International Journal of Engineering Education Vol. 37, No. 4, pp. 1080–1089, 2021
Printed in Great Britain

0949-149X/91 $3.00+0.00

Promoting Systems Thinking and Abstract Thinking in High-School Electronics Students: Integration of Dedicated Tasks into Project-Based Learning*

AZIZ SHEKH-ABED, ORIT HAZZAN and AHARON GERO
Department of Education in Technology and Science, Technion – Israel Institute of Technology, Haifa 32000, Israel.
E-mail: azizs@campus.technion.ac.il, oritha@ed.technion.ac.il, gero@technion.ac.il

Project-based learning is receiving ongoing attention in engineering education. Recently, emphasis has been placed on the importance of systems thinking and abstract thinking among high-school and undergraduate students. In light of the recently reported positive correlation between systems thinking and abstract thinking, the research described in this paper examined whether these two thinking skills could be promoted simultaneously through project-based learning that included dedicated tasks. The study, which used quantitative and qualitative tools, involved 36 high-school electronics students (Grade 12). According to the findings, a significant improvement (large effect) was achieved in both types of thinking. Specifically, the students adopted some of the systems thinker's and abstract thinker's features.

Keywords: project-based learning; systems thinking; abstract thinking; electronics students

1. Introduction

Project-based learning is a pedagogical approach in which students deal with complex tasks taken from real-world scenarios [1]. This approach is receiving ongoing attention in engineering education, both at the high-school [2, 3] and higher education levels [4, 5]. Recently, emphasis has been placed on the importance of systems thinking and abstract thinking among high-school [6, 7] and undergraduate students [8, 9]. Systems thinking provides a framework for examining the interrelations between system's components [10]. Abstract thinking makes it possible to focus on the details relevant to the current viewpoint, while temporarily ignoring the less significant information at the stage in question [11]. Both types of thinking are perceived as essential for the analysis and design of engineering systems [12, 13]. Therefore, advancing systems thinking and abstract thinking in high-school students may enable them to effectively cope with complex engineering tasks given as part of project-based learning.

In light of the recently reported positive correlation between systems thinking and abstract thinking [14], the present study examined whether these two types of thinking could be promoted *simultaneously* among high-school electronics students through project-based learning that included dedicated tasks. Indeed, some studies have indicated an improvement in each of the two types of thinking separately [15, 16]. However, to the best of the authors' knowledge, a simultaneous advancement in both thinking skills has not yet been explored.

The contribution of this study is twofold. Theoretically, the research expands the body of knowledge on systems thinking and abstract thinking. Practically, the study may enhance the development of educational activities promoting the two types of thinking, both in high schools and in higher education.

The paper opens with a review of the theoretical foundations on which the research is based, namely project-based learning, systems thinking and abstract thinking. Next, the research questions are formulated, and the research methodology is described. After presenting and discussing the main findings, some conclusions are drawn.

2. Theoretical Background

2.1 Project-based Learning

As mentioned, project-based learning is a pedagogical approach in which students deal with complex tasks taken from real-world scenarios [1]. Under the guidance of the teacher, the learner copes with the tasks both individually and as a team member, and enjoys a great deal of autonomy [17, 18].

The roots of project-based learning lie in Kilpatrick's [19] and Dewey's [20] concept of "learning by doing". This idea is reflected in constructionism, according to which effective learning occurs when the student builds a product that he/she can share with others [21].

Project-based learning has many benefits. In the cognitive realm, it develops higher-order thinking skills, such as systems thinking [22] and critical thinking [23]. From the affective perspective, pro-

* Accepted 8 March 2021.

ject-based learning creates interest and thus strengthens the student's and teacher's intrinsic motivation [24, 25]. Finally, in the social aspect, it promotes students' interpersonal skills, such as communication and teamwork [26].

At the same time, critics point to the challenges associated with project-based learning [27]. Among other findings, the following challenges have been reported: considerable investment of time compared to traditional learning, both on the parts of the student and the teacher [28]; learners' difficulty in coping with complex tasks (cognitive domain) [29]; teachers' difficulty in adapting to guidance (affective domain) [30]; and students' difficulty in adapting to teamwork, which is sometimes new to them (social domain) [31].

Many studies have characterized project-based learning in engineering education [3, 4]. For instance, project-based learning implemented in transportation engineering has improved students' higher-order thinking skills as well as interpersonal skills [32]. Similarly, a project-based engineering curriculum has fostered academic motivation in undergraduate students [33].

2.2 Systems Thinking and Abstract Thinking

Systems thinking, rooted in general system theory [34], constitutes a framework for observing the interrelations between system's components [10]. In view of the growing complexity of engineering systems, the role of engineering systems thinking is becoming more central [35]. Its importance is further validated in the context of Industry 4.0, which is based on cyber-physical systems, Internet of Things and big data [36].

It is common to classify the characteristics of the so-called systems thinker into four groups: knowledge and background, cognitive characteristics, capabilities, and interpersonal skills [6, 37]. The knowledge and background category includes, inter alia, interdisciplinary knowledge. Cognitive characteristics refer, among other things, to comprehending the functioning of the system without the need for all the details, understanding the interrelations between the system's components, and taking into account non-traditional engineering considerations (e.g., economic, organizational, etc.). The capabilities category includes the ability to perform requirements analysis, formulate the solution in general and perform functional analysis and optimization. Finally, interpersonal skills refer to good human relations and team leadership.

In engineering and technology education, efforts have been made to promote systems thinking at different levels, e.g., high schools [38, 39], two-year colleges [40] and academia [8]. This activity is manifested in various scopes, from specific courses aimed at imparting systems thinking skills [41] to three-year programs [15]. Thus, for example, studies conducted among electrical [28], mechanical [42] and industrial [43] engineering students found that systems thinking can be developed as part of active learning carried out in teams. Similarly, at the high-school level, it was found that a course focusing on the design of a control system by teams advanced twelve grade students' systems thinking [38].

Abstract thinking makes it possible to focus on the details relevant to the current viewpoint, while temporarily ignoring the less significant information for the stage in question [11]. As such, it is relevant for engineers from a variety of fields, including hardware and software engineers [44].

Abstraction level is the degree of complexity in which a system is observed [45]. An arbitrary number of abstraction levels can be defined between the highest level, in which the system is examined from a global perspective, and the lowest level where the attention to details is maximal. In hardware, for example, computer architecture is considered as a relatively high abstraction level while logic gates constitute a relatively low abstraction level [46]. Below the logical level is the device level, which focuses on the electrical properties of the transistors constructing the logic gate. Below the device level is the layout level that deals with the physical structure of the transistors themselves.

Similarly, in software one can define three relatively high abstraction levels, the objective level, describing the software objectives; the conceptual level, referring to the software as an information processing system; and the functional level, characterizing functional structures, states and processes (e.g., data structures and algorithms). Below them are the lower levels, i.e., the logical level, which focuses on the logical meaning of software objects and the physical level, dealing with the syntactic knowledge needed for code writing [47].

In view of its importance, many studies have focused on the characterization and development of abstract thinking among high-school [7, 16] and undergraduate students [9, 48]. From these and other studies, one can formulate the features of the so-called abstract thinker. These properties can be classified into three groups: knowledge and background, cognitive characteristics and capabilities [13, 47]. The first group is based on relevant education. The second category refers to the identification of the appropriate abstraction level for a given phase of problem solving and switching between abstraction levels. The third group (in the context of software engineering) includes, among other things, software requirements analysis, building structure charts and algorithm development.

For the sake of simplicity, for both types of thinking, from now on we merge the two categories focusing on cognition, i.e., cognitive characteristics and capabilities, into one category, namely cognitive skills.

Recently, a significant moderate positive correlation has been found between systems thinking and abstract thinking among high-school electronics students. It turns out that the systems thinker and the abstract thinker share common cognitive skills, but differ in characteristics related to knowledge and background and interpersonal skills. In the cognitive realm, the analogy between the two types of thinking encompasses three skills: requirements analysis, seeing the overall picture, and comprehending the interaction and interdependence between components [14]. It should be stressed, however, that this analogy is partial. For instance, while systems thinking emphasizes the importance of understanding the functioning of a system without the need for all the details [10], abstract thinking is characterized by switching between abstraction levels, including low levels where great attention to detail is essential [11].

3. Research Goal and Questions

The study examined whether systems thinking and abstract thinking of twelve grade electronics students could be promoted simultaneously through project-based learning that included dedicated assignments.

The following research questions were formulated:

- Did a change occur in students' systems thinking (cognitive skills) via project-based learning that included dedicated assignments? If so – what characterized this change?
- Did a change occur in students' abstract thinking (cognitive skills) via project-based learning that included dedicated assignments? If so – what characterized this change?

4. Methodology

4.1 Participants

The study involved 36 twelve grade students who studied electronics at a school in northern Israel. The characteristics of the students were similar to those of Israeli twelve grade students majoring in electronics.

4.2 Intervention

During the school year, students executed, in teams of two students and under the guidance of an experienced teacher, their final project. The project focused on the design and implementation of a system combining hardware and software. The product was based on a programmable device, such as PC or Arduino. In addition, the product included hardware components, such as sensors and displays. The project was carried out in an electronics laboratory, equipped with measuring equipment (e.g., oscilloscopes and multi-meters). At the beginning of the year, the team submitted the project proposal, and at the end – the final report. Examples of final projects are given in Appendix A.

During the year, 14 assignments dealing with systems thinking and abstract thinking were integrated into the curriculum (i.e., the project-based learning described above). These dedicated tasks were based on the systems thinker's and abstract thinker's cognitive skills, described in Section 2.2 and adapted for high-school students majoring in electronics. The features and tasks were validated by two engineering education experts and five electronics teachers with extensive experience in guiding projects combining hardware and software. All assignments were team-based, the time allotted for completion each of them was about two weeks, and they were evaluated by the teacher (formative assessment). In order to cope with some of the challenges involved in project-based learning (Section 2.1), the assignments were of increasing difficulty, and the teacher had many years of experience in guiding projects.

It is important to note that students' knowledge did not allow maneuvering between lower abstraction levels in hardware, such as the layout level and the device level (Section 2.2). However, their knowledge did permit switching between abstraction levels in software, including the lower ones. Therefore, the tasks on abstract thinking were limited to the context of software. In contrast, students' knowledge enabled the assignments on systems thinking to deal with both hardware and software.

Table 1 shows sample tasks, and Appendix B describes two assignments in detail. As explained in Section 2.2, the analogy between systems thinking and abstract thinking is partial and encompasses three cognitive skills (first three rows in Table 1).

4.3 Procedure

The study used both quantitative and qualitative tools. As described in Section 4.2, during the school year, students executed projects that combined hardware and software. Fourteen assignments dealing with systems thinking and abstract thinking were incorporated into the curriculum. At the beginning and end of the year, students took an achievement test. This multiple-choice test was designed to evaluate students' systems and abstract

Table 1. Systems thinking and abstract thinking: sample assignments

Skill	Systems thinking assignment	Abstract thinking assignment
Requirements analysis	Formulate the system requirements	Formulate the software requirements
Seeing the overall picture	Formulate the general concept of the system	Describe the software from the conceptual perspective
Understanding the interaction and interdependence between components	Build a block diagram and explain the interrelations between components	Build a structure chart and explain the interrelations between modules
Taking into account non-traditional engineering considerations	Address relevant economic/ environmental issues	–
Paying attention to details when needed	–	Describe the software in detail

thinking (cognitive skills). At the end of the year, the final reports were collected.

Since the correlation between systems thinking and abstract thinking is moderate [14], the quantitative data were analyzed using one-way repeated measures MANOVA. The qualitative data were classified into categories using directed content analysis [49]. The analysis was based on the systems thinker's and abstract thinker's cognitive skills reviewed in Section 2.2 and adapted for high-school electronics students.

4.4 Tools

The achievement test dealt with the analysis of a system opening and closing a parking lot gate. This system was not part of a final project of any of the teams. The test included 18 multiple-choice questions (one correct answer and three distractors), half of which focused on systems thinking and half – on abstract thinking. The questions were of equal value. The test lasted one hour, and no auxiliary material could be used. In order to reduce the effect of early exposure, the test was given in two versions, one at the beginning of the school year (pretest) and the other at the end (posttest). There were slight differences between the versions (e.g., numeric values, names of variables, etc.). The test (two versions) was validated by two experts in engineering education. A sample of the questions appears in Appendix C.

5. Findings

Table 2 shows students' mean score *M* (ranging between 0 and 100) and standard deviation *SD* in the pretest and posttest.

One-way repeated measures MANOVA revealed a significant difference in students' scores between the beginning and end of the school year ($F(2, 34) = 11.50$, $p < 0.01$; Wilks' $\Lambda = 0.60$, partial $\eta^2 = 0.40$). Univariate tests indicated a significant difference in both systems thinking ($p < 0.01$; partial $\eta^2 = 0.25$) and abstract thinking scores ($p < 0.01$; partial $\eta^2 = 0.37$). The improvement in both types of thinking was characterized by a large effect size.

Table 2. Systems thinking and abstract thinking: descriptive statistics (achievement test)

		Systems thinking		Abstract thinking	
Instrument	***N***	***M***	***SD***	***M***	***SD***
Pretest	36	55.53	20.82	52.76	17.00
Posttest	36	66.80	12.00	66.54	16.57

Analysis of the qualitative data (final reports) shows that at the end of the school year the students adopted some of the systems thinker's and abstract thinker's skills. Thus, for example, students performed requirements analysis:

> "The robot should be easy to move from place to place." (systems thinking)
>
> "The software should be immune to errors that may occur during reasonable use, such as improper input." (abstract thinking)

They saw the overall picture:

> "The robot moves in four directions (forward, backward, right and left) depending on the glove movement worn on the user's hand." (systems thinking)
>
> "The software provides relevant information on the condition of the plant (i.e., light intensity, temperature and soil moisture) as a function of time." (abstract thinking)

and understood the interaction and interdependence between components:

> "The DHT22 sensor measures temperature and humidity and transmits the values, using the One Wire Protocol, to the microcontroller." (systems thinking)
>
> "The function *Tracking_time* returns the current time to the main function." (abstract thinking)

In addition, students took into account non-traditional engineering considerations (systems thinking):

> "The product is cheap compared to similar products in the market."

and paid attention to details when needed (abstract thinking):

> "The function *Soil_humidity* reads values from the soil moisture sensor, converts them to percentiles by dividing them to 1024, then multiplying by 100."

The findings are summarized in Table 3.

Table 3. Systems thinker's and abstract thinker's skills among students (end of year)

Skill	Systems thinking	Abstract thinking
Requirements analysis	"The robot should be resistant to malfunctions that may occur during reasonable use."	"The software should be easy to use even for the unprofessional user."
Seeing the overall picture	"The robot moves on four wheels and its movement is controlled by a smartphone via wireless communication."	"The software calculates the distance of nearby objects and alerts when a particular object is too close."
Understanding the interaction and interdependence between components	"The microcontroller receives data from the Bluetooth receiver according to the UART protocol."	"The function *Water heating* gets the water temperature (by the *temp* variable) from the function *Water temperature measurement*."
Taking into account non-traditional engineering considerations	"The product is made of environmentally friendly material."	–
Paying attention to details when needed	–	"The function reads values from the sensor. If the value is bigger than 380, then all the wheels should move forward; if the value is smaller than 310, then all the wheels should move backward."

6. Discussion

Both systems thinking and abstract thinking are essential for the analysis and design of engineering systems [12, 13]. Thus, promoting systems thinking and abstract thinking in high-school students may enable them to effectively cope with complex engineering assignments given as part of project-based learning. In view of the recently reported positive correlation between systems thinking and abstract thinking [14], the present study examined whether these two types of thinking could be developed simultaneously through project-based learning that included dedicated tasks. According to the findings, a significant improvement was achieved in both systems and abstract thinking. This improvement was characterized by a large effect size.

It was also found that at the end of the year students adopted some of the systems thinker's [6, 37] and abstract thinker's skills [13, 47]. Some of these abilities were common to both types of thinking, i.e., requirements analysis, seeing the overall picture, and understanding the interaction and interdependence between components. Other skills promoted were unique to systems thinking (e.g., taking into account non-traditional engineering considerations) and abstract thinking (e.g., paying attention to details when needed). The test results and quotes taken from the final reports indicated that at the end of the school year students' levels of systems thinking and abstract thinking were moderate on average.

The literature mentions successful attempts of promoting systems thinking among engineering students [42, 43] and even earlier [38, 39]. These studies suggest that systems thinking can be developed in an active learning environment, such as the one in the present study. Similarly, many studies have focused on advancing abstract thinking among high-school [7, 16] and undergraduate students [9, 48]. However, to the best of the authors' knowledge, it has not yet been reported that the two types of thinking can be promoted simultaneously, as found here. We believe that this result, obtained at the high-school level, is applicable in higher education as well. A thorough study should be conducted to verify this hypothesis.

The present study had two main limitations: (a) a relatively small number of participants, and (b) lack of a control group. Both limitations were due to the low number of twelve grade electronics students at the school where the study took place. Therefore, it was not possible to form a control group of a reasonable size. To overcome these limitations and increase the findings' trustworthiness, qualitative tools were used alongside quantitative ones.

The contribution of this study is twofold. Theoretically, the research expands the body of knowledge on systems thinking and abstract thinking. Practically, the study may enhance the development of educational activities promoting the two types of thinking, both in high schools and in higher education. Thus, for example, it might be appropriate to give priority to programs that advance the two skills simultaneously over programs that focus on each type of thinking separately. These contributions are further validated in light of the many efforts invested in developing such programs [15, 16].

7. Conclusions

The study indicates that systems thinking and abstract thinking can be promoted simultaneously among twelve grade electronics students via project-based learning that includes dedicated assignments. In addition to a significant positive change (large effect) achieved in both types of thinking, it was found that the students adopted some of the systems thinker's and abstract thinker's skills.

References

1. D. Kokotsaki, V. Menzies and A. Wiggins, Project-based learning: A review of the literature, *Improving Schools*, **19**(3), pp. 267–277, 2016.
2. J. Edmunds, N. Arshavsky, E. Glennie, K. Charles and O. Rice, The relationship between project-based learning and rigor in STEM-focused high schools, *Interdisciplinary Journal of Problem-Based Learning*, **11**(1), article 3, 2017.
3. F. Tauro, Y. Cha, F. Rahim, M. Sattar Rasul, K. Osman, L. Halim, D. Dennisur, B. Esner and M. Porfiri, Integrating mechatronics in project-based learning of Malaysian high school students and teachers, *International Journal of Mechanical Engineering Education*, **45**(4), pp. 297–320, 2017.
4. R. Savage, K. Chen and L. Vanasupa, Integrating project-based learning throughout the undergraduate engineering curriculum, *Journal of STEM Education*, **8**(3–4), pp. 15–27, 2008.
5. M. Frank, I. Lavy and D. Elata, Implementing the project-based learning approach in an academic engineering course, *International Journal of Technology and Design Education*, **13**(3), pp. 273–288, 2003.
6. M. Frank and S. Kordova, Developing systems thinking through engaging in multidisciplinary high-tech projects, *International Journal of Project Organization and Management*, **5**(3), pp. 222–238, 2013.
7. S. Grover and R. Pea, Computational thinking in K–12: A review of the state of the field, *Educational Researcher*, **42**(1), pp. 38–43, 2013.
8. N. J. Hayden, D. M. Rizzo, M. Dewoolkar, M. D. Neumann, S. Lathem and A. Sadek, Incorporating a systems approach into civil and environmental engineering curricula: Effect on course redesign, and student and faculty attitudes, *Advances in Engineering Education*, **2**(4), pp. 1–27, 2011.
9. H. Koppelman and B. van Dijk, Teaching abstraction in introductory courses, *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, ACM, pp. 174–178, 2010.
10. P. M. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday, New York, 1990.
11. B. Timothy, *Introduction to Object-Oriented Programming*, Pearson Education, India, 2008.
12. R. M. Jaradat, Complex system governance requires systems thinking-how to find systems thinkers, *International Journal of System of Systems Engineering*, **6**(1–2), pp. 53–70, 2015.
13. J. Kramer and O. Hazzan, The role of abstraction in software engineering, *Proceedings of the 28th International Conference on Software Engineering*, ACM, pp. 1017–1018, 2006.
14. A. Gero, A. Shekh-Abed and O. Hazzan, Interrelations between systems thinking and abstract thinking: The case of high-school electronics students, *European Journal of Engineering Education*, Published online, 2020.
15. A. Gero and E. Zach, High school programme in electro-optics: A case study on interdisciplinary learning and systems thinking, *International Journal of Engineering Education*, **30**(5), pp. 1190–1199, 2014.
16. V. Barr and C. Stephenson, Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community?, *ACM Inroads*, **2**(1), pp. 48–54, 2011.
17. J. Macías-Guarasa, J. M. Montero, R. San-Segundo, Á. Araujo and O. Nieto-Taladriz, A project-based learning approach to design electronic systems curricula, *IEEE Transactions on Education*, **49**(3), pp. 389–397, 2006.
18. D. Cridmond, Scaffolding strategies for integrating engineering design and scientific inquiry in project-based learning environments, in M. Barak and M. Hacker (eds), *Fostering Human Development Through Engineering and Technology Education*, Sense Publishers, Rotterdam, pp. 235–255, 2011.
19. W. H. Kilpatrick, The project method, *Teachers College Record*, **19**(4), pp. 319–335, 1918.
20. J. Dewey, *Experience and Education*, Macmillan, New York, 1938.
21. S. Papert and I. Harel, *Constructionism: Research Reports and Essays*, Ablex Publishing Corporation, 1991.
22. S. Nagarajan and T. Overton, Promoting systems thinking using project-and problem-based learning, *Journal of Chemical Education*, **96**(12), pp. 2901–2909, 2019.
23. W. Pan and J. Allison, Exploring project based and problem based learning in environmental building education by integrating critical thinking, *International Journal of Engineering Education*, **26**(3), pp. 547–553, 2010.
24. C. L. Chiang and H. Lee, The effect of project-based learning on learning motivation and problem-solving ability of vocational high school students, *International Journal of Information and Education Technology*, **6**(9), pp. 709–712, 2016.
25. S. F. Lam, R. W. Y. Cheng and W. Y. Ma, Teacher and student intrinsic motivation in project-based learning, *Instructional Science*, **37**(6), article 565, 2009.
26. J. S. Vogler, P. Thompson, D. W. Davis, B. E. Mayfield, P. M. Finley and D. Yasseri, The hard work of soft skills: Augmenting the project-based learning experience with interdisciplinary teamwork, *Instructional Science*, **46**(3), pp. 457–488, 2018.
27. D. G. Lamar, P. F. Miaja, M. Arias, A. Rodriguez, M. Rodriguez and J. Sebastian, A project-based learning approach to teaching power electronics: Difficulties in the application of project-based learning in a subject of Switching-Mode Power Supplies, *IEEE EDUCON 2010 Conference*, pp. 717–722, 2010.
28. A. Gero, Enhancing systems thinking skills of sophomore students: An introductory project in electrical engineering, *International Journal of Engineering Education*, **30**(3), pp. 738–745, 2014.
29. B. Catz, N. Sabag and A. Gero, Problem based learning and students' motivation: The case of an electronics laboratory course, *International Journal of Engineering Education* **34**(6), pp. 1838–1847, 2018.
30. A. Habók and J. Nagy, In-service teachers' perceptions of project-based learning, *SpringerPlus*, **5**(1), article 83, 2016.
31. A. C. Alves, D. Mesquita, F. Moreira and S. Fernandes, Teamwork in project-based learning: Engineering students' perceptions of strengths and weaknesses, *Proceedings of the Fourth International Symposium on Project Approaches in Engineering Education*, pp. 23–32, 2012.
32. E. H. Fini, F. Awadallah, M. M. Parast and T. Abu-Lebdeh, The impact of project-based learning on improving student learning outcomes of sustainability concepts in transportation engineering courses, *European Journal of Engineering Education*, **43**(3), pp. 473–488, 2018.
33. M. J. Terrón-López, M. J., García-García, P. J. Velasco-Quintana, J. Ocampo, M. R. Vigil Montaño and M. C. Gaya-López, Implementation of a project-based engineering school: Increasing student motivation and relevant learning, *European Journal of Engineering Education*, **42**(6), pp. 618–631, 2017.

34. L. V. Bertalanffy, *General System Theory: Foundations, Development, Applications*, George Braziller, New-York, 1968.
35. J. P. Monat and T. F. Gannon, Applying systems thinking to engineering and design, *Systems*, **6**(3), article 34, 2018.
36. N. Jazdi, Cyber physical systems in the context of Industry 4.0, *IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1–4, 2014.
37. M. Frank, Assessing the interest for systems engineering positions and other engineering positions' required Capacity for Engineering Systems Thinking (CEST), *Systems Engineering*, **13**(2), 161–174, 2009.
38. A. Gero and O. Danino, High-school course on engineering design: Enhancement of students' motivation and development of systems thinking skills, *International Journal of Engineering Education*, **32**(1A), pp. 100–110, 2016.
39. C. Chalmers and R. Nason, Systems thinking approach to robotics curriculum in schools, In *Robotics in STEM Education*, Springer, Cham, pp. 33–57, 2017.
40. A. Gero and I. Shlomo, Promoting systems thinking in two-year technology students: An interdisciplinary course on medical ultrasound systems, *International Journal of Engineering Education*, **37**(2), pp. 564–572, 2021.
41. F. Camelia, T. L. Ferris and M. B. Behrend, The effectiveness of a systems engineering course in developing systems thinking, *IEEE Transactions on Education*, **63**(1), pp. 10–16, 2019.
42. C. M. Degen, K. H. Muci-Küchler, M. D. Bedillion, S. Huang and M. Ellingsen, Measuring the impact of a new mechanical engineering sophomore design course on students' systems thinking skills, *International Mechanical Engineering Congress and Exposition*, American Society of Mechanical Engineers Digital Collection, 2018.
43. S. Koral-Kordova and M. Frank, Improving capacity for engineering systems thinking (CEST) among industrial engineering students, *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1378–1380, 2012.
44. O. Mishali, Y. Dubinsky and I. Maman, Towards IDE support for abstract thinking, *Proceedings of the 2nd International Workshop on the Role of Abstraction in Software Engineering*, pp. 9–13, 2008.
45. B. Liskov and J. Guttag, *Abstraction and Specification in Program Development*, MIT press, Cambridge, 1986.
46. J. Sanguinetti, Abstraction and standardization in hardware design, *IEEE Design and Test of Computers*, **29**(2), pp. 8–13, 2012.
47. N. Ye and G. Salvendy, Expert-novice knowledge of computer programming at different levels of abstraction, *Ergonomics*, **39**(3), pp. 461–481, 1996.
48. O. Hazzan, Reflections on teaching abstraction and other soft ideas, *ACM SIGCSE Bulletin*, **40**(2), pp. 40–43, 2008.
49. H. Hsieh and S. Shannon, Three approaches to qualitative content analysis, *Qualitative Health Research*, **15**(9), pp. 1277–1288, 2005.

Appendix A – Final Project

The final project, mentioned in Section 4.2, focused on the design and implementation of a system combining hardware and software. The product was based on a programmable device, such as PC or Arduino. In addition, the product included hardware components, e.g., sensors and displays. Below are two examples.

A.1 Preparation of Baby Milk

The system is based on an Arduino Mega 2560 microcontroller. The user selects the desired amount of milk by pressing a button on the front of the product, or using an app on a smartphone (connected to the system via Bluetooth). The system pours into a bottle the amount of milk powder required as well as the amount of water at a suitable temperature. Relevant messages are displayed on the product screen and at the same time in the app on the smartphone.

A.2 Self-balancing Robot

The robot is based on an Arduino Nano microcontroller. The robot, having only two wheels, has an acceleration and angle sensor installed to maintain balanced movement. In addition, the robot is fitted with an ultrasonic sensor whose function is to detect obstacles in the direction of movement. The robot's movement (forward, backward, right, left and stop) is controlled from a smartphone via Bluetooth.

Appendix B – Tasks

The tasks designed for promoting systems thinking and abstract thinking (Section 4.2) were based on the cognitive skills of the systems thinker and abstract thinker, described in Section 2.2 and adapted for high-school electronics students. Below are two tasks focusing on understanding the interaction and interdependence between components. The first assignment deals with systems thinking and the second with abstract thinking. Both tasks refer to the system for preparing baby milk, described in Appendix A.1.

B.1 Building a Block Diagram

The team should build a block diagram and explain the interdependence between the components. Fig. B1 shows a possible solution. According to the solution, the user selects the desired amount of milk using one of the four buttons on the front of the product or using a dedicated app on a smartphone (connected to the system via a Bluetooth receiver).

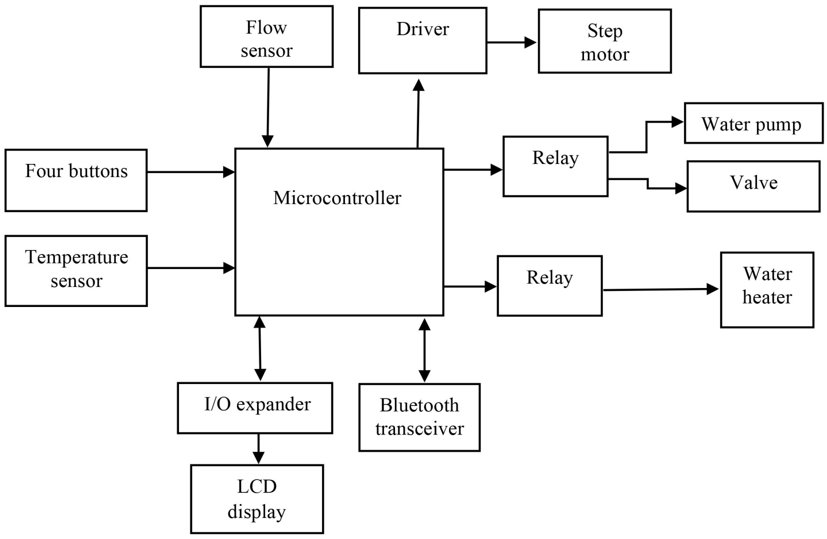

Fig. B1. Preparation of baby milk – block diagram.

The microcontroller receives the water temperature from the temperature sensor. If the temperature is below 32C (suitable for the baby), then the microcontroller sends a signal to the relay to activate the heating element. If the water temperature is higher than 32C, then the system sends an alert and waits until the temperature drops to 32C. If the water temperature is 32C, there is no need for heating or waiting.

When the temperature reaches 32C, the microcontroller sends a signal to the driver to operate the step motor. The latter injects the required amount of milk powder into a bottle. Additionally, the microcontroller sends a signal to the relevant relay to open the valve and operate the water pump. The amount of water flowing into the bottle is determined by the amount of milk selected by the user and measured by the flow sensor.

The system messages are transmitted to the LCD display via the I/O expander. The I/O expander sends back acknowledge (ACK) bits to indicate that the messages have been received successfully. In addition, the messages are sent to the smartphone via the Bluetooth transmitter.

B.2 Building a Structure Chart

The team should build a structure chart and explain the interdependence between the functions. Fig. B2 shows a possible solution. According to the solution, the main function "Preparation of baby milk" calls four secondary functions in series. First, the function "Amount choice" returns the amount of milk selected by the user to the main function (via the variable "amount"). Then, the function "Water heating" is activated, depending on the water temperature it receives (via the variable "temp") from the function "Water temperature measurement". Next, the function "Adding milk powder" is activated, depending on the amount of milk it receives (via the variable "amount") from the main function. Finally, the function "Adding water" is activated, according to the amount of milk it receives (via the variable "amount") from the main function and the amount of water in the bottle which it receives (via the variable "water_amount") from the function "Water amount measurement".

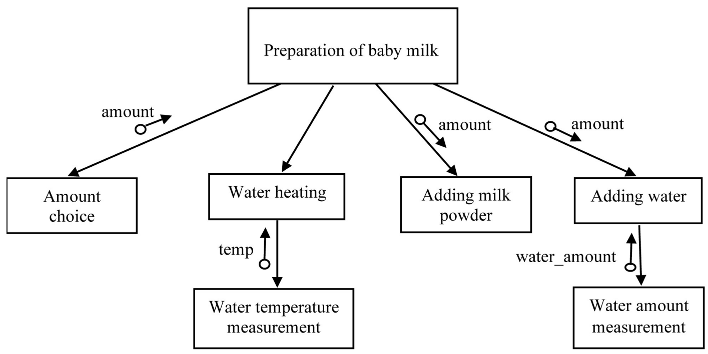

Fig. B2. Preparation of baby milk – structure chart.

Appendix C – Achievement Test

The test for evaluating systems thinking and abstract thinking, mentioned in Section 4.4, was a multiple-choice test (one correct answer and three distractors). It included 18 questions, half of which dealt with systems thinking and half with abstract thinking. The test was given in two versions, one at the beginning of the school year (pretest) and the other at the end (posttest). There were slight differences between the versions (e.g., numeric values, names of variables, etc.). The test focused on analyzing a system combining hardware and software, as described below:

> "At a certain school, the principal requested the technician to install a system that permits the opening and closing of the school parking lot gate by dialing from a mobile phone. Each member of the school staff wishing to open the parking lot gate, would dial a certain telephone number and the gate would open by a 24V DC motor. After the gate opens, the system would wait for 12 seconds, and if it discovered through a proximity sensor that no one was going through the gate, the gate would close. Two position sensors would be installed in proximity to the gate. One sensor would signal to the system that the gate was completely closed, and the other sensor would signal the system that the gate was completely open.
>
> In the software, the main function calls the following secondary functions:
>
> - Car_Gate – The function checks whether a vehicle crosses the gate;
> - Order_Gate – The function checks whether a gateway request is received;
> - Con_Gate – The function checks the gate status (fully open, fully closed or partially open);
> - Open_CloseGate – The function opens/closes the gate."

Two questions are given below. The first question deals with systems thinking and the second with abstract thinking.

1. In which case should the motor be connected to an electric relay instead of a current driver?
   A. When a step motor is used instead of a DC motor.
   B. When a servo motor is used instead of a DC motor.
   C. When a DC motor is used instead of an AC motor.
   D. When an AC motor is used instead of a DC motor.
2. In Fig. C1, a software code is given. Which line has a syntax error?
   A. 13
   B. 14
   C. 18
   D. 27

```
1.  int val = 0;
2.  int Car_Gate();
3.  void Open_CloseGate();
4.  void setup() {
5.    pinMode(4, OUTPUT);
6.    pinMode(5, OUTPUT);
7.  }
8.  void loop() {
9.    Open_CloseGate();
10. }
11. void Open_CloseGate()
12.   int st = Car_Gate();
13.   if(st==1) {
14.     digitalWrite(4, 0);
15.     digitalWrite(5, 1);
16.     delay(10000);
17.   }
18.   else if(st=0) {
19.     digitalWrite(4, LOW);
20.     digitalWrite(5, HIGH);
21.     delay(10000);
22.   }
23. }
24. Car_Gate() {
25.   int s;
26.   val = analogRead(A0);
27.   if(val>200)
28.     s = 1;
29.   else
30.     s = 0;
31.   return s;
32. }
```

Fig. C1. Achievement test – software code

Aziz Shekh-Abed holds a PhD in engineering education from the Technion – Israel Institute of Technology, Haifa, Israel. His research thesis dealt with systems thinking and abstract thinking in high-school students, and was done under the supervision of Aharon Gero and Orit Hazzan. Dr Shekh-Abed holds an MA in science education and a BSc in technology education, both from Tel Aviv University, Tel Aviv, Israel. He is an adjunct lecturer in the Department of Electrical and Computer Engineering at Ruppin Academic Center, Emek Hefer, Israel. Dr Shekh-Abed has about 20 years of experience in teaching electrical and computer engineering at high schools.

Orit Hazzan is a professor in the Technion's Department of Education in Technology and Science since October 2000. Her research focuses on computer science, software engineering and data science education. Within this framework she researches cognitive, social and managerial processes on the individual, the team and the organization levels, in all kinds of organizations. She has published about 120 papers in professional refereed journals and conference proceedings, and seven books. In 2006–2008 she served as the Technion's Associate Dean of Undergraduate Studies. In 2007–2010 she chaired the High School Computer Science Curriculum Committee assigned by the Israeli Ministry of Education. In 2011–2015 Hazzan was the faculty Dean. From 2017 to 2019, Hazzan served as the Technion's Dean of Undergraduate Studies. Additional details can be found in her personal homepage: https://orithazzan.net.technion.ac.il/

Aharon Gero holds a BA in physics (*Summa Cum Laude*), a BSc in electrical engineering (*Cum Laude*), an MSc in electrical engineering, and a PhD in theoretical physics, all from the Technion – Israel Institute of Technology, Haifa, Israel. In addition, he has an MBA (*Cum Laude*) from the University of Haifa, Israel. Dr Gero is an Assistant Professor in the Department of Education in Technology and Science at the Technion, where he heads the Electrical Engineering Education Research Group. Before joining the Technion, he was an instructor at the Israeli Air-Force Flight Academy. Dr Gero's research focuses on electrical engineering education and interdisciplinary education that combines physics with electronics, at both the high school and higher education levels. Dr Gero has received the Israeli Air-Force Flight Academy Award for Outstanding Instructor twice and the Technion's Award for Excellence in Teaching 15 times. In 2006, he received the Israeli Air-Force Commander's Award for Excellence, and in 2016 was awarded the Yanai Prize for Excellence in Academic Education. From September 2016 through February 2017 Dr Gero was a Visiting Scholar at the School of Engineering Education, Purdue University, Indiana, US, and from March 2017 through August 2017 he was a Visiting Scholar at the School of Education, Tel Aviv University, Israel.