Fostering Computational Thinking Through Educational Robots

Emilia-Felicia Coșniță[1], Păun Antonescu[2]

1. National University of Science and Technology Politehnica Bucharest, Romania, emilia.cosnita@stud.mec.upb.ro
2. National University of Science and Technology Politehnica Bucharest, Romania, panton38@hotmail.com

DOI: 10.63467/alls15.art13

Abstract

The study analyzes the development of computational thinking and technical skills through educational robots, combining theoretical instruction with the development of haptic robotic devices. It proposes a teaching framework that facilitates learning through robotics projects, using force-feedback haptic devices, enabling students to acquire essential skills, improve their digital abilities, and adapt to real-world challenges.

The study evaluates the impact of integrating educational robots on cognitive development within an appropriate teaching environment, utilizing the equipment in the robotics lab. The results indicate an improvement in deduction and reasoning abilities, programming skills, use of haptic interfaces, as well as in building and manipulating robots. The implementation of robots and haptic devices has led to significant changes in both teaching strategy and the students' learning process.

Keywords: computational thinking; educational robots; project-based learning; haptic; teacher.

JEL Classification: C63, C69, I21, I290.

1. Introduction

Computational thinking refers to the approach of problem-solving and information processing in a way that can be addressed through algorithms and computers. It involves identifying, representing, and solving problems in a form that can be processed by a computer system. Additionally, it entails discovering solutions by breaking down problems into smaller steps (dividing and conquering complexity), recognizing patterns, and applying mathematical logic to develop optimal solutions (Wing, 2006). According to Jeanette Wing, a prominent advocate of computational thinking, it is an essential tool not only in the field of computer science but also in other disciplines, as it provides a method for structuring critical thinking and facilitating the understanding of the complexity of the world (Wing, 2006; Grover & Pea, 2013). Furthermore, computational thinking promotes the development of higher cognitive skills, such as problem-solving, creativity, and abstract thinking, which are crucial in preparing students for the challenges of the 21st century (Barr & Stephenson, 2011).

Educational robotics activities enable students to practice essential skills such as problem decomposition, abstraction, algorithm design, debugging, iteration, and generalization, representing six key facets of computational thinking (Shute et al., 2017). The use of educational robots in learning activities contributes to enhancing students' cognitive abilities. Through robot-assisted learning, students gain a better understanding of programming concepts such as sequencing, conditions, and loops, thus promoting problem-solving skills (Evripidou et al., 2021).

This study aimed to assess five competencies of computational thinking in students participating in a robotics club, where they practically built educational robots in the form of force-feedback haptic devices, compared to students who engaged in formal activities within a formal setting. At the end of the study, feedback was gathered from students, and conclusions were drawn.

2. Literature Review

Students who participated in robot-assisted learning activities demonstrated a better understanding of programming concepts and developed problem-solving skills (Chen & Chung, 2023). These activities provide constructive learning experiences by stimulating the visual, auditory, and tactile senses, thereby facilitating the development of cognitive skills and computational thinking (CT) in students.

A proposed model for educational robotics activities is the CCPS (Creative Computational Problem Solving) model, which integrates the process of creative problem-solving with the use of educational robots. This model includes phases such as understanding the problem, generating ideas, formulating the robot's behavior, programming the behavior, and evaluating the solution, thus promoting the development of CT skills in students (Romero et al., 2017).

Pou, Canaleta & Fonseca integrated educational robotics activities and computational thinking within a project-based learning (PBL) framework in a secondary school in Barcelona, Spain. Students used visual programming platforms, such as Scratch, to develop CT skills and competencies in the fields of science, technology, engineering, arts, and mathematics (STEAM). The study's results showed a significant improvement in these concepts and skills compared to other educational methodologies, highlighting the effectiveness of integrating educational robots into the school curriculum.

3. Methodology

The methodology used was project-based learning (PBL). The goal of PBL is to transform the educational process through the integration of digital tools, thereby facilitating the development of essential skills for the 21st century. This approach aims to personalize learning, allowing students to learn at their own pace and access diverse educational resources tailored to their individual needs.

Students presented the results of their project-based learning (PBL) in the form of a haptic device with force-feedback, using the Arduino programming environment. As students from the mathematics and computer science specialization who participated in the robotics club, there was no need to provide them with a conceptual introduction to the content of the programming environment. They were able to access and use various information sources, including online resources and technical databases, to enhance their knowledge and find solutions to the challenges they encountered. The teacher acted as a guide, clarifying doubts and supporting them in creating a final product.

In the PBL methodology, collaboration among peers is crucial, involving the sharing of knowledge for the benefit of the group, to achieve a common goal. Additionally, trial and error are considered essential factors in the process.

The study included 104 students (69.23% boys, 30.77% girls) aged between 14 and 18 years. They were divided into two groups, one control group (CG) and one study group (SG), with 52 students in each. In each group, 8 teams were formed, four with 6 students and four with 7 students.

The same curriculum was applied to all groups. The proposed activities took into account the theoretical knowledge taught in formal activities within compulsory education, age-related characteristics, and the available components in the robotics club. Some of the components were 3D printed using the lab's printer. Collaboration among peers and cooperative learning were encouraged to help students better identify concepts, analyze problems, and build relationships with their colleagues.

3.1. The aim of the research

The study aims to analyze the performance of students who built educational robots in the form of force-feedback haptic devices in a non-formal setting at the robotics club, compared to students who studied programming using traditional training methods in a formal environment.

It examined how the development of computational thinking through the integration of educational robotics can occur naturally, without overburdening the teacher or the computer scientist, as well as how teaching can become more effective in the process.

3.2. The objectives of the research

O1. Establishing the theoretical concepts corresponding to the competencies aimed at developing computational thinking (CT) and selecting the hardware components to be used in building educational robots.

O2. Developing the teaching scenarios and identifying their stages for non-formal sessions.

O3. Comparing the performance of students in the control group with that of students in the study group after the completion of the force-feedback haptic devices built as educational robots.

3.3. Aspects of the Assessed Competencies

Type	Description
C1. Problem Decomposition	C1.1. Identifying and defining the problem: students learn to recognize and clearly formulate problems, which is essential for finding appropriate solutions. C1.2. Analyzing and structuring information: involves organizing and evaluating available data to better understand the problem. C1.3. Generating and evaluating solutions: students are encouraged to propose multiple solutions and critically analyze them to choose the most effective approach.
C2. Abstraction	C2.1. Identifying the essential elements of a problem: students learn to distinguish relevant information and isolate it from insignificant details, thereby facilitating a general understanding of the problem. C2.2. Abstract representation of a problem: involves formulating the problem in a generalized manner, using concepts and symbols that allow the application of solutions in various contexts. C2.3. Generalizing solutions: students are encouraged to apply the solutions found in new contexts, demonstrating the transferability and efficiency of the approaches developed.
C3. Algorithm	C3.1. Analyzing the problem statement and establishing the steps to solve it: involves

Type	Description
Design	clearly understanding the problem's requirements and identifying the necessary stages to reach the solution. Students learn to break complex problems into simpler subproblems, facilitating the development of an efficient algorithm. C3.2. Representing algorithms in pseudocode: students learn to express algorithms in a semi-formal form using pseudocode, which allows a clear understanding of their logic before actual implementation in a programming language. C3.3. Adhering to structured programming principles in algorithm development: refers to applying structured programming principles, such as the use of sequencing, selection, and iteration, to create efficient and easily understandable algorithms. Students learn to apply these principles to develop clear and efficient solutions to given problems.
C4. Iteration	C4.1. Applying loops for repeating instructions: students use loops (such as for, while) to repeat code sequences, thus streamlining the process of solving repetitive problems. C4.2. Testing and adjusting solutions: involves evaluating the obtained results and modifying algorithms to improve their performance or correctness. C4.3. Continuous improvement of solutions: students are encouraged to review and enhance their solutions through successive iterations, thereby developing self-assessment and continuous improvement skills.
C5. Generalization	C5.1. Identifying recurring patterns: students learn to recognize common patterns and structures in various problems, thereby facilitating the application of similar solutions in new contexts. C5.2. Applying solutions in varied contexts: involves using learned strategies and techniques to solve problems in different fields, demonstrating flexibility and adaptability. C5.3. Transferring knowledge between domains: students are encouraged to apply concepts and methods learned in one field to solve problems in other fields, thereby developing knowledge transfer skills.

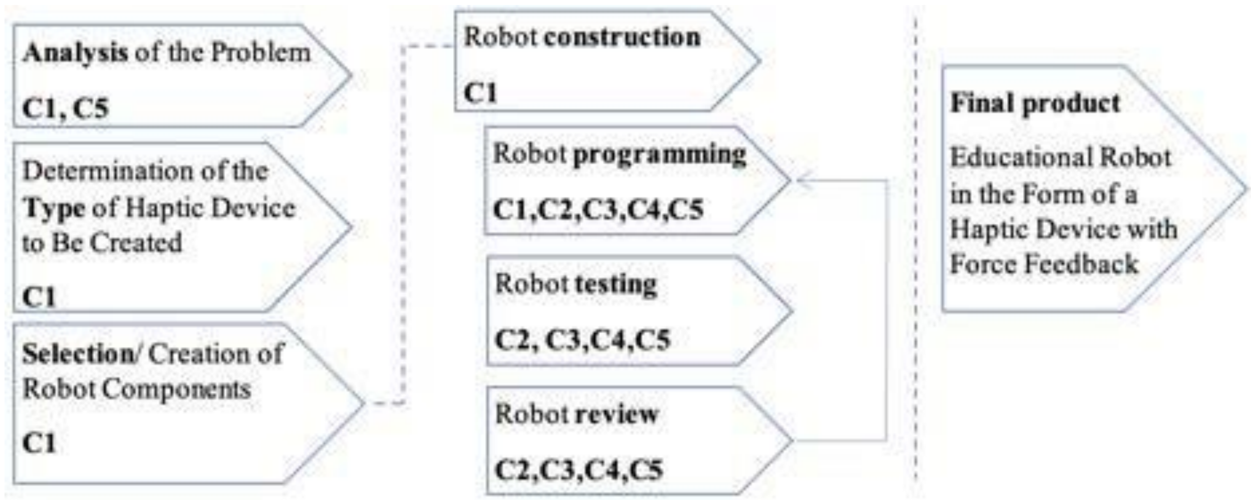

Figure 1. Design of the Educational Robot as a Haptic Feedback Device

3.4. Research steps for non-formal activities

STEPS	Activities	Teacher	Student
1. Introduction to haptic feedback concept	o Explaining the operating principle of haptic devices and their applications in various fields. Presenting the basic components: resonant linear actuators, development	o Explains the operating principle of haptic devices and their applications in various fields. Presents basic components: resonant linear actuators, development boards, and	o Listens to explanations, asks questions, and expresses ideas

	boards, and Arduino microcontrollers.	Arduino microcontrollers. Provides examples of the use of haptic devices in various fields.	
2. Planning and designing the device	o Establishing the purpose of the haptic device. Modifying the device design, selecting the type of actuator, and determining the mode of user interaction.	o Guide students in defining the project's purpose and technical requirements. Assist in identifying the necessary resources and establishing the work plan. Encourage critical thinking and creativity in the design process.	o Collaborating in groups to establish the desired functionality of the device. Drawing sketches and diagrams of the proposed device. Identifying technical solutions
3. Assembling the hardware	o Mounting components on a development board (breadboard). Connecting the breadboard, Arduino board, sensors, and	o Demonstrates proper assembly techniques, supervises student work to ensure safety and accuracy, and provides assistance with	o Assembles components on a breadboard, connects wires according to circuit diagrams, and tests connections for

	actuators according to the circuit diagrams. Verifying connections to prevent assembly errors.	technical issues.	correct operation.
4. Programming the device	o Writing and uploading the code in Arduino IDE to control the intensity and pattern of vibrations. Using libraries to access predefined effects. Testing and adjusting the code to achieve the desired feedback.	o Explains basic Arduino programming concepts, introduces functions and libraries for haptic feedback control, provides example code, and assists with debugging.	o Writes and uploads code using the Arduino IDE, tests and adjusts code to achieve desired feedback, and documents modifications and improvements.
5. Testing and evaluating the device	o Conducting tests to assess the effectiveness of haptic feedback under real-world usage conditions. Gathering user feedback to improve the device.	o Organizes testing sessions, provides evaluation criteria and constructive feedback, and encourages reflection on the learning process	o Actively participates in real-world testing, collects data, observes device performance, and suggests improvements based on

		and outcomes.	feedback received.
6. Reflection and continuous improvement	o Analyzing the learning process and identifying possible improvements. o Encouraging students to propose changes and experiment with different hardware and software configurations.	o Facilitates discussions about lessons learned and challenges encountered, encourages students to identify ways to enhance the project, and supports documentation of the process and results.	o Reflects on the learning experience and progress made, identifies strengths and areas for improvement in the project, and proposes ideas for future projects or further development of the device.

4. Results and Discussions

The study lasted five months and involved 104 students from grades IX-XII, 52 from formal education and 52 who voluntarily enrolled in the robotics club and built educational robots in the form of haptic devices with force feedback. In programming (software component), the graphical interface of the Arduino IDE environment and the C++ programming environment were used. The two groups consisted of 32 girls (30.77%) and 72 boys (69.23%), who were equally divided by age and gender.

Pre-testing and post-testing were conducted with 30 items each, targeting the five competencies of computational thinking, which had varying degrees of difficulty. Each competency was scored with 20 points. No points were awarded automatically. Knowledge from the mandatory curriculum of formal education was not assessed. Both tests focused on cognitive development and did not refer to practical aspects or robot construction. No additional points were awarded for functional robots.

At the beginning of the research, the hypothesis was tested, according to which there was no significant difference between the variations of the sample in the CG and SG groups across the five competencies.

Competences		C1		C2		C3		C4		C5	
	df	Mean	SS	Mean	SS	Mean	SS	Mean	SS	Mean	SS
Between groups	1	2.2193	2.2193	1.7127	1.7127	0.7073	0.7073	1.5396	1.5396	0.1659	0.1659
Within groups	102	9.146	932.868	8.600	877.188	9.576	976.752	8.053	821.4075	8.4943	866.41
Total (n-1)	103		935.0869		878.9006		977.4589		822.947		866.58
F			**0.24266**		**0.19915**		**0.07387**		**0.19119**		**0.01953**
p			0.623352		0.656351		0.786338		0.662857		0.889128

Table 1. Levene test for pre-test

For all five competencies, the p-value is quite large (C1: 0.62335, C2: 0.656351, C3: 0.786338, C4: 0.662857, C5: 0.889128), demonstrating that there is no statistically significant difference in the knowledge levels between students in the GC and SG groups. For C1, C2, C3, C4, C5, the independent sample F-statistic is small (C1: 0.24266, C2: 0.19915, C3: 0.07387, C4: 0.19119, C5: 0.01953). The difference between the variations of the sample in the two groups for the five competencies is not large enough to be statistically significant. The hypothesis was valid. k=2 groups, n=104 subjects.

At the end of the robotics club sessions, post-testing was conducted. We considered the hypothesis that the difference between the variations of the sample in the GC and SG groups is not significant. The hypothesis was tested using Levene's test.

Competences		C1		C2		C3		C4		C5	
	df	Mean	SS	Mean	SS	Mean	SS	Mean	SS	Mean	SS
Between groups	1	23.2317	23.2317	31.367	31.367	26.5412	26.5412	51.4566	51.457	37.560	37.5601
Within groups	102	7.1207	726.316	9.442	963.083	8.2606	842.577	8.0273	818.78	8.0423	820.32
Total (n-1)	103		749.548		994.45		869.12		870.239		857.876
F			**3.26254**		**3.32208**		**3.21301**		**6.41022**		**4.67031**
p			0.07383		0.07129		0.07602		0.01287		0.03303

Table 2. Levene test for post-test

For C1, C2, C3, the hypothesis was validated, but the p-value is very close to the 0.05 threshold (C1: 0.0738, C2: 0.07129, C3: 0.07602). For C4 and C5, the hypothesis was NOT validated. The p-value (C4: 0.012872, C5: 0.03303) indicates that there is a statistically significant difference in the knowledge levels between students in the CG and SG groups.

To verify the differences between the results obtained by students who built educational robots in the form of haptic devices with force feedback in a non-formal setting, at the robotics club, and those who studied programming using traditional training in a formal environment, Pearson's correlation coefficient was used (**Table 3**).

Competences		C1	C2	C3	C4	C5
Control group (CG)	Means scores	10.212	11.327	10.827	11.212	10.442
	Stand.dev.	1976.67	1627.44	1467.44	1362.67	1392.83
Study group (SG)	Means scores	9.462	9.827	10.154	11.404	9.788
	Stand.dev.	1502.92	1735.44	2014.77	2036.52	2074.67
Pearson coef. (R)		**-0.108**	**0.0446**	**0.006**	**-0.304**	**-0.0466**

Table 3. Pearson correlation coefficient calculated by competencies

For the competencies C1: -0.108, C4: -0.304, C5: -0.0466, no significant correlation was found. The other competencies, although they have a positive correlation (C2: 0.0446, C3: 0.006), according to Davies, these competencies have a weak relationship. There is a fragile relationship between the results of students who built educational robots in the form of haptic devices and those of students who only did traditional implementation.

5. Conclusions

The study aimed at developing computational thinking demonstrated that fundamental competencies such as problem decomposition and abstraction, as well as algorithm design, are not strongly correlated with building educational robots. These skills are critical for understanding and addressing complex problems, and their development occurs gradually, without significant jumps between learning stages (Grover & Pea, 2013). Between the students in the CG and SG groups, competencies that involve the ability to break down complex problems into smaller subproblems and to identify the essence of a problem by eliminating irrelevant details—processes essential for efficiently solving these problems—showed no remarkable differences. This supports Wing's (2006) conclusion, which emphasizes that problem decomposition is a fundamental foundation in computational thinking because it facilitates the organization and management of complex information in a systematic manner.

In contrast, the competencies of generalization and iteration showed significant improvements in cognitive development, indicating a substantial enhancement in logical thinking and the ability to apply solutions to a broader set of problems. Barr and Stephenson (2011) highlight that developing the ability to generalize and iterate solutions significantly contributes to improved problem-solving skills and a deeper understanding of complex concepts. Generalization involves identifying common patterns across different problems, while iteration refers to the repeated use of procedures or algorithms to refine solutions. Research in computer science education shows that these skills are essential for developing a flexible and creative thinker (Grover & Pea, 2013).

Thus, the significant differences in the development of these competencies are related to the nature of the learning process, where generalization and iteration are much more intensive and challenging, stimulating a higher level of critical thinking and the ability to adapt solutions to various contexts.

The study validated the conclusions from Romero et al. (2017), demonstrating that structured interventions based on CCPS can reduce trial-and-error behaviors and stimulate cognitive processes related to problem understanding, idea generation, and solution formulation.

At the end of the study, the students from the CG group demonstrated fundamental knowledge in the fields of electrical engineering and electronics, having a solid understanding of the essential principles underlying these disciplines. They were also familiar with the basics of robotics, with sufficient skills to understand key concepts such as control systems, sensors, and actuators. Their mathematical skills were at a basic level, which allowed them to apply simple mathematical concepts in solving technical and scientific problems.

Another important aspect was compliance with current health and safety regulations in the workplace, with students having a good understanding of the safety standards required for practical activities. They also demonstrated the ability to organize the workspace according to ergonomic requirements, ensuring that the work environment was adequate and comfortable, thus contributing to accident prevention and the efficiency of the activities carried out. These fundamental competencies contributed to their preparation for addressing more complex tasks in the field of robotics and related technologies, and their integration into the learning processes was essential for the success of the study.

When presenting the educational robot (final project), students from the SG group demonstrated advanced skills in:

- Explaining the specific terminology used in the robot construction process, highlighting key terms and fundamental concepts.
- Describing the applications of haptic devices, as well as identifying various areas of

daily life where these can be implemented, such as medicine, industry, or education.

- Recognizing robot typologies based on the structure of the kinematic unit, which allowed them to classify different types of robots according to their movement modes.
- Classifying robots based on their field of application, demonstrating an understanding of their diversity in sectors such as manufacturing, healthcare, or home automation.
- Detailing the process of constructing educational robots, including the essential steps and technologies used in their creation.
- Explaining the electrical connection interface of robots, highlighting how various electrical components are integrated to ensure the proper functioning of robotic systems.
- Describing the essential technical parameters of educational robots, such as motorization, sensor control, and interaction with the external environment.

These competencies reflect an advanced level of understanding and application of knowledge in the field of robotics, preparing students for both a practical and theoretical approach to emerging technologies.

References

Barr, Valerie & Stephenson, Chris. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. ACM Inroads. 2. https://doi.org/10.1145/1929887.1929905.

Chen, C.-H., & Chung, H.-Y. (2023). Fostering Computational Thinking and Problem-Solving in Programming: Integrating Concept Maps Into Robot Block-Based Programming. Journal of Educational Computing Research, 62(1), 186-207. https://doi.org/10.1177/07356331231205052 (Original work published 2024).

Grover, Shuchi & Pea, Roy. (2013). Computational Thinking in K–12 A Review of the State of the Field. Educational Researcher. 42. pp. 38-43. https://doi.org/10.3102/0013189X12463051.

James A. Davies, *Elementary Survey Analysis*. New Jersey: Prentice Hall, 1971.

Romero, Margarida & Lepage, Alexandre & Lille, Benjamin. (2017). Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education. 14. https://doi.org/10.1186/s41239-017-0080-z.

Shute, Valerie & Sun, Chen & Asbell-Clarke, Jodi. (2017). Demystifying computational thinking. Educational Research Review. 22. https://doi.org/10.1016/j.edurev.2017.09.003.

S. Evripidou, A. Amanatiadis, K. Christodoulou and S. A. Chatzichristofis, "Introducing Algorithmic Thinking and Sequencing Using Tangible Robots," in IEEE Transactions on Learning Technologies, vol. 14, no. 1, pp. 93-105, Feb. 2021. https://doi.org/10.1109/TLT.2021.3058060.

Valls Pou, A., Canaleta, X., & Fonseca, D. (2022). Computational Thinking and Educational Robotics Integrated into Project-Based Learning. *Sensors*, *22*(10), 3746. https://doi.org/10.3390/s22103746.

Wing, Jeannette. (2006). Computational Thinking. Communications of the ACM. 49. pp. 33-35. https://doi.org/10.1145/1118178.1118215.