefits for EU society”.

In 2006 the European Parliament and the Council [10] published a recommendation identifying eight Key Competences for Lifelong Learning: Communication in the Mother Tongue; Communication in Foreign Languages; Mathematical Competence and Basic Competences in Science and Technology; Digital Competence; Learning to Learn; Social and Civic Competences; Entrepreneurship; and Cultural Awareness and Expression. Four years afterwards, the value of this recommendation is recognized in the Europe 2020 Strategy [11].

The 2006 recommendation already points to Digital Competence as a fundamental basic skill. Digital Competence is there defined as follows:

"Digital Competence involves the confident and critical use of Information Society Technology (IST) for work, leisure and communication. It is underpinned by basic skills in ICT: the use of computers to retrieve, assess, store, produce, present and exchange information, and to communicate and participate in collaborative networks via the Internet." [10].

The recommendation provides explanation on the essential knowledge, skills and attitudes needed to be digitally competent. The foreseen knowledge includes the understanding of the functioning of main computer applications; of the risks of the Internet and online communication; of the role of technologies in supporting creativity and innovation; of the validity and reliability of online information; of the legal and ethic principles behind the use of collaborative tools [12].

The needed skills are seen as the ability to manage information; the capacity to distinguish the virtual from the real world and to see the connections between these two domains; the ability to use Internet-based services and to use technologies to support critical thinking, creativity and innovation.

In terms of attitudes, the recommendation gauges as essential that citizens are critical and reflective towards information, that they are responsible users and interested in engaging in online communities and networks.

Different reports and research papers argue about promoting the inclusion of Computational Thinking (CT) in education and the pervasiveness of technologies, which leads to the subsequent need to acquire Digital Competence to be functional in our knowledge society; digital inclusion depends more on knowledge and skills than on access and use [13]. In a similar direction, digital ´rhetoric´ discourses claim the necessity to develop digital literacy for full participation in life [14], while policy documents often emphasize the need to invest in digital skills enhancement for economic growth and competitiveness [13, 15]. Computer-related proficiency, according to yet another digital rhetoric strand, is the key to employability and improved life chances [14].

According to OECD [16], “education standards need to include the kind of skills and competences that can help students become responsible and performing users of technology and to develop the new competences required in today’s economy and society which are enhanced by technology, in particular those related to knowledge management”. In the referred report, these skills were defined to include processes related to knowledge management in network environments. Moreover, it stated that these skills should be gained at school. Such a broad definition leaves open the question about in which specific subject domains or on which school levels the elements of digital competence should be taught.

One of the few papers that provide some answers to the question is Erstad´s [17]. He broadened digital literacy to media literacy and suggested the following aspects of media literacies as part of school-based learning: 1) Basic skills, 2) Media as an object of analysis, 3) Knowledge building in subject-domains, 4) Learning strategies, and 5) Digital Bildung / Cultural competence. Besides this, Erstad emphasized the user-generated content creation (Web2.0, editing software) in which students have an active role in knowledge practices.

5 Skills, attitudes and concepts related to computational thinking

In 2006, Jeannette Wing wrote an article for the Communications of the ACM where she used the term “computational thinking” to articulate a vision that everyone, not just those who major in computer science, can benefit from thinking like a computer scientist [8]. More recently, Cuny, Snyder and the own Wing defined this term: Computational thinking is the thought processes 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.

Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines.

The interpretation of the words "problem" and "solution" is broad. Usually they mean not just mathematically well-defined problems whose solutions are completely analyzable, e.g., a proof, an algorithm, or a program, but also real-world problems whose solutions might be in the form of large, complex software systems. Thus, computational thinking overlaps with logical thinking and systems thinking. It includes algorithmic thinking and parallel thinking, which in turn engage other kinds of thought processes, such as compositional reasoning, pattern matching, procedural thinking, and recursive thinking. Computational thinking is used in the design and analysis of problems and their solutions, broadly interpreted.

While the biggest growth in our 21st century job market is for workers with CT skills, our schools currently produce less than one third the number of qualified applicants [18]. This lack of quality curricula is particularly true for schools dominated by students with low socioeconomic status [19, 20]. Moreover, research has shown that female and minority students are dissuaded from pursuing CS/STEM education and careers [21, 22, 23, 24]. Dominant groups benefit from preparatory privilege (resources and support at home and in school; as well as cultural expectations), which make for an unlevel playing field [21]. As a result of conscious and unconscious biases, women and minorities are placed into low-level and feeder CS courses in high school, known for disengaging students [21]. Moses & Cobb [25] label this the civil rights issue of the 21st century, with people of low SES, women, and minorities being the “designated serfs of the information age”.

These factors have led to the current situation in which only, for some countries, 11% of bachelor degrees and 22% of master degrees in Computer Science are awarded to women, according to the Taulbee Survey.

We believe that a very promising strategy for addressing the many challenges described above is to embed computational thinking activities in traditional STEM courses.

ISTE and CSTA collaborated with leaders from higher education, industry, and K–12 education to develop an operational definition of CT. The operational definition provides a framework and vocabulary for CT that will resonate with all K–12 educators [31].

Finally, we resume CT in three groups that define this new way of thinking: skills that CT promotes, attitudes that are supported, and concepts of computational thinking.

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 organizing and analyzing data.
- Representing data through abstractions such as models and simulations.
- Automating solutions through algorithmic thinking (a series of ordered steps).
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources.
- Generalizing and transferring this problem-solving process to a wide variety of problems.

These skills are supported and enhanced by a number of dispositions or attitudes that are essential dimensions of CT. These dispositions or attitudes include:

- Confidence in dealing with complexity.
- Persistence in working with difficult problems.
- Tolerance for ambiguity.
- The ability to deal with open-ended problems.
- The ability to communicate and work with others to achieve a common goal or solution.

Concepts of computational thinking that implicit in the operational definition are the following:

- Data Collection: The process of gathering appropriate information
- Data Analysis: Making sense of data, finding patterns, and drawing conclusions
- Data Representation: Depicting and organizing data in appropriate graphs, charts, words, or images
- Problem Decomposition: Breaking down tasks into smaller, manageable parts
- Abstraction: Reducing complexity to define main idea
- Algorithms and Procedures: Series of ordered steps taken to solve a problem or achieve some end.
- Automation: Having computers or machines do repetitive or tedious tasks.
- Simulation: Representation or model of a process. Simulation also involves running experiments using models.

- Parallelization: Organize resources to simultaneously carry out tasks to reach a common goal.

The Computer Science Teachers Associations (CSTA) and the International Society for Technology in Education (ISTE) developed a Computational Thinking Concepts Guide in order to help teachers to incorporate the concepts of Computational Thinking into existing lesson plans, projects, and demonstrations of any disciplinary subject.

Concepts are followed by definitions and some teaching examples:

Abstraction is identifying and extracting relevant information to define main idea(s). For example, show how a daily planner uses abstraction to represent a week in terms of days and hours, helping us to organize our time

Algorithm Design is creating an ordered series of instructions for solving similar problems or for doing a task. One example is how in mathematics, when we add and subtract fractions with different denominators, we follow an algorithm. Another more common example is when a chef writes a recipe for a dish, she is creating an algorithm that others can follow to replicate the dish.

Automation is having computers or machines do repetitive tasks. Automation can be used to perform tasks that would take a very long time to complete using a manual process, such as identifying the migration patterns of a specific demographic based on census data.

Data Collection is gathering information, like when students collect the birthday and gender of their peers and record it in a spreadsheet

Data Analysis is making sense of data by finding patterns or developing insights. If we have previously collected some data sets with anonymized personal data, such as height, shoe size, favorite color, etc., data analysis can be used to highlight information that is meaningful and relevant to students.

Data Representation is depicting and organizing data in appropriate graphs, charts, words, or images. Data can be plotted manually on the whiteboard or via projector so that students can see the process of how the organization unfolds.

Decomposition is breaking down data, processes, or problems into smaller, manageable parts. In mathematics, we can decompose a number such as 256.37 as follows: 2*102+5*101+6*100+3*10-1+7*10-2; and in science we decompose a projectile's velocity into its components along the x- and y-axis.

Parallelization is simultaneous processing of smaller tasks from a larger task to more efficiently reach a common goal. An example of parallelization in computing is when a single task (such as the analysis of a DNA sequence) is broken into smaller tasks and simultaneously analyzed by different computers so that the analysis can be processed more efficiently.

Pattern Generalization is creating models, rules, principles, or theories of observed patterns to test predicted outcomes. In mathematics, we write generalized formulas in terms of variables instead of numbers so that we can use them to solve problems involving different values. For example, the slope of any straight line can be described as a function of $y = mx + b$.

Pattern Recognition is observing patterns, trends, and regularities in data. For example, to identify trends in stock price cycles that may suggest when clients should be bought and sold.

Simulation is developing a model to imitate real-world processes. One example is to illustrate the movement of a solar system by modeling the gravitationally curved path of an object around a point in space.

6 CT and STEM

STEM can enrich computational learning. Research has also shown that the reverse is true; the use of computational tools has been shown to enable deeper learning of STEM content areas for students [26, 27, 28, 29, 30]. The use of computational tools and computational thinking skills can deepen the learning of STEM content.

A final motivation for bringing CT into STEM classrooms is to better prepare students for the modern landscape of the STEM disciplines. Computation is an indispensable component of STEM disciplines as they are practiced in the professional world. In the last twenty years, nearly every STEM field has seen the birth or reconceptualization of a computational counterpart, from Computational Engineering and Bioinformatics to Chemometrics and Neuroinformatics. The appearance of “data-driven” computational fields has a rich history intertwined with statistics and dynamic systems theory, but recent advances in high-speed computation and analytical methods have created some of the most powerful tools in understanding phenomena across all spectrum of human inquiry. The 1998 Nobel Prize in Chemistry was awarded to John A. Pople and Walter Kohn for their innovative work in the development of computational methods quantum chemistry. Such a prestigious award hailed the acceptance of computation as a valid and rigorous tool for investigating chemical phenomena. Across research laboratories, engineering and design firms, medical practices, and beyond, computational tools are at the heart of what it means to be a STEM practitioner in the 21st century. Bringing these computational tools and practices into the classrooms gives learners a more realistic view of what STEM fields are as well as better prepares students for STEM careers, a major goal of STEM education.

7 Conclusion

Our society is involved in a very fast changing world from the point of view of the technology. This change must be reflected into the educational system by means not in a development of a new branch or new subjects or changing some themes but in a different manner for teaching the concepts of the subjects.

Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions. We can then present these solutions in a way that a computer, a human, or both, can understand.

Almost all fields of innovation are related to computing in some way. Therefore, the necessity of knowledge in computing is essential in the global economy market, being a basic tool for competition in the majority of jobs. Taking into account this situation, Computational Thinking is a new and fundamental way of thinking and problem solving, described as a way for solving problems, designing systems and understanding human behavior by drawing on the concepts fundamental to computer science.

Four main characteristics of the Computational Thinking, abstraction, decomposition, pattern recognition and algorithm design, can be use in all subjects at school or university, and also in industry, services or other sectors of the economy.

References:

[1] J.E. Mills, D.F. Treagust, Engineering Education – Is a problem-based or project-based learning the answer?, *Australian Journal of Engineering Education*, 2003.

[2] House of Lords, Select Committee on Science and Technology, *Higher Education in Science, Technology, Engineering and Mathematics (STEM) subjects*, 2nd report of session 2012-2013, 2012.

[3] C. Mohtadi, M. Kim, J. Schlosser, *Why integrate computational thinking into a 21st century engineering curriculum?* 41st SEFI Conference, 16-20 September 2013, Leuven, Belgium, 2013.

[4] M. Weiser, The Computer for the Twenty-First Century, *Scientific American*, Sept. 1991, pp. 94-10.

[5] M. Weiser, Hot Topics: Ubiquitous Computing, *Computer*, Oct. 1993

[6] Chris Harrison, Jason Wiese, and Anind K. Dey, Achieving Ubiquity: The New Third Wave, *Media Impact*, July-September 2010.

[7] Papert, S., An exploration in the space of Mathematics Education, *International Journal of Computers for Mathematics*, Vol. 1, No. 1, 1996, pp. 95-123.

[8] Wing, J. M. Computational thinking. *Communications of the ACM*, 49 (3), 2006, pp. 33-35.

[9] Liu, J. & Wang, L., Computational Thinking in Discrete Mathematics, *IEEE 2nd International Workshop on Education Technology and Computer Science*, 2010, pp. 413-416.

[10] European Parliament and the Council. (2006). Recommendation of the European Parliament and of the Council of 18 December 2006 on key competences for lifelong learning. *Official Journal of the European Union*, L394/310, 2006.

[11] European Commission. Europe 2020: A strategy for smart, sustainable and inclusive growth, *COM* 2020, 2010.

[12] Ferrari, A. Digital Competence in Practice: An Analysis of Frameworks. *JRC Technical Reports*, European Commission, 2012.

[13] Eshet-Alkalai, Y. Digital Literacy. A Conceptual Framework for Survival Skills in the Digital Era. Journal of Educational Multimedia & Hypermedia, 13 (1), 2004, pp. 93-106.

[14] Sefton-Green, J., Nixon, H., & Erstad, O. Reviewing Approaches and Perspectives on

"Digital Literacy". *Pedagogies: An International Journal*, 4, 2009, pp. 107-125.

[15] Hartley, J., Montgomery, M., & Brennan, M. *Communication, cultural and media studies: The key concepts*. Psychology Press, 2002.

[16] OECD. Are the New Milleniums Learners Making the Grade? Technology use and educational performance in PISA. OECD, 2010.

[17] Erstad, O. Educating the Digital Generation. *Nordic Journal of Digital Literacy*, 1, 2010, pp. 56-70.

[18] Levy, F. & Murname, R. *The new division of labor: How computers are creating the new job market*. Princeton, NJ: Princeton University Press, 2004.

[19] Warschauer, M. Technology and School Reform: A view from both sides of the track. *Educational Policy Analysis Archives* 8 (4), 2000.

[20] Warschauer, M. *Laptops and literacy: Learning in the wireless classroom*. New York: Teachers College Press, 2006

[21] Margolis, J., Estrella, R., Goode, J., Jellison Home, J., Nao, K. *Stuck in the shallow end: Education, race, and computing*. Cambridge Mass: MIT Press, 2008.

[22] Kozol, J. *Savage Inequalities*. New York: Harper Perennial, 1992.

[23] Kao, G. Group images and possible selves among adolescents: Linking stereotypes to expectations by race and ethnicity. *Sociological Forum* 15 (3), 2000, pp. 407-430.

[24] Noguera, P. *City Schools and the American Dream*. New York: Teachers College Press, 2003.

[25] Moses, R., & Cobb, C. *Radical equations: Civil rights from Mississippi to the Algebra Project*. Boston: Beacon Press, 2002.

[26] Guzdial, M. Software---realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments*, 4(1), 1994, 001–044. doi:10.1080/1049482940040101

[27] National Research Council. *Report of a Workshop of Pedagogical Aspects of Computational Thinking*. Washington, D.C.: The National Academies Press, 2011.

[28] Repenning, A., Webb, D., & Ioannidou, A. Scalable game design and the development of a checklist for getting computational thinking into public schools. In Proceedings of the *41st ACM technical symposium on Computer science education*, 2010, pp. 265–269.

[29] Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 2013, pp. 1–30.

[30] Wilensky, U., & Reisman, K. Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories— an embodied modeling approach. *Cognition and Instruction*, 24 (2), 2006, pp. 171–209.

[31] Computer Science Teachers Associations (CSTA) and International Society for Technology in Education (ISTE). Computational thinking teacher resources (2nd ed.), 2011, http://www.csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf