Robotics and Autonomous Systems 75 (2016) 661–670

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences

Soumela Atmatzidou *, Stavros Demetriadis

Aristotle University of Thessaloniki, Department of Informatics, Greece

HIGHLIGHTS

- Students reach the same level of Computational Thinking (CT) skills development independent of their age and gender.
- Computational Thinking skills in most cases need time to fully develop (students' scores improve significantly towards the end of the activity).
- Girls appear in many situations to need more training time to reach the same skill level compared to boys.
- The different modality (written and oral) of the CT skill assessment instrument may have an impact on students' performance.

ARTICLE INFO

Article history:
Available online 30 October 2015

Keywords:
Computational thinking
Educational robotics
Secondary education
Skill development

ABSTRACT

This work investigates the development of students' computational thinking (CT) skills in the context of educational robotics (ER) learning activity. The study employs an appropriate CT model for operationalising and exploring students' CT skills development in two different age groups (15 and 18 years old) and across gender. 164 students of different education levels (Junior high: 89; High vocational: 75) engaged in ER learning activities (2 hours per week, 11 weeks totally) and their CT skills were evaluated at different phases during the activity, using different modality (written and oral) assessment tools. The results suggest that: (a) students reach eventually the same level of CT skills development independent of their age and gender, (b) CT skills in most cases need time to fully develop (students' scores improve significantly towards the end of the activity), (c) age and gender relevant differences appear when analysing students' score in the various specific dimensions of the CT skills model, (d) the modality of the skill assessment instrument may have an impact on students' performance, (e) girls appear in many situations to need more training time to reach the same skill level compared to boys.

1. Introduction

This work presents and discusses a specific didactic approach to support the development of students' computational thinking (CT) skills in educational robotics (ER) activities. As Wing [1] argues, computational thinking (CT) is a fundamental skill for everyone and it should be considered as an important component of every child's analytical ability along with reading, writing, and arithmetic. Recently, there has been growing recognition of the importance of CT in controlling and managing cognitive activities, as well as understanding and solving problems in a wide range of contexts, not only in the field of computer science, but in all disciplines [2].

Robotics can be used as a tool that offers opportunities for students to engage and develop computational thinking skills [3,4]. Educational robotics is being introduced in many schools as an innovative learning environment, enhancing and building higher order thinking skills and abilities, and helping students solve complex problems [5]. Furthermore, a guided instruction approach using robots facilitates teamwork, develops conceptual understanding, enhances critical thinking, and promotes higher-order learning in the domains of mathematics and science [6].

This paper describes the implementation of ER activity in secondary school, focusing on the different possible impacts that the instructional approach might have on the development of students' CT skills depending on their age and gender. Guided by worksheets, students worked in small groups to solve robot programming problems. The level of their CT skills was evaluated at different times during the activity, with focus on five key CT constructs—abstraction, generalisation, algorithm, modularity and decomposition.

* Corresponding author. Tel.: +30 6977718143.
E-mail addresses: atmatzid@csd.auth.gr (S. Atmatzidou), sdemetri@csd.auth.gr (S. Demetriadis).

http://dx.doi.org/10.1016/j.robot.2015.10.008
0921-8890/

2. Background

Robotics is usually seen as an interdisciplinary activity drawing mostly in Science, Mathematics, Informatics and Technology and offering major new benefits to education in general at all levels [7,8]. Educational robotics is a powerful, flexible, teaching and learning tool, encouraging students to construct and control robots using specific programming languages [7]. The roots of ER are to be found in Seymour Papert's work, creator of the Logo programming language [9]. Papert suggests that learning is most effective when students are experiencing and discovering things for themselves. He also argues that robotics activities have tremendous potential to improve classroom teaching [9,10]. Drawing on the theoretical underpinnings of Papert's constructionism and Vygotsky's socio-cognitive approaches, ER activities help students transform themselves from passive to active learners, constructing new knowledge by collaborating with their peers and developing essential mental skills by acting as researchers. Many studies indicate that ER activities have a positive impact on the development of students' critical thinking, problem solving and metacognitive skills [5,11,12] and also on the learning of a programming language [7,13,14]. Other studies demonstrate how ER promotes a joyful mode of learning, while advancing students' motivation, collaboration, self-confidence and creativity [15–17]. Many researchers argue that robotics programs provide a valuable avenue to increase students' interest and participation in science, technology, engineering and mathematics (STEM), while they motivate them to pursue a career in one of these fields (e.g. [18–20]). However, certain researchers point out that although robotics seems to be an excellent tool for teaching and learning and a compelling topic for students of all ages, the pedagogy of teaching with robotics is still in its infancy [7,21]. It is also noted that more research is needed to point out how to work with educational robotics to help students develop specific skills [10,22].

As this study focuses on ER as a means for advancing students' CT skills, we concisely review next the CT theoretical framework and studies on the ER-CT relationship. Wing [1] describes CT as a type of analytical thinking that draws on concepts fundamental to computer science and provides a way for solving problems, designing systems, and understanding human behaviour. CT roots go back to Papert's ideas of the computer being the children's machine that would allow them to develop procedural thinking through programming, and refers to ways of algorithmically solving problems and to the acquisition of technological fluency [9].

In the literature there are multiple definitions of CT and several suggestions about which skills and abilities are relevant to CT and how to integrate CT in the curricula of all grades. Wing [2] asserts that CT has the potential to advance the students' problem-solving skills through processes such as abstraction, generalisation, decomposition, algorithm design and separation of concerns. Astrachan et al. [23] emphasise skills such as: developing computational artefacts, abstracting, analysing problems and artefacts, and communicating and working effectively in teams. Still others argue that the key concepts of CT are abstraction, automation, simulation, evaluation, algorithm building, conditional logic, debugging, decomposition, problem analysis, distributed computing and effective teamwork [24–26]. Emphasis is also given to the view that the educational benefits of CT transfer to any domain – not only in the field of computer science – by enhancing and reinforcing intellectual skills [1,27]. Yadav [28] argues emphatically that '*CT in education has the potential to significantly advance the problem-solving skills of K-12 students*'.

Naturally, researchers have started exploring also the potential of educational robotics to promote the development of CT [4,29–31]. Certain studies emphasise that children who program robots learn and apply core CT concepts such as abstraction, automation, analysis, decomposition, modularisation and iterative design [4,29,30]. A 2011 study by National Science Foundation [4] provided evidence that student programmers in a robotics project, developed abstraction, automation, and analysis related skills, while programming the robot agent to interact with its environment. However, it is worth mentioning according to researchers that the field requires systematic assessment procedures.

Research engaging younger children reported also positive outcomes, demonstrating that children 4–6 years old can build simple robotics projects becoming acquainted with powerful ideas of engineering, technology, and computer programming while also building CT skills [30,32,33]. More specifically, a study with 53 kindergarten children [33] using Lego WeDo robots and the CHERP (Creative Hybrid Environment for Robotics Programming) language, reported that the children were involved and understood basic programming and CT concepts relevant to sequencing and choosing the correct instructions. A similar study by Kazakoff et al. with 27 kindergarten children, focusing solely on sequencing, showed improvement of the students' scores from the first activity to the last [29].

Regarding elder children (Junior and High School students), studies report also positive results on the development of CT skills. Grover [27] developed a curriculum for teaching CT Language and CT principles in schools. The results indicated that students after the intervention were capable of using certain CT related vocabulary and principles (such as conditional logic and decomposition), whereas other concepts like abstraction, representation and algorithmic flow control were seldom used. Another study by Touretzky et al. [34] engaging children aged 10–17 (some of them with special abilities), focused on abstraction across different programming environments and especially on deep and abstract understanding of programming concepts. The researchers concluded that – despite the limitations – robotics is a helpful tool for young students, "*facilitating a more abstract understanding*". Penmetcha [35] investigated the effects of ER activity on university students exploring the relationship between robotics and developing programming and algorithmic thinking. The results showed that robotics fulfil their purpose as a medium for incorporating CT practices, regardless of the students' background, and can be used to teach concepts such as designing, programming and testing at a more abstract level. As in the other studies, limitations were reported relevant to the study small sample size [27]. Finally, a case study by Eguchi [36] explores the effects of a robotics competition on students' CT and problem solving skills reporting an overall very positive effect.

Overall, although the CT concept has attracted considerable attention, the literature on implementing CT in a K-12 setting is still relatively sparse [28]. There is also lack of empirical evidence in defining the explicit CT boundaries [37], although recent articles begin to describe what it looks like [4,30,38,39]. More than that, research into how CT can be introduced in the classroom is on the early stages and there is shortage of description about how children can learn and develop CT skills [27,28,37]. Another issue is to understand at what age – or grade level – children are ready to be familiar with advanced concepts such as abstraction, automation, decomposition, etc. and how to teach those skills progressively [4]. Likewise, there is little agreement on strategies for assessing the development of CT in young people [23,38,40,41]. Existing studies typically employ a student group of specific age thus limiting the generalisation of the results to other age groups (e.g. [8,29,33]), have small sample sizes (e.g. [27,29,34,35]), and do not provide explicit teacher guidance on how to organise a well-guided ER activity to promote students' CT skills. Researchers also differ in the way they build an operational CT skills framework to apply to their studies. Table 1 presents the various CT skill models employed in various ER studies.

Another issue of interest is the gender differences observed in studies on STEM learning activities. Much research has

Table 1
CT skills models employed in various ER studies.

Article	Context	CT skills model
Lee et al. [4]	K-12	Abstraction, Automation, and Analysis.
Grover [27]	10 students mean age: 13	Computational Thinking Language (CTL) [42] Abstraction, Taskbreakdown, Conditional logic, Representation, Algorithm, and Debugging.
Penmetcha [35]	26 university students	Abstraction, Algorithm, Programming, and Designing.
Bers et al. [33]	53 kindergarten students	Sequencing
Kazakoff et al. [29]	27 kindergarten students	Sequencing
Touretzky et al. [34]	31 students aged 10–17	Abstraction between three software frameworks (Kodu, Alice, Lego NXT), Recognise fundamental programming concepts.
Bers et al. [33]	53 kindergarten students	Debugging, Correspondence, Sequencing, and Control flow.
Eguchi [36]	168 students aged 10–19	Problem Solving, Debugging, Prototyping, Decomposition, Logical thinking, Creating step-by-step procedure, Analysing Skills, Critical Thinking, Iteration, and Debugging.

documented gender differences, showing that men have higher levels of self-efficacy and higher probability of success in STEM-related fields (e.g., [43,44]). However, over the past few decades the gender gap has narrowed. The stereotypic gender role might have a clear impact on attitudes about technology but this can be positively changed under the right conditions [45,46]. Studies indicate that both genders can have a successful and rewarding experience being exposed to robotics activities. Milto et al. [47] found that although men were more confident in their abilities than women, in an introductory engineering class women and men displayed equivalent competency in robotics activities. Similarly, Nourbakhsh et al. [48] investigated the gender differences in a robotics course involving high school students. According to the study, although girls entered the course with less confidence than boys and were more likely to have struggled with programming, by the end of the course girls' confidence increased more than boys'. Another study by Cheng [49] reported that in terms of assembling and programming Lego robots, while there were slightly higher average scores for male students it was not of significant difference. However, research on comparing the development of CT skills between genders in K-12 robotics activities is relatively sparse. While research has been conducted on gender differences in many science and mathematics areas [50] limited research has been carried out into gender differences in robotics and programming achievement especially in early childhood [51].

Research motivation and key research question

Considering the above background, the current study aimed to conduct an instructional well-guided ER activity, recruit a relatively large student sample size and explore the impact of the activity on students' CT skills, comparing student groups of different age and gender. Thus, the overarching research question set by the study is: "Are students of different age and gender developing CT skills in the same way in the context of educational robotics activity?"

3. Method

3.1. Participants

For the purpose of our study we conducted a series of robotic training seminars in public schools in the area of Thessaloniki. In total, 164 students of two different school levels (Junior high and High vocational) participated in the study. Specifically, in the seminars were engaged:

- Junior high (J): 89, K-9 students (age: 15, 48 boys and 41 girls)
- High vocational (H): 75, K-12 students (age: 18, 64 boys and 11 girls).

3.2. A model for CT skills

To operationalise the CT theoretical approach, we focused on five core dimensions of the broader CT conceptual framework. These included: abstraction, generalisation, algorithm, modularity and decomposition. The proposed model encompasses skills that can easily emerge when students engage in educational robotics activities. In detail, the proposed model for CT skills presented in Table 2.

3.3. Implementation procedure

In total, we conducted 8 training robotics seminars (4 at Junior high and 4 at High vocational schools') during the 2012–2013 school year. The Lego Mindstorms NXT 2.0 educational robotics kit was used in all seminars. Organised and supervised by the main researchers (authors of this work), each seminar comprised 11 sessions (2 h each, conducted once a week). Trained postgraduate students ("trainers") assisted with the practicalities of the activity (e.g. organising student groups, handling out worksheets, encouraging and scaffolding teams, administering questionnaires, etc.). The seminars were conducted during the typical school time schedule and the class teachers remained in the classroom during the activity, simply helping to maintain the flow of each lesson. In detail, the sessions were as follows:

1st session: In the beginning, the teacher introduced robotics in general, the Lego Mindstorms NXT robot and the Lego NXT-G programming environment. Then she handed out the Profile Questionnaire (PQ) to be filled in individually by students. Working in groups, the students implemented their first program using their own robot kit. Emphasis was placed on the concept of algorithm and the importance of developing precise instructions that when implemented they lead to the solution of the problems.

2nd session: The objective here was students' familiarisation with some basic programming concepts (sequential structure and loop structure). The students also became familiar with the motors, the touch sensor, the sound sensor and finally with some basic feature of NXT, such us displaying images on the screen of the robot. In this session, students programmed their robots to dance and presented them to the other groups. The session placed focus on the abstraction and generalisation concepts. Participants were prompted to reflect on the role of these two concepts in their own problem solving activities.

3rd and 4th sessions: The students worked on the control structure and on how to use the ultrasonic sensor and the wait block. They also practised conversion of numbers to text in order to show a numerical value on the screen. In the last activity of the 4th session the challenge was to create a robotic alarm system that detects motion and sound. At the end of the fourth session we administered the first questionnaire (Q1) in order to assess the students' level of CT skills development. In 3rd session the focus here was on modularity and decomposition and their importance in optimising the structure of an algorithm implementation. From the 4th session onwards, the activities challenged the students to engage in practising all the concepts of the CT model and develop relevant skills.

5th and 6th sessions: The students became familiar with the operation of light sensor, the creation of reusable subprograms

Table 2
The CT skills model applied in the current study.

CT skills	Description	Student skills (The student should be able to...)
Abstraction	Abstraction is the process of creating something simple from something complicated, by leaving out the irrelevant details, finding the relevant patterns, and separating ideas from tangible details [52]. Wing [2] argues that the essence of CT is abstraction.	1. Separate the important from the redundant information. 2. Analyse and specify common behaviours or programming structures between different scripts. 3. Identify abstractions between different programming environments.
Generalisation	Generalisation is transferring a problem-solving process to a wide variety of problems [38].	Expand an existing solution in a given problem to cover more possibilities/cases.
Algorithm	Algorithm is a practice of writing step-by-step specific and explicit instructions for carrying out a process. Kazimoglu et al. [37] argue that selection of appropriate algorithmic techniques is a crucial part of CT.	1. Explicitly state the algorithm steps. 2. Identify different effective algorithms for a given problem. 3. Find the most efficient algorithm.
Modularity	Modularity is the development of autonomous processes that encapsulate a set of often used commands performing a specific function and might be used in the same or different problems [38].	Develop autonomous code sections for use in the same or different problems.
Decomposition	Decomposition is the process of breaking down problems into smaller parts that may be more easily solved. Wing [2] argues that CT is using decomposition when attacking or designing a large complex task.	Break down a problem into smaller/simpler parts that are easier to manage.

(make a new "My Block"), the use of the lamp block and the parallel programming. The students programmed a recycler robot, where the robot moves following a black line and sorts items to be recycled depending on their colours.

7th and 8th sessions: The students worked on the concept of variable and basic arithmetic operators. In this context, the students implemented a security guard robot that moves around a building and detects every motion, sound and change in lightness.

9th and 10th sessions: Students were given activities of increased difficulty to practise their developing CT skills in the context of more complex authentic problems, such as a car that moves following the traffic code, etc. The project allowed children to demonstrate the powerful ideas they learned over the previous sessions as well as to apply them and continue learning by solving a new problem. A second questionnaire (Q2) was administered at the end of this session to assess the students' current level of CT skills development.

11th session: In the final session, student groups were given the "final challenge" that is a demanding robot programming task for groups to compete against each other. The winner was the group that proposed an effective and efficient task solution (optimised code and fastest solution).

After the completion of each seminar, two other instruments were used to capture the students' level of CT skills and also their views regarding the ER training experience. These were: (a) a 'think-aloud' protocol implementation, (b) a student's opinion questionnaire. Overall, the procedure of each training seminar and the various data collection instruments are presented in Fig. 1.

3.4. Didactic model

In each seminar, students worked in groups of three (or four if necessary) and were guided by worksheets in the investigating robot programming tasks of gradually increased complexity. These enabled them to start constructing understanding and developing the CT skills prescribed by our model. The worksheets also directed students to assume the roles of analyst (analyse the problem), algorithm designer (describe the algorithm), programmer (write the code), or debugger/evaluator (review and assess the solution). The students exchanged roles successively as the activities evolved.

During the sessions, the trainers acted as facilitators to scaffold students while solving programming tasks. After the 5th session, trainers gradually faded their support. This means that detailed guidance was gradually replaced by simple prompts to students to assume the relevant role and practise the acquired skills on their own capacity [5]. The trainers were ready to fade-in again and support students should the circumstances require it. To trigger students' reflection and development of CT modelled skills, prompts such as the following (see Table 3) were included in the worksheets. Peers were expected to spend some time discussing how to answer these prompts; then one peer was assigned the responsibility of writing down the group answer to the worksheet.

3.5. Measures, instruments and data analysis

The instruments that we used to collect evaluation data (and respective measures) are as follows:

Profile Questionnaire (PQ): An individual questionnaire was administered in the beginning of each seminar. The questionnaire recorded some simple demographic data (e.g. student gender), the students' background on computer use (for example, frequency of computer use, computer experience, etc.) and experience with robotics (such as previous knowledge on constructing and programming robots).

Two intermediate Questionnaires (Q1 and Q2): Q1 was handed out after the 4th and Q2 after the 10th session. Both questionnaires asked students to solve programming problems and practise CT skills during their solution process; for example, identify common programming structures that guide robot behaviour in two tasks (abstraction), propose a more general solution (generalisation), describe step-by-step the solution process (algorithm), etc. The assessment of students' answers in Q1 and Q2 was based on a graded criterion instrument (rubric) using a 4-point Likert scale (1 = 'unsatisfactory', 2 = 'quite satisfactory', 3 = 'satisfactory', 4 = 'excellent'). There were specific criteria for each construct of the CT model (abstraction, generalisation, algorithm, modularity, decomposition) and so each student was assigned a grade for each CT construct after answering the Q1 and Q2 questionnaires. A mean value was also calculated across all CT constructs (in Q1 and Q2 respectively). We consider the Q1 and Q2 measurements as indicators of student's level of CT skills development at certain phases during the training seminar (Q1 after the 4th session, Q2 after the 10th session). In the following, we refer to Q1 measurement as "Students' starting CT skill level" (or simply CT-4) and to Q2 measurement as "Students' final CT skill level" (or simply CT-10). As CT-4 is a measurement reflecting students' CT skills early in the training, we use it as covariate in our statistical analysis. We would like to clarify that although administering a pre-test

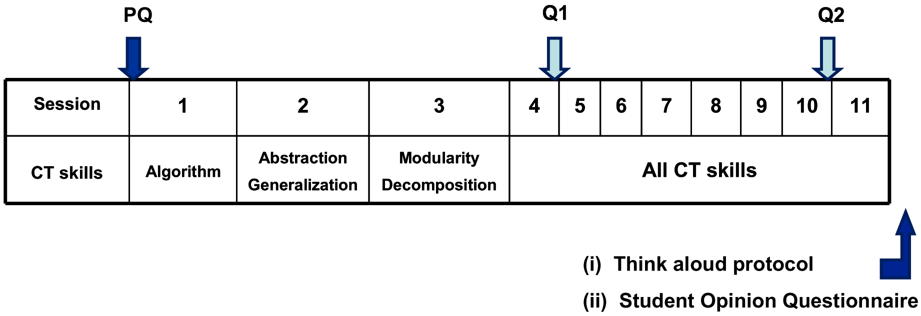

Fig. 1. Seminar organisational structure: lower row indicates session when CT skills were introduced; arrows indicate sessions when evaluation interventions were conducted.

Table 3
CT skills and relevant prompts to trigger students' self-reflection.

Abstraction	What is common in robot behaviour in both programs? How would you describe this common behaviour? What is the common programming structure? Which is the information you actually need? What is irrelevant detail and not necessary in your description?
Generalisation	Propose a more general solution for the activity above, that can cover a wider variety of cases. Is the proposed solution more general and why?
Algorithm	Write step-by-step the operations needed so that the robot can do what the problem asks. What are the steps I will need to do to solve this problem?
Modularity	Are there any parts of the code that you have met before? Have you created your own blocks for these? What are they? Do you expect to need some parts of this particular code in the future or in a different problem?
Decomposition	Can I break down this complex problem into smaller ones? Can I solve and explain the smaller problems, building up a solution towards the complex problem?

before any training was feasible, we though as a better approach to first provide students with a common programming tool for expressing CT (in our case, the Lego Mindstorms programming software) and then collect initial data after few sessions (session 4). We argue that this approach enabled us to: (a) help students develop a homogeneous background that led to more reliable measurement of their initial CT level (students can express their CT using the same programming tool), and (b) compare students' "short training" CT development (session 4) to "long training" CT development (session 10).

Student Opinion Questionnaire (SOQ): An opinion questionnaire was handed out to students to be filled individually after the completion of the training. The instrument recorded: (a) students' subjective views on understanding CT concepts and developing relevant skills, and (b) students' views and opinions regarding the outcomes of the overall learning experience on four key aspects: (1) development of students' CT skills, (2) understanding of basic programming concepts, (3) students' in-group collaboration (benefits and possible drawbacks), and (4) likes and dislikes relevant to the overall activity.

Think-aloud protocol: After the training, students individually were given a certain robot programming task and were asked to describe aloud the process they would follow to solve it. Simultaneously, the researcher prompted students to reflect on CT concepts relevant to their solution. The assessment of the student's proposed solution was based also on the same graded criterion instrument (rubric) as before. We consider this grade as an indicator of student's CT skills when evaluated in a different context than that of the Q1 and Q2 questionnaires. The main difference is that the 'think-aloud' method allows students to express their thinking more freely as opposed to the highly structured form of the questionnaire instruments. In the following, we refer to 'think-aloud' measurement as "Students' TA CT skill level" (or simply CT-TA). As before, we have 5 individual measures for each CT construct and a mean CT-TA grade calculated across all CT constructs.

Interview: After the think-aloud activity students were asked (as a semi-structured interview) to freely state their opinion on key aspects of the activity (the four aspects described above in the SOQ section).

Observation: Systematic monitoring of the students' work was applied by taking notes on a structured form (observation sheets). Both the supervising researcher and trainers filled in the sheets and then extensively discussed their observations to reach consent and decide on their importance. So, an observation table was gradually developed displaying researchers' observations in order of their discussed importance.

3.6. Results

3.6.1. Statistical analysis

Profile questionnaire data revealed that none of the participating students had any previous experience with robotics. After the data collection, the statistical processing was as follows:

(a) Table 4 presents statistical controls applied on students' CT-4 and CT-10 scores in Junior high (J) and High vocational (H) groups.
(b) Table 5 presents statistical controls applied on students' CT-4 and CT-10 scores analysed in each of the five dimensions of the CT model.
(c) Table 6 presents statistical control applied on the students' CT-TA scores (both total and analytical scores for each of the five CT dimensions).
(d) Table 7 presents statistical control applied on students' CT-4 and CT-10 scores across different gender groups. Two gender groups were used: Girls and Boys at Junior (J) level. The Girls/Boys distribution in the High (H) group was highly uneven and this group was excluded from across gender comparisons.
(e) Table 8 presents statistical controls applied on students' CT-4 and CT-10 scores analysed in each one of the five dimensions of our CT model. As before (Table 7) data refer to gender groups only within J group.
(f) Table 9 presents statistical control applied on the students' CT-TA scores across gender (both total and analytical scores for each of the five CT dimensions).

Table 4
Comparing CT-10 and CT-4 scores between J and H groups.

Level	N	CT-4		CT-10		Paired t-test	ANCOVA
		M	*(SD)*	M	*(SD)*	CT-10 compared to CT-4 (same student group)	comparing CT-10 across student groups (CT-4 as covariate)
J	89	2.96	(0.51)	3.08	(0.59)	$t(88) = -1.91,\ p = 0.059$	$F(1, 161) = 0.289$
H	75	2.36	(0.69)	2.71	(0.77)	$t(74) = -6.69,\ p = 0.00^*$	$p = 0.592,\ \eta^2 = 0.02$
Total	164	2.69	(0.67)	2.91	(0.70)	$t(163) = -5.27,\ p = 0.00^*$	

* Significant difference at the 0.05 level.

Table 5
Comparing CT-10 and CT-4 scores analytically for the five CT dimensions.

CT skills	Level	CT-4		CT-10		Paired t-test	ANCOVA
		M	*(SD)*	M	*(SD)*		Comparing CT-10 across student groups with CT-4 as covariate
Abstraction	J	2.50	(0.81)	2.52	(0.87)	$t(88) = -0.22,\ p = 0.83$	$F(1, 161) = 0.014$
	H	2.61	(0.83)	2.57	(0.85)	$t(74) = 0.59,\ p = 0.55$	$p = 0.907,\ \eta^2 = 0.00$
Generalisation	J	2.57	(0.81)	2.70	(0.81)	$t(88) = -1.18,\ p = 0.24$	$F(1, 161) = 0.189$
	H	2.15	(0.89)	2.48	(0.98)	$t(74) = -2.79,\ p = 0.01^*$	$p = 0.665,\ \eta^2 = 0.01$
Algorithm	J	3.08	(0.74)	2.98	(0.64)	$t(88) = 1.06,\ p = 0.29$	$F(1, 161) = 0.446$
	H	2.48	(0.77)	2.81	(0.79)	$t(74) = -5.19,\ p = 0.00^*$	$p = 0.505,\ \eta^2 = 0.03$
Modularity	J	3.34	(0.78)	3.57	(0.87)	$t(88) = -2.19,\ p = 0.03^*$	$F(1, 161) = 0.0$
	H	2.11	(1.04)	2.80	(1.25)	$t(74) = -6.28,\ p = 0.00^*$	$p = 0.998,\ \eta^2 = 0.00$
Decomposition	J	3.11	(0.96)	3.66	(0.70)	$t(88) = -5.58,\ p = 0.00^*$	$F(1, 161) = 11.861$
	H	2.27	(1.03)	2.83	(1.06)	$t(74) = -5.31,\ p = 0.00^*$	$p = 0.001^*,\ \eta^2 = 0.69$

* Significant difference at the 0.05 level.

Table 6
Comparing CT-TA scores between J and H groups.

CT skills	J		H		t-test
	M	*(SD)*	M	*(SD)*	
Abstraction	2.20	(0.87)	2.31	(0.95)	$t(162) = -0.80,\ p = 0.42$
Generalisation	2.28	(1.09)	2.68	(1.06)	$t(162) = -2.40,\ p = 0.02^*$
Algorithm	2.77	(0.76)	2.60	(0.97)	$t(140) = 1.21,\ p = 0.23$
Modularity	2.60	(1.33)	2.36	(1.02)	$t(161) = 1.31,\ p = 0.19$
Decomposition	3.03	(1.19)	2.84	(1.05)	$t(162) = 1.06,\ p = 0.29$
Total CT-TA	2.62	(0.69)	2.60	(0.75)	$t(162) = -0.16,\ p = 0.87$

* Significant difference at the 0.05 level.

Table 7
Comparing CT-10 and CT-4 scores between gender groups (J level only).

Gender	N	CT-4		CT-10		Paired t-test	ANCOVA
		M	*(SD)*	M	*(SD)*	CT-10 compared to CT-4 (same student group)	Comparing CT-10 across student groups (CT-4 as covariate)
Girl	41	2.81	(0.49)	3.09	(0.65)	$t(40) = -3.43,\ p = 0.00^*$	$F(1, 86) = 1.146$
Boy	48	3.02	(0.54)	3.08	(0.51)	$t(47) = -0.71,\ p = 0.48$	$p = 0.287,\ \eta^2 = 0.013$
Total	89	2.92	(0.53)	3.09	(0.58)	$t(88) = -2.68,\ p = 0.01^*$	

* Significant difference at the 0.05 level.

3.6.2. Students' Opinion Questionnaire (SOQ)

Data from SOQs and interviews helped us understand students' opinions regarding the overall activity. Key findings can be summarised as follows:

(i) Students' subjective impression was that they acquired certain CT skills. They reported that they can detect and describe the common behaviours or programming structures used in different tasks ($M = 4.03$, SD $= 0.77$) and also that they can suggest a more general solution for a given problem ($M = 4.00$, SD $= 0.79$).

(ii) Students reported that the guidelines in the worksheets helped them develop a certain problem-solving process ($M = 3.73$, SD $= 0.80$). They find this process useful to think of ("it comes to mind") when solving problems in other domains as well ($M = 3.58$, SD $= 0.81$). Some relevant students' statements are: "*Now I think differently and solve problems more easily*" and "*I changed my way of thinking in problem solving even in other subjects such as mathematics*".

(iii) The students stated that they became familiar with basic programming constructs ($M = 4.16$, SD $= 0.68$) and that they would like to continue with programming. In particular, H level students mentioned that they better understood some basic programming concepts they learned in other programming environments, such as the control structure ("If…then…else") and the loop structure ("For…Next", "Do While…"). They also said that working with the robots not only helped them develop a deeper understanding of programming ($M = 4.11$, SD $= 0.67$) but also kept them in-

Table 8
Comparing CT-10 and CT-4 scores analytically for the five CT dimensions (J level only).

CT skills	Gender	CT-4		CT-10		Paired t-test	ANCOVA
		M	*(SD)*	M	*(SD)*		Comparing CT-10 across school levels (CT-4 as covariate)
Abstraction	Girl	2.42	*(0.78)*	2.62	*(0.91)*	$t(40) = -1.56,\ p = 0.13$	$F(1, 86) = 1.866$
	Boy	2.57	*(0.84)*	2.44	*(0.84)*	$t(47) = 0.88,\ p = 0.38$	$p = 0.175,\ \eta^2 = 0.021$
Generalisation	Girl	2.43	*(0.70)*	2.66	*(0.92)*	$t(40) = -1.66,\ p = 0.11$	$F(1, 86) = 0.00$
	Boy	2.70	*(0.87)*	2.74	*(0.91)*	$t(47) = -0.25,\ p = 0.80$	$p = 0.989,\ \eta^2 = 0.00$
Algorithm	Girl	3.01	*(0.71)*	2.99	*(0.65)*	$t(40) = 0.20,\ p = 0.85$	$F(1, 86) = 0.037$
	Boy	3.15	*(0.77)*	2.97	*(0.65)*	$t(47) = 1.15,\ p = 0.26$	$p = 0.848,\ \eta^2 = 0.00$
Modularity	Girl	3.18	*(0.79)*	3.54	*(0.94)*	$t(40) = -2.02,\ p = 0.05^*$	$F(1, 86) = 0.073$
	Boy	3.48	*(0.76)*	3.60	*(0.82)*	$t(47) = -0.98,\ p = 0.33$	$p = 0.787,\ \eta^2 = 0.001$
Decomposition	Girl	3.00	*(0.97)*	3.66	*(0.82)*	$t(40) = -4.59,\ p = 0.00^*$	$F(1, 86) = 0.123$
	Boy	3.20	*(0.94)*	3.67	*(0.60)*	$t(47) = -3.39,\ p = 0.00^*$	$p = 0.727,\ \eta^2 = 0.001$

* Significant difference at the 0.05 level.

Table 9
Statistical analysis comparing CT-TA between gender groups (J level only).

CT skills	Girls ($N = 41$)		Boys ($N = 48$)		Independent t-test
	M	*(SD)*	M	*(SD)*	
Abstraction	2.31	*(0.94)*	2.11	*(0.80)*	$t(87) = 1.12,$ $p = 0.27$
Generalisation	2.32	*(1.08)*	2.24	*(1.11)*	$t(87) = 0.33,$ $p = 0.74$
Algorithm	2.91	*(0.69)*	2.65	*(0.80)*	$t(87) = 1.66,$ $p = 0.10$
Modularity	2.68	*(1. 33)*	2.53	*(1.34)*	$t(87) = 0.54,$ $p = 0.60$
Decomposition	3.12	*(1. 25)*	2.95	*(1.16)*	$t(87) = 0.68,$ $p = 0.50$
Total CT-TA	2.71	*(0.73)*	2.54	*(0.66)*	$t(82) = 1.14,$ $p = 0.26$

terest and motivated them to keep working on programming ($M = 3.42$, SD $= 0.66$).

(iv) Regarding collaboration, the students enjoyed working in groups ("*three minds are better than one*", "*we motivate each other when working together*") and assuming CT relevant roles ($M = 4.06$, SD $= 0.72$) with the most popular role being that of the "Programmer".

(v) Finally, the students found the robotics experience very interesting ($M = 4.38$, SD $= 0.63$), reporting that they would like to continue practising ER in the future ($M = 3.65$, SD $= 0.84$) and engage in more challenging tasks. Indicative of their interest is the fact that when finishing with the worksheets, they explored different programming structures ("blocks") – even those they had not learned yet – and different ideas to expand and improve their solutions.

3.7. Discussion and conclusions

The current work analysed the development of students' computational thinking skills in the context of educational robotics, with special focus on the impact that the instructional approach may have on student groups of different ages and genders. The study provides evidence from evaluation instruments administered at various times during the activity, thus offering a picture of how CT skills develop as students' work progresses. Students' CT skills are also evaluated using different modalities in assessment instruments (questionnaires answered in written and problem-solving think aloud protocols). Finally, researchers' observations and qualitative data from students' opinion questionnaires help triangulate data and deeper understand their meaning.

A first observation is that students develop the same level of CT skills at the end of their training independently of age. Additionally, CT skills in most cases are significantly improved as the training proceeds (comparing CT-4 and CT-10 scores in Tables 4 and 5). This is clear for the total population and for each of the two groups, although for the J group appears as a strong tendency ($p = 0.059$) not exactly reaching the level of significance (Table 4, paired t-test and ANCOVA). Thus, one key conclusion is that the satisfactory development of CT skills needs a considerable number of training sessions – independently of student's age – and is not simply a matter of a few training sessions. This conclusion is in line with studies emphasising that skill development in general requires adequate amount of training time [33,53].

Reflecting further on Table 5, we see that significant differences between CT-10 and CT-4 measures are identified in certain cases independently of student's level (age) (clearly for the Modularity and Decomposition dimensions), while in other cases such differences are evidenced only for the H group (Algorithm and Generalisation dimensions) or not at all (Abstraction). To explain these differences we resort to researchers' observations regarding the group composition and students' preference for writing. Most students in the H group are boys not so willing to provide answers in written (this is in line with studies suggesting that boys are significantly more reluctant writers than girls, for example [54]). By contrast, students in the J group are almost equally distributed across gender and adopt a more positive attitude towards expressing themselves in written (compared to boys in H group).

Keeping this in mind, we explore the implications of data in Table 5. As the development of the Abstraction skill for both groups reaches a high level already in session 4 (not to be surpassed in the next sessions), this is an indication that students from session 4 onwards deal with programming tasks without further development of this skill in a way that is reflected in the measures. Also, the additional workload of expressing this skill in written does not seem to affect students in the High group. However, Generalisation and Algorithm skills are further developed (from CT-4 to CT-10) only for the High group (Table 5). This is probably explained by the observation that younger students in the J group are more willing to follow instructions and provide answers in written (so their scores are high already from the 4th session (CT-4)), while students in the H group improve significantly from CT-4 to CT-10 as they gradually familiarise themselves with following the worksheet guidelines and become more willing to provide written documents expressing their thinking. These explanations are further supported by the fact that the aforementioned differences are not observed when the modality of the assessment instrument changes (see also comments below regarding Table 6). Finally, considering the Modularity and Decomposition dimensions we observe significant differences between CT-10 and CT-4, for both J and H groups. Regarding Modularity, we believe that the significant improvement of

CT-10 score for the J group is mainly due to the improvement of the girls' CT-10 score in the group (see also Modularity in Table 8, we comment on that further below). Regarding Decomposition, we see that both boys and girls in J group improve significantly their CT-10 score (see also Table 8) and this, we believe, is due to the increase of problem complexity as the training proceeds. Increased problem complexity gives the opportunity to students of both groups (J and H) to practise the skill more extensively and this is reflected in their scores. Additionally, we identify – only for Decomposition – a statistically significant difference between the two groups favouring students in the J group (ANCOVA in Table 5). We suggest that this is another manifestation of the unwillingness of boys in the H group to routinely follow instructions. Students in this group do not actually think it is necessary to decompose the problem into smaller ones to solve it. However, this attitude could also be linked to the cognitive maturity of elder adolescents in group H as compared to the younger adolescents in group J, which enables the former to manage more complex programming solutions without decomposing them.

Moving on to Table 6 (CT-TA scores), we see that when evaluating students' CT skills orally (Think-Aloud protocol) no between-group differences are identified (except for Generalisation, favouring the H group). This, corroborates our already stated conclusions that: (a) development of CT skills happens in the same way for both groups independently of age, and (b) CT skills measures might be affected by the workload imposed on students from the recording instrument modality. When students are asked to provide written evidence of their skills, they might appear to underperform because of poorly following the instructions (as in Decomposition, Table 5). However, it is not clear why the H group outperforms J in Generalisation (Table 6). One possible explanation is that the oral modality allows the specific profile male students in group H to thoroughly express their more complex thinking required to describe a generalised problem solution. Thus, we might have here another indication of the interaction between students' scores and assessment instrument modality, which should be seriously considered by researchers in relevant studies. In all other dimensions (and also in the total CT skills score) no significant differences are recorded.

Next, we focus on the analysis of scores between gender groups (Tables 7–9). A key conclusion here is that, although boys and girls reach the same CT skills level (ANCOVA in Table 7), there is, however, a significant difference between CT-10 and CT-4 scores for the girls' subsample indicating that the girls need longer time to reach the same skills level. This difference is also reflected on the total population (paired t-tests in Table 7). This outcome is in line with other studies suggesting that girls seem to require more time, compared to boys, when it comes to skills development (see [55]).

Table 8 presents analytically the CT-10 and CT-4 skills scores in the five dimensions for boys and girls. The previously discussed pattern ("both genders reach the same skill level but girls need more time") appears again for the Abstraction (strong tendency for girls, $p = 0.13$), Generalisation (strong tendency for girls, $p = 0.11$), Modularity (significant difference for girls, $p = 0.05$), but not for the Algorithm or the Decomposition dimensions. For the Decomposition, we believe, the explanation is the same as before; the increased complexity of programming tasks as the training proceeds allow students of both genders to practise decomposition more systematically and this is reflected in their scores.

Finally, some interesting evidence emerges in Table 9. On one hand, no significant differences in CT skill scores appear (neither for the total CT-TA score nor for any dimension, except for strong tendency in Algorithm, favouring girls). The "no significant" outcome is compatible with the overall gender pattern that boys and girls reach finally the same skills level. On the other hand, however, the strong tendency in Algorithm seems to be at odds with what has been discussed so far, as the Algorithm relevant skill is the only one developed in the same way by both boys and girls (Table 8). One possible explanation might be that while girls in the J group understand and express the algorithmic dimension of a programming task as efficiently as boys (Table 8), nevertheless, when they are additionally given the opportunity to express their algorithmic thinking orally, they tend to do that more effectively than boys (Table 9). Anyway, we acknowledge that more research is needed to clarify that point.

By reflecting on researchers' observations we report the most important of them as follows: (a) Despite any initial difficulties in grasping the concept of abstraction, students were able to easily identify the common programming concepts when comparing different scenarios. This conclusion is in line with quantitative data indicating that Abstraction is easily grasped and practised by students. (b) In the beginning, the students, faced difficulties in understanding the concept of generalisation and suggesting more general solutions. However, at the end of the training, interesting generalisations were observed in students' solutions. Especially students in the H group assimilated the concept more easily and used it in the activities often without any intervention from trainers. This corroborates the findings in Table 6 where elder adolescents (the H group) seem to practise Generalisation significantly better when the assessment modality is oral. Thus, Generalisation appears to be a CT skill which develops better in elders and this is perhaps related to the cognitive developmental level of the H group. Certainly, more research is needed to further clarify the issue. (c) Most students had difficulty in describing the algorithm with clarity and accuracy. They preferred to describe a process in general rather than analyse it step by step. Perhaps this is due to the cognitive load induced when analytically expressing the algorithm. Here, again, a modality effect is identified (girls tend to orally describe the algorithm better than boys— Table 9). (d) The students, encouraged by the trainers, practised the skill of modularity in their activities by creating their own programming "blocks". The students in J group familiarised with and integrated the skill more than the students in H group. This last observation is in line with quantitative data (Table 5) showing that J group applies Decomposition better. We attribute that behaviour mostly to H group students' unwillingness to follow instructions for decomposing problems, being able to manage the code as a whole.

Overall, this study provides evidence that: (a) students of different ages (15 vs. 18) and genders eventually reach the same level of CT skills development; this view is supported by evidence from assessment instruments using two modalities. (b) Time is an essential commodity for CT skills development; skills level evaluated in later session have been found in most cases to be significantly improved when compared to initial session. (c) When analysing the particular skills of the CT model certain differences are identified which are related to the following factors: age and student cognitive developmental level, students' attitudes relevant to following instructions and afford workload induced by the task, and also gender. (d) The assessment instrument modality may have an impact on students' scores as boys are, generally, more reluctant writers compared to girls. When this attitude is intense then boys may appear to underperform if skills evaluation is based on instruments of written modality. (e) Girls appear in most cases to need more time (training sessions) in order to reach the same skill level as boys. (f) Provided that the overall instructional context is supportive and the learning activity time is adequate, students may overcome their initial difficulties and successfully develop their CT skills.

Understanding the above conclusions should be done while also considering the limitations of the study. It is important to remember that educational robotics activities cannot be conducted under full experimental control and many factors might interact in an unexpected and – relatively – uncontrolled way. The current study provides evidence coming from various data collecting methods and with assessment instruments of different modality, something that – we believe – increases the validity of the conclusions. However, it was not possible to include a control group in the design that would allow exploring the issue of whether the CT skills in ER activities develop in the same way compared to a control non-ER instructional condition. An additional limitation is the exclusion of the H group from across gender controls due to the highly uneven distribution of girls/boys in the sample. This does not permit the current study to simultaneously apply across-age and across-gender controls that could further shed light on the gender relevant differences and reveal any possible interaction between the two factors. Finally, the study did not administer any pre-intervention controls of students' "preference for writing" attitude and general ability levels. Our experience indicates that such tests could provide valuable information regarding some of the observed gender and group relevant differences.

Acknowledgements

The authors would like to thank all school teachers, postgraduate student-trainers and the students involved in the ER seminars for their helpful collaboration and also the journal editor and the manuscript reviewers for their constructive comments.

References

[1] J.M. Wing, Computational thinking, Commun. ACM 49 (2006) 33–35. http://dx.doi.org/10.1145/1118178.1118215.
[2] J.M. Wing, Computational thinking and thinking about computing, Philos. Trans. A Math. Phys. Eng. Sci. 366 (2008) 3717–3725. http://dx.doi.org/10.1098/rsta.2008.0118.
[3] A. Repenning, D. Webb, A. Ioannidou, Scalable game design and the development of a checklist for getting computational thinking into public schools, in: SIGCSE 10 Proc. 41st ACM Tech. Symp. Comput. Sci. Educ., 2010, pp. 265–269. http://dx.doi.org/10.1145/1734263.1734357.
[4] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, et al., Computational thinking for youth in practice, ACM Inroads 2 (2011) 32–37. http://dx.doi.org/10.1145/1929887.1929902.
[5] S. Blanchard, V. Freiman, N. Lirrete-Pitre, Strategies used by elementary schoolchildren solving robotics-based complex tasks: Innovative potential of technology, Procedia-Soc. Behav. Sci. 2 (2010) 2851–2857. http://dx.doi.org/10.1016/j.sbspro.2010.03.427.
[6] J.M. Chambers, M. Carbonaro, M. Rex, S. Grove, Scaffolding knowledge construction through robotic technology: A middle school case study, Electron. J. Integr. Technol. Educ. 6 (2007) 55–70. http://ww.notiziedeattualita.com/ficheros/d9504e7adfc58045ce694570b00fe6f3.
[7] D. Alimisis, Teacher education on robotics-enhanced constructivist pedagogical methods. School of Pedagogical and Technological Education (ASPETE), 2009. http://www.terecop.eu/en/Products1.html.
[8] C. Rogers, M. Portsmore, Bringing engineering to elementary school, J. STEM Educ. 5 (2004) 17–29.
[9] S. Papert, Mindstorms: Children, Computers, and Powerful Ideas, Basic Books, Inc., New York, 1980.
[10] F.B. Vavassori Benitti, Exploring the educational potential of robotics in schools: a systematic review, Comput. Educ. 58 (2011) 978–988. http://dx.doi.org/10.1016/j.compedu.2011.10.006.
[11] S. Atmatzidou, S. Demetriadis, Evaluating the role of collaboration scripts as group guiding tools in activities of educational robotics: Conclusions from three case studies, in: 2012 IEEE 12th Int. Conf. Adv. Learn. Technol., IEEE, 2012, pp. 298–302. http://dx.doi.org/10.1109/ICALT.2012.111.
[12] M. Petre, B. Price, Using robotics to motivate "back door" learning, Educ. Inf. Technol. 9 (2004) 147–158. http://dx.doi.org/10.1023/B:EAIT.0000027927.78380.60.
[13] I.R. Nourbakhsh, K. Crowley, A. Bhave, E. Hamner, T. Hsiu, A. Perez-Bergquist, et al., The robotic autonomy mobile robotics course: Robot design, curriculum design and educational assessment, Auton. Robots 18 (2005) 103–127. http://dx.doi.org/10.1023/B:AURO.0000047303.20624.02.
[14] S. Atmatzidou, I. Markelis, S. Demetriadis, The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning, in: Work. Proc. Int. Conf. Simulation, Model. Program. Auton. Robot., 2008, pp. 22–30.
[15] D. Miller, I. Nourbakhsh, R. Siegwart, Robots for education in handbook of robotics, in: Springer Handb. Robot., 2008, pp. 1283–1301. http://dx.doi.org/10.1007/978-3-540-30301-5_56.
[16] A. Eguchi, What is educational robotics? Theories behind it and practical implementation, in: Soc. Inf. Technol. Teach. Educ. Int. Conf. 2010, Association for the Advancement of Computing in Education, AACE, Chesapeake, VA, 2010, pp. 4006–4014. http://www.editlib.org/p/34007.
[17] A. Khanlari, Effects of robotics on 21st century skills, Eur. Sci. J. 9 (27) (2013) 27–36.
[18] A. Vollstedt, M. Robinson, E. Wang, Using robotics to enhance science, technology, engineering, and mathematics curricula, in: Proceedings of American Society for Engineering Education Pacific Southwest Annual Conference, 2007.
[19] M. Mataric, N. Koenig, D. Feil-Seifer, Materials for enabling hands-on robotics and STEM education, in: AAAI Spring Symposium: Semantic Scientific Knowledge Integration, 2007, pp. 99–102. Retrieved, September 2012 from http://www.aaai.org/Papers/Symposia/Spring/2007/SS-07-09/SS07-09-022.pdf.
[20] R. Mead, S. Thomas, J. Weinberg, From grade school to grad school: An integrated STEM pipeline model through robotics, in: Robot K-12 Education: A New Technology for Learning, IGI Global, 2012, pp. 302–325. http://dx.doi.org/10.4018/978-1-4666-0182-6.ch015 (Chapter 15).
[21] M. Mataric, Robotics education for all ages, in: Proceedings, AAAI Spring Symp. Access. Hands-on AI Robot. Educ. Palo Alto, CA, 2004, pp. 22–24. https://www.aaai.org/Papers/Symposia/Spring/2004/SS-04-01/SS04-01-004.pdf.
[22] D. Alimisis, Educational robotics: Open questions and new challenges, Themes Sci. Technol. Educ. 6 (2013) 63–71.
[23] O. Astrachan, T. Barnes, D.D. Garcia, J. Paul, B. Simon, L. Snyder, CS principles: piloting a new course at national scale, in: Proc. 42nd ACM Tech. Symp. Comput. Sci. Educ.—SIGCSE'11, ACM Press, New York, USA, 2011, pp. 397–398. http://dx.doi.org/10.1145/1953163.1953281.
[24] A. Settle, L. Perkoviæ, S. Hwang, J. Jones, A framework for computational thinking across the curriculum, in: Proc. Fifteenth Annu. Conf. Innov. Technol. Comput. Sci. Educ.—ITiCSE'10, ACM Press, New York, USA, 2010, pp. 123–127. http://dx.doi.org/10.1145/1822090.1822126. Retrieved from http://via.library.depaul.edu/tr/13.
[25] C. Dierbach, H. Hochheiser, S. Collins, G. Jerome, C. Ariza, T. Kelleher, et al., A model for piloting pathways for computational thinking in a general education curriculum, in: Proc. 42nd ACM Tech. Symp. Comput. Sci. Educ.—SIGCSE'11, ACM Press, New York, 2011, pp. 257–262. http://dx.doi.org/10.1145/1953163.1953243.
[26] M. Berland, V.R. Lee, Collaborative strategic board games as a site for distributed computational thinking, Int. J. Game-Based Learn. 1 (2) (2011) 65–81. http://dx.doi.org/10.4018/ijgbl.2011040105.
[27] S. Grover, Robotics and engineering for middle and high school students to develop computational thinking, in: Annual Meeting of the American Educational Research Association, New Orleans, LA, 2011, pp. 1–15.
[28] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, J.T. Korb, Introducing computational thinking in education courses, in: Proceedings 42nd ACM Technical Symposium on Computer Science Education, ACM Press, New York, 2011, pp. 465–470. http://dx.doi.org/10.1145/1953163.1953297.
[29] E.R. Kazakoff, A. Sullivan, M.U. Bers, The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood, Early Child. Educ. J. 41 (2013) 245–255. http://dx.doi.org/10.1007/s10643-012-0554-5.
[30] M.U. Bers, The TangibleK robotics program: Applied computational thinking for young children, Early Child. Res. Pract. 12 (2010) 1–20. Retrieved, September 2014 from http://ecrp.uiuc.edu/v12n2/bers.html.
[31] S. Grover, R. Pea, Computational thinking in K–12: A review of the state of the field, Educ. Res. 42 (2013) 38–43. http://dx.doi.org/10.3102/0013189X12463051.
[32] M.U. Bers, L. Flannery, E.R. Kazakoff, J.A. Crouser, A curriculum unit on programming and robotics, 2010. Retrieved, June 2014 from http://ase.tufts.edu/devtech/tangiblek/ClassroomCurriculumVersion1.02Nov82010.pdf.
[33] M.U. Bers, L. Flannery, E.R. Kazakoff, A. Sullivan, Computational thinking and tinkering: Exploration of an early childhood robotics curriculum, Comput. Educ. 72 (2014) 145–157. http://dx.doi.org/10.1016/j.compedu.2013.10.020.
[34] D.S. Touretzky, D. Marghitu, S. Ludi, D. Bernstein, L. Ni, Accelerating K-12 computational thinking using scaffolding, staging, and abstraction, in: Proceeding 44th ACM Technical Symposium Computer Science Education—SIGCSE'13, ACM Press, New York, 2013, p. 609. http://dx.doi.org/10.1145/2445196.2445374.
[35] M.R. Penmetcha, Exploring the effectiveness of robotics as a vehicle for computational thinking, Purdue University, 2012. Retrieved, March 2014 from http://search.proquest.com/docview/1221535411.
[36] A. Eguchi, Learning experience through robocupjunior: Promoting engineering and computational thinking skills through robotics competition, in: K-12 Comput. Sci. Comput. Think. Initiat., Indianapolis, Indiana, 2014, pp. 1–18. https://peer.asee.org/20743.
[37] C. Kazimoglu, M. Kiernan, L. Bacon, L. Mackinnon, A serious game for developing computational thinking and learning introductory computer programming, Procedia-Soc. Behav. Sci. 47 (2012) 1991–1999. http://dx.doi.org/10.1016/j.sbspro.2012.06.938.
[38] V. Barr, C. Stephenson, Computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads 2 (2011) 48–54. http://dx.doi.org/10.1145/1929887.1929905.

[39] L. Werner, J. Denner, S. Campe, D.C. Kawamoto, The fairy performance assessment: Measuring computational thinking in middle school, in: Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ.—SIGCSE'12, ACM Press, New York, 2012, pp. 7–12. http://dx.doi.org/10.1145/2157136.2157200.
[40] W. Allan, B. Coulter, M.B. Garden, J. Denner, J. Erickson, I. Lee, et al. Computational Thinking for Youth What does computational thinking for youth look like in practice? in: ITEST Small Group on Computational Thinking, 2010. http://stelar.edc.org/publications/computational-thinking-youth.
[41] M.C. Linn, A.V. Aho, M.B. Blake, R. Constable, Y.B. Kafai, J.L. Kolodner, et al., Report of a Workshop on The Scope and Nature of Computational Thinking, Press, The National Academies, Washington, 2010, http://www.nap.edu/catalog.php?record_id=12840.
[42] G.H. Fletcher, J.J. Lu, Education: Human computing skills: rethinking the K-12 experience, Commun. ACM 52 (2009) 23–25. http://dx.doi.org/10.1145/1461928.1461938.
[43] American Association of University Women, Greenberg-Lake–the Analysis Group, Shortchanging Girls, Shortchanging America: A nationwide poll to assess self esteem, educational experiences, interest in math and science, and career Aspirations of Girls and Boys Ages 9–15, American Association of University Women, (1991). http://www.aauw.org/files/2013/02/shortchanging-girls-shortchanging-america-executive-summary.pdf.
[44] D.H. Shunk, F. Pajeres, The development of academic self-efficacy, in: A. Wigfield, J. Eccles (Eds.), Development of Achievement Motivation, Vol. 1446, 2004, pp. 16–32. http://www.uky.edu/~eushe2/Pajares/SchunkPajares2001.PDF.
[45] C. Stein, K. Nickerson, Botball robotics and gender differences in middle school teams, in: Proceedings American Society for Engineering Education Annual Conference, 2004. http://dpm.kipr.org/papers/asee04-gender.pdf.
[46] J.B. Weinberg, J.C. Pettibone, S.L. Thomas, M.L. Stephen, C. Stein, The impact of robot projects on girl's attitudes toward science and engineering, in: RSS Robotics in Education Workshop, 2007, pp. 1–2. https://www.siue.edu/engineering/pdf/WeinbergRSSWorkshop2007.pdf.
[47] E. Milto, C. Rogers, M. Portsmore, Gender differences in confidence levels, group interactions, and feelings about competition in an introductory robotics course, in: 32nd Annu. Front. Educ., IEEE, 2002. http://dx.doi.org/10.1109/FIE.2002.1158224. F4C7-14.
[48] I.R. Nourbakhsh, E. Hamner, K. Crowley, K. Wilkinson, Formal measures of learning in a secondary school mobile robotics course, in: IEEE Int. Conf. Robot. Autom. Proceedings, IEEE, 2004, pp. 1831–1836. http://dx.doi.org/10.1109/ROBOT.2004.1308090.
[49] C.C. Cheng, P.L. Huang, K.H. Huang, Cooperative learning in lego robotics projects: Exploring the impacts of group formation on interaction and achievement, J. Netw. 8 (2013) 1529–1535. http://dx.doi.org/10.4304/jnw.8.7.1529-1535.
[50] C. Hill, C. Corbett, A.St Rose, Why So few? Women in Science, Technology, Engineering, and Mathematics, AAUW, 2010, http://eric.ed.gov/ERICWebPortal/recordDetail?accno=ED509653.
[51] A. Sullivan, M.U. Bers, Gender differences in kindergarteners' robotics and programming achievement, Int. J. Technol. Des. Educ. 23 (2013) 691–702. http://dx.doi.org/10.1007/s10798-012-9210-z.
[52] J. Spolsky, The Law of Leaky Abstractions, 2002. Retrieved, September 2014 from http://www.joelonsoftware.com/articles/LeakyAbstractions.html.
[53] D. Clark, E. Tanner-Smith, S. Killingsworth, Digital games, design, and learning: A systematic review and meta-analysis, Rev. Educ. Res. (2015) http://dx.doi.org/10.3102/0034654315582065.
[54] T. Merisuo-Storm, Girls and boys like to read and write different texts, Scand. J. Educ. Res. 50 (2006) 111–125. http://dx.doi.org/10.1080/00313830600576039.
[55] M. Papastergiou, Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation, Comput. Educ. 52 (2009) 1–12. http://dx.doi.org/10.1016/j.compedu.2008.06.004.

Soumela Atmatzidou is a Ph.D. candidate in the Informatics Department, Aristotle Univercity of Thessaloniki (AUTh) since 2010. She is Informatics teacher in Secondary schools since 1989. She received a Bachelor Degree from Physics Department (1983), a Master Diploma (M.Sc.) in Electronic Physics (1998) and also a M.Sc. in ICT in Education from the Informatics Department (2008), all from AUTh. Her research interests focus educational robotics activities and explore how supportive interventions during an ER activity may help students develop and improve high order skills.

Stavros Demetriadis is currently Associate Professor in the Department of Informatics, Aristotle University of Thessaloniki, Greece teaching courses and conducting research in the area of Learning Technologies. He holds a B.Sc. degree in Physics, M.Sc. degree in Electronic Physics, and Ph.D. degree in Multimedia Educational Technology from Aristotle University. He has published more than 150 research papers in International Journals and International/National Conference proceedings. His research interests include: Computer-Supported Collaborative Learning (CSCL) with emphasis on Collaboration Scripts, Adaptive and Intelligent systems for Collaborative learning support, Multimedia learning, Educational Robotics and Tangible Interfaces for introductory programming.