International Journal of Engineering Education Vol. 39, No. 1, pp. 154–162, 2023
Printed in Great Britain

0949-149X/91 $3.00+0.00

Learning and Instruction that Combine Multiple Levels of Abstraction in Engineering: Attitudes of Students and Faculty*

MOHAMMED ALI HADISH[1], SHAHAR KVATINSKY[2] and AHARON GERO[1]
[1] Department of Education in Technology and Science, Technion – Israel Institute of Technology, Haifa 32000, Israel.
[2] The Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel. E-mail: hadish16@campus.technion.ac.il, shahar@ee.technion.ac.il, gero@technion.ac.il

One type of thinking needed by engineers is abstract thinking, i.e., higher-order thinking that permits one to solve problems while maneuvering between several levels of complexity (levels of abstraction). In light of the above, the Faculty of Electrical and Computer Engineering (Technion – Israel Institute of Technology) decided to combine two introductory courses, focusing on different levels of abstraction, into a single undergraduate course "Digital Systems and Computer Structure" integrating multiple levels of abstraction, i.e., logical, micro-architecture and architecture levels. The study presented here characterized the attitudes of students and course faculty toward learning and instruction that combine several levels of abstraction. The research, which used quantitative and qualitative tools, involved 103 students and eight teaching staff members. According to the findings, students hold positive attitudes toward learning that incorporates multiple levels of abstraction, both cognitively and affectively, and the correlation between the components is positive, moderate, and significant. Students argue that this type of learning is interesting, provides a complete picture of computer systems, promotes higher-order thinking, and is relevant to industry work, but is also characterized by a high cognitive load. Course faculty claim that teaching that incorporates multiple levels of abstraction is enjoyable, imparts higher-order thinking among students but is very demanding. As for the behavioral aspect, the vast majority of students and instructors prefer learning and teaching that integrate multiple levels of abstraction over those that focus on a few levels.

Keywords: abstract thinking; students' attitudes; instructors' attitudes; digital systems; computer structure

1. Introduction

Alongside the lingering shortage of engineers in the Western world, including Israel [1, 2], there is a notable gap between the skills of engineering graduates and those needed in the industry [3]. To provide a partial solution to this gap, the Accreditation Board for Engineering and Technology (ABET) has formulated the capabilities required of engineering graduates. Among other things, the engineer should identify, formulate, and solve complex engineering problems, perform engineering design, and develop and conduct experimentation [4].

One type of thinking underlying the abilities mentioned above is abstract thinking [5]. Abstract thinking is the capability to focus on the significant details of a given stage while temporarily ignoring the information that is less relevant to the current phase [6]. Abstract thinking is higher-order thinking that permits one to solve problems while maneuvering between multiple levels of abstraction [7], where a level of abstraction is the degree of complexity in which the problem is examined [8]. Abstract thinking is relevant in a wide range of fields, such as mathematics, science, engineering, and business administration [9]. It is of particular importance in hardware and software engineering, which requires an examination of a variety of topics at different levels of detail [10]. For instance, in chip design it is necessary to assign billions of elements, and the only feasible way to design such a complex system is by abstraction. Most experts agree that the key principles of abstract thinking are identifying the level of abstraction relevant to a given stage and the ability to move between levels of abstraction [11, 12].

In light of the above, the Faculty of Electrical and Computer Engineering (Technion – Israel Institute of Technology) decided to combine two basic courses, dealing with different levels of abstraction, into a single undergraduate course integrating multiple levels of abstraction. The course "Digital Systems", which focused on Boolean algebra and the analysis and design of combinational and sequential systems, and the course "Logic Design and Introduction to Computers", which dealt with advanced logic design and computer structure, were combined into the mandatory course "Digital Systems and Computer Structure". The latter covers three levels of abstraction, i.e., logical, micro-architecture and architecture levels and switches between them. It is important to emphasize that in most of the world's leading universities, the courses corre-

* Accepted 10 October 2022.

sponding to "Digital Systems" and "Logic Design and Introduction to Computers" are taught separately. However, the University of California, Berkeley offers a unified course similar in content to the above-mentioned course [13].

The research described here characterized, using quantitative and qualitative tools, the attitudes of students and course faculty toward learning and instruction that combine multiple levels of abstraction. To the best of the authors' knowledge, such characterization was performed here for the first time. The findings expand the body of knowledge on the subject and may promote the training of engineers.

The paper opens with a theoretical background that reviews computational thinking in general and abstract thinking in particular. Next, the course "Digital Systems and Computer Structure" is described. Then, the research goal and methodology are presented. Finally, the main findings are discussed.

2. Computational Thinking and Abstract Thinking

Computational thinking, a term coined by Papert [14], refers to the thought processes involved in problem-solving in general and in performing computational research in particular [15]. Thus, computational thinking is relevant in a broad range of disciplines beyond computer science and software engineering [16]. Moreover, computational thinking has been identified, both in Europe [17] and in the United States [18], as one of the 21st century skills required to function effectively in modern society.

Most researchers agree that the four main components of computational thinking are [19]:

- Decomposition – separating the problem into sub-problems or simpler components.
- Pattern Recognition – finding similarities and differences between the current problem (and its components) and similar problems known from experience.
- Abstract Thinking – focusing on details relevant to the current stage and temporarily ignoring the unnecessary information.
- Algorithmic Thinking – formulating a solution based on a series of well-defined steps.

In addition to the four elements outlined above, some scholars mention the following components: problem formulation, proposing and evaluating several solutions, and generalizing the chosen solution [15].

Abstract thinking, as an element of computational thinking, is relevant in various areas [9], and currently plays a central role in engineering [10]. This is due to two main reasons: the growing intricacy of engineering systems [20], and the framework of Industry 4.0, which is based, among other things, on big data [21]. Thus, engineers must deal with increased complexity, particularly in the hardware and software industry [22].

A theoretical basis for abstract thinking was proposed by Piaget in his famous theory of cognitive development. According to which, abstract thinking is supposed to evolve in the fourth and final phase of cognitive development, i.e., the formal operations stage (ages 12–15). This phase involves, inter alia, the abilities to use symbols, reflect, reason, and think abstractly [23].

A key concept in abstract thinking is a level of abstraction, where the latter is the level of complexity at which a problem is examined [8]. An arbitrary number of levels can be defined between the highest level, where the problem is observed from a global viewpoint, and the lowest level at which great attention to detail is required [24]. Since abstract thinking is relevant in many areas [9, 10], levels of abstraction can be used in a variety of disciplines, e.g., physics [25], biochemistry [26], and software engineering [12].

For example, the architecture level is the boundary between software and hardware. At this level of abstraction, a set of software instructions that the computer can support is defined. Below this level is the micro-architecture level, dealing with the highest level of realization of the features described at the previous level. The next level, logic gates, focuses on the logical properties of the components defined at the micro-architecture level. Below the logical level are the circuit level, describing the electrical properties of the circuit that realizes the logic gate, and the device level, dealing with the electrical properties of the transistors constructing the circuit. Finally, the layout level focuses on the physical structure of the transistors themselves [27].

It is common to classify the characteristics of the so-called abstract thinker into three groups: knowledge and background, cognitive skills, and capabilities [11, 12]. The first category consists of relevant education. The second group refers to identifying the appropriate level of abstraction for a given stage and maneuvering between levels of abstraction. The last category is comprised of job-specific skills. For example, a software engineer should, among other things, formulate the software requirements and build a structure chart [28]. These features are based on studies conducted among diverse populations, including high-school [29, 30] and university students [24, 31].

Recently, a positive correlation has been found between abstract thinking and systems thinking among high-school students majoring in electronics [32]. It seems that the abstract thinker and systems thinker share cognitive skills but differ in features related to knowledge and background. In addition, there are researchers who point to similarities between abstract thinking and critical thinking [33] and between abstract thinking and creative thinking [16].

3. The Course "Digital Systems and Computer Structure"

As described in Section 1, the Faculty of Electrical and Computer Engineering (Technion – Israel Institute of Technology) decided to combine the course "Digital Systems", focusing on Boolean algebra and the analysis and design of combinational and sequential systems, and the course "Logic Design and Introduction to Computers", dealing with advanced logic design and computer structure, into a single course "Digital Systems and Computer Structure". The latter covers three levels of abstraction, i.e., logical, micro-architecture and architecture levels (Section 2), switches between them, and thus provides a more complete picture of compute structure and functionality.

The undergraduate-level course "Digital Systems and Computer Structure" aims to provide beginning students with analysis and design skills at the three levels of abstraction mentioned above. This mandatory course (five credit points) lasts 13 weeks and is comprised of four hours of lectures and two hours of tutorials every week. Five two-hour workshops are held during the semester, dealing with Verilog programming. The course faculty consists of lecturers and teaching assistants. The teaching method in all sessions is front facing. A prerequisite for the course is the course "Introduction to Computer Science", focusing on the algorithmic approach to problem-solving and programming in C language.

The contents of the course overlap approximately with the contents of the two preceding courses and include: Boolean algebra, the digital model, combinational systems, sequential systems, finite state machines, von Neumann machine, hardware description language, addressing modes, branches, stack, routines, traps, pipelined processor, structural, control and data hazards.

As mentioned, the course covers several levels of abstraction and maneuvers between them. Thus, for example, it is taught how to add binary numbers using a full adder (logical level). The latter is then used as a black box in the arithmetic logic unit (micro-architecture level) to support the add instruction (architecture level). Another example deals with the instruction set architecture (architecture level). The latter is mapped to machine language, and then it is taught how to implement each instruction in different types of processors (micro-architecture level). For each processor type, specific realizations in the logical level are discussed.

The course is based on the textbooks: *Digital Design: A Systems Approach* [34] and *Computer Architecture: A Quantitative Approach* [35]. The assessment is based on computer exercises (simulations in Verilog), a mid-semester test, and a final examination.

For comparison, each of the preceding courses lasted 13 weeks and consisted of two hours of lectures and one hour of tutorial every week. The teaching method and the assessment were similar to those of the new course.

4. Goal

The study characterized students' and instructors' attitudes toward learning and teaching that combine multiple levels of abstraction in engineering. The following questions were formulated:

- What are students' attitudes (cognitive, affective and behavioral aspects) toward learning that integrates several levels of abstraction?
- What are instructors' attitudes (cognitive, affective and behavioral aspects) toward teaching that integrates several levels of abstraction?

5. Methodology

5.1 Participants

One hundred and three undergraduate students (second-third semesters) who attended the course "Digital Systems and Computer Structure" took part in the study. The students have not previously been exposed to learning that combines multiple levels of abstraction. The characteristics of the participants were similar to those of average undergraduate students at the Faculty of Electrical and Computer Engineering.

In addition, members of the course's teaching staff (three lecturers and five teaching assistants) participated in the study. The lecturers held advanced degrees in electrical and computer engineering and had substantial teaching experience. The teaching assistants were senior electrical and computer engineering students and graduate students.

5.2 Method

The research applied quantitative and qualitative instruments to enable the presentation of various perspectives of the phenomenon under study and increase the findings' trustworthiness [36].

Table 1. Closed-ended questionnaire – sample statements

Component	Polarity	Statement
Cognitive	Positive	The combination of digital systems and computer structure is correct because it allows the computer to be analyzed at different levels of detail
	Negative	It is better to study digital systems separately from computer structure so that it will be possible to devote the necessary time to each subject
Affective	Positive	The combination of digital systems and computer structure is interesting because the student can view the study material from several perspectives
	Negative	The combination of digital systems and computer structure is boring because it repeats the same topics several times

At the end of the course, students completed an anonymous closed-ended questionnaire, aimed to characterize their attitudes (cognitive and affective components) toward learning that combines multiple levels of abstraction. Additionally, at the end of the course the students filled out an anonymous open-ended questionnaire and nine of them were interviewed. The open-ended questionnaires and the semi-structured interviews were designed to characterize the respondents' attitudes toward learning that incorporates multiple levels of abstraction. At the end of the course, members of the teaching staff were interviewed (semi-structured interviews) about their attitudes toward instruction that combines several levels of abstraction.

The quantitative data were statistically analyzed. It was first examined, using the Kolmogorov-Smirnov test for normality, whether a normal distribution of the attitude component scores (cognitive and affective) could be assumed. Then, according to the results obtained, the appropriate correlation coefficient between the components was calculated.

The interviews were recorded and transcribed in full. By means of directed content analysis [37], conducted by two experts in engineering education, the qualitative data (open-ended questionnaires and interviews) were classified into categories. The analysis was based on the tri-component attitude model of Rosenberg and Hovland [38]. Only information that has risen at least three times was included in the analysis.

Ethical approval (No. 2020-018) was obtained from the Institutional Review Board.

5.3 Tools

The closed-ended questionnaire was a five-level Likert-like scale, ranging from "strongly disagree" to "strongly agree". The scale was based on a tool suggested by Fishbein [39]. This self-reporting questionnaire was comprised of 17 statements reflecting cognitive and affective aspects toward learning that combines multiple levels of abstraction. Some of the statements expressed positive attitudes and others – negative ones. The statements were validated by two engineering education experts. Cronbach's alphas of the attitude components (α = 0.75, cognitive component; α = 0.71, affective component) pointed to acceptable internal consistency. Some statements are given in Table 1. A sample of the open-ended questions is found in Appendix A and a sample of the interview questions is provided in Appendix B.

6. Findings

The main findings are presented below. First, the findings concerning students' attitudes are described, and then – those of the teaching staff.

6.1 Students' Attitudes

Table 2 displays the mean score ($1 \leq M \leq 5$) and standard deviation (SD) of the attitude (and its components) of the course graduates toward learning that combines several levels of abstraction. According to the data, the students hold a positive attitude, both cognitively and affectively.

Based on the Kolmogorov-Smirnov test for normality, a normal distribution for the scores of the cognitive component can be assumed ($D = 0.09, p > 0.05$), but not for the scores of the affective component ($D = 0.18, p < 0.05$). Therefore, Spearman's rank correlation coefficient was calculated between the two attitude components and was found to be positive, moderate, and significant ($r = 0.52, p < 0.01$).

Content analysis identified cognitive, affective, and behavioral aspects of students' attitudes toward learning that combines several levels of abstraction. From the cognitive viewpoint, the vast majority of respondents (90%) claim that such learning provides a complete picture of computer systems:

> "Educationally, it [learning] is well built. You start from the simplest thing . . . start from Boolean variables . . . then Boolean expressions, logic gates,

Table 2. Attitude component scores (mean and standard deviation)

Component	M	SD
Cognitive	3.70	0.72
Affective	4.33	0.61
In total	3.80	0.65

combinational components . . . you progress until you build a complete processor!" (interview)

"The combination is excellent. I would be happy if more courses were like this. That is, not to study only mathematics [Boolean algebra] without an apparent purpose, and only then to study all its uses [computer structure] in a separate course." (questionnaire)

About one-third of the respondents (35%) argue that learning that integrates multiple levels of abstraction is appropriate for beginning students:

"The level [of learning] was appropriate . . . I felt it wasn't something I couldn't handle." (interview)

"It [learning] was great . . . it was a real challenge . . . and while challenging it wasn't impossible." (interview)

But it imposes a heavy cognitive load resulting from the wide scope of content (32%):

"The load was too high." (questionnaire)

"There were lots and lots of content." (interview)

According to about a quarter of the respondents (24%), this type of learning promotes higher-order thinking:

"It [learning] encouraged me to think outside the box and be creative . . . to go with my line of thought." (interview)

"It [learning] has given me directions of thinking that I haven't had until today . . . I think it [combination of several levels of abstraction] develops ways of thinking." (interview)

And is relevant to industry work (21%):

"[We got] a sense of what we're going to do in the future [in the industry]." (interview)

"[What is good about the learning is] its connection to industry." (questionnaire)

Affectively, 28% of the respondents think that learning that combines several levels of abstraction is interesting:

"The connection between these two things [digital systems and computer structure] is interesting." (questionnaire)

"Very interesting . . . I liked the progress from a very internal level of the processor to an external one." (interview)

Finally, in the behavioral domain, the vast majority of respondents (83%) prefer learning that incorporates multiple levels of abstraction over learning that focuses on a few levels:

"[If it were up to me,] I would prefer to study the course in this format." (interview)

"I prefer to study the current [new] course over separate courses [a course in digital systems and a course in computer structure]." (questionnaire)

Fig. 1 displays students' attitudes toward learning that combines multiple levels of abstraction.

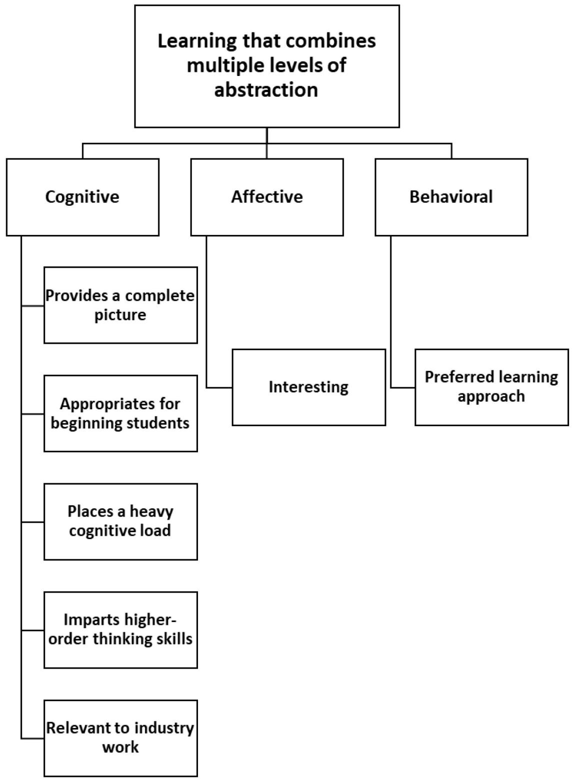

Fig. 1. Students' attitudes toward learning that combines multiple levels of abstraction.

6.2 Instructors' Attitudes

Content analysis revealed cognitive, affective and behavioral aspects of instructors' attitudes toward teaching that combines several levels of abstraction. In the cognitive domain, 63% of the respondents claim that this type of teaching promotes higher-order thinking among students:

"This course teaches how to use levels of abstraction . . . We have certain components that we have learned how to construct. Now we have a new problem. I don't want to build these basic components again, but to use the basic components to build something that is more complex, like the processor we are building in the course."

"Some of the subjects we teach [switching between levels of abstraction] are subjects that are universal . . . and I always emphasize that they can be applied not only in electronics but also in biology or chemistry."

But it is very demanding due to the broad scope of content:

"It [teaching] is very demanding. We have four hours of lectures a week, two hours of tutorials [a week] and we have a two-hour workshop five times in a semester . . . We have a lot of contents [to teach]."

"I sometimes come to the tutorial like for war. . . I say 'I have to make it'."

From the affective viewpoint, 38% of the respondents think that teaching that combines several levels of abstraction is enjoyable:

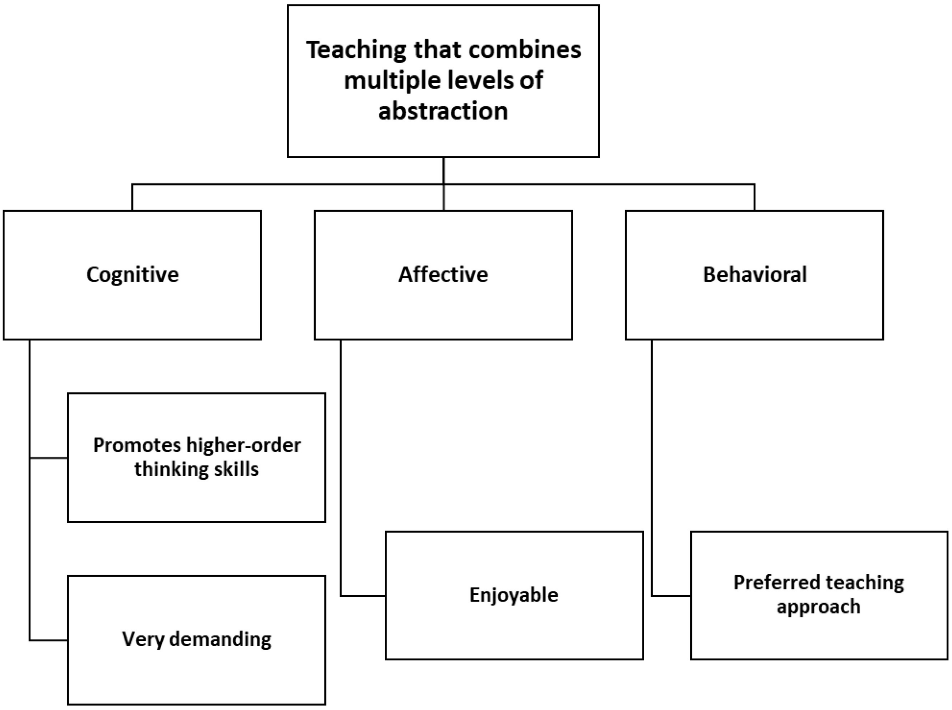

Fig. 2. Instructors' attitudes toward teaching that combines multiple levels of abstraction.

> "I really like teaching this connection [between digital systems and computer structure]."
>
> "I really enjoy teaching this stuff."

Finally, from the behavioral aspect, the vast majority of respondents (88%) prefer teaching that incorporates multiple levels of abstraction over teaching that focuses on a few levels:

> "I prefer to teach this course [over the two separate courses]."

Fig. 2 shows instructors' attitudes toward teaching that combines multiple levels of abstraction.

7. Discussion

According to the findings, the graduates of the course "Digital Systems and Computer Structure" hold a positive attitude toward learning that combines multiple levels of abstraction, both cognitively and affectively. Furthermore, the correlation between the attitude components is positive, moderate, and significant. A practical implication is that students' interest in this type of learning can be further increased by improving their rational arguments on it (and vice versa). It should be noted that although sometimes the attitude components are consistent [40], this is not always the case [41].

In the cognitive domain, students argue that learning that combines multiple levels of abstraction provides a complete picture of the computer (due to the integration of the logical, micro-architecture, and architecture levels), relevant to industry work and suitable for beginning students. Students believe that this type of learning develops higher-order thinking, and explicitly address creative thinking. Course faculty claim that teaching that combines several levels of abstraction does promote higher-order thinking among students but they refer to abstract thinking. In this context it is interesting to mention that there are researchers who point to similarities between abstract thinking and creative thinking [16], so it is possible that both students and faculty relate to a similar type of thinking.

Students' and instructors' attitudes are congruent with research results suggesting that representation that combines multiple levels of abstraction promotes learning, especially when dealing with complex systems [42, 43], such as physical [25] or biochemical systems [26]. Moreover, incorporating multiple levels of abstraction permits learners to switch between these levels and thus contributes to their cognitive and professional development [26].

Students and course faculty agree that the load characterizing learning and teaching that incorporate multiple levels of abstraction is very high. Indeed, as described in Section 3, course instruction is more extensive than is common in university courses. It seems that the heavy load constitutes a major difficulty, and it will be addressed later on.

In the affective domain, both students and instructors find interest and enjoyment in combining digital systems with computer architecture. This integration of multiple levels of abstraction demonstrates how Boolean algebra and digital systems are implemented in "real-life" scenarios, namely, computer structure. It is worth mentioning that the incorporation of such "real-world" examples often increases intrinsic motivation [44]. The behavioral component is in line with the cognitive and affective aspects described above. According to it, the vast majority of students and instructors prefer learning and teaching that integrate multiple levels of abstraction over those that focus on a few levels.

It is interesting to note that there is a partial similarity between the cognitive component of students' attitudes toward learning that integrates several levels of abstraction and the cognitive component of students' attitudes toward learning that combines several fields of knowledge, i.e., interdisciplinary learning. As mentioned above, students argue that the former provides a complete picture, relevant to industry work and develops cognitive skills, but is also characterized by a high load. Similarly, studies report that interdisciplinary learning allows the learner to look at the subject from several perspectives [45], and thus may help him/her to function well in complex work environments that characterize modern society [46]. Moreover, students claim that interdisciplinary learning foster cognitive skills [47, 48], but is also accompanied by a heavy load [49, 50]. Resemblances between the two types of learning also exist in the affective domain. As stated, students find interest in learning that combines multiple levels of abstraction. Similarly, students are interested in interdisciplinary learning [51–53]. As for faculty, instructors argue that interdisciplinary teaching is demanding and consumes a lot of resources [47], similar to the findings reported here.

A possible explanation for these resemblances lies in the fact that each type of education is based on combinations. Learning and instruction that integrate multiple levels of abstraction are often unidisciplinary but do combine different levels of abstraction. Interdisciplinary education usually deals with one level of abstraction, but does incorporate different disciplines. Moreover, as noted in Section 2, a positive correlation has been found between abstract thinking and systems thinking, with the latter usually related to interdisciplinarity [32].

In general, the authors recommend developing courses that combine several levels of abstraction. However, in view of the high load that accompanies such courses, they should be scheduled in a semester in which the load is not heavy in the first place and assigned with highly-qualified faculty.

The study had one major limitation: the number of students who took part in the study was relatively small. To overcome this limitation and to increase the findings' trustworthiness, qualitative tools were used alongside quantitative ones [36].

The main theoretical contribution of the study is in characterizing students' and instructors' attitudes toward learning and teaching that integrate multiple levels of abstraction in engineering. To the best of the authors' knowledge, such characterization was carried out here for the first time. In practice, the research may improve the teaching of courses that combine multiple levels of abstraction and promote the readiness of engineering graduates for their work in the industry. These contributions are particularly important in light of the inadequate abstract thinking skills of beginning engineering students [54].

8. Conclusions

The study shows that students hold positive attitudes toward learning that combines several levels of abstraction, both cognitively and affectively, and that the correlation between the attitude components is positive, moderate, and significant. Both students and instructors claim that education that incorporates multiple levels of abstraction promotes higher-order thinking, but is characterized by a high load.

Acknowledgements – The authors would like to express their gratitude to Prof. Ran Ginosar (Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology) for his contribution to the study.

References

1. L. Paku, M. Jensen and R. Evans, Attracting future engineers through practical and collaborative initiatives, in *Proceedings of the 29th Australasian Association for Engineering Education Conference*, p. 521, 2018.
2. A. Gero and A. Friesel, Academic motivation in beginning students of electrical engineering: A case study of Danish and Israeli universities, *Global Journal of Engineering Education*, **22**(3), pp. 204–209, 2020.
3. A. Yadav, D. Subedy, M. A. Lundeberg and C. F. Bunting, Problem-based learning: Influence in students' learning in an electrical engineering course, *Journal of Engineering Education*, **100**(2), pp. 253–280, 2011.
4. Accreditation Board for Engineering & Technology, *Criteria for Accrediting Engineering Programs*, ABET, 2020.
5. P. J. Denning, D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner and P. R. Young, Computing as a discipline, *Computer*, **22**(2), pp. 63–70, 1989.
6. T. Budd, *Introduction to Object-Oriented Programming*, Pearson Education, India, 2008.
7. O. Mishali, Y. Dubinsky and I. Maman, Towards IDE support for abstract thinking, in *Proceedings of the 2nd International Workshop on the Role of Abstraction in Software Engineering*, pp. 9–13, 2008.

8. F. Tsui, A. Gharaat, S. Duggins and E. Jung, Measuring levels of abstraction in software development, *International Conference on Software Engineering and Knowledge Engineering*, pp. 466–469, 2011.
9. Y. Patt and S. J. Patel, *Introduction to Computing Systems: From Bits and Gates to C and Beyond*, McGraw-Hill, 2008.
10. J. Kramer, Is abstraction the key to computing?, *Communications of the ACM*, **50**(4), pp. 37–42, 2007.
11. O. Hazzan and J. Kramer, Assessing abstraction skills, *Communications of the ACM*, **59**(12), pp. 43–45, 2016.
12. N. Ye and G. Salvendy, Expert-novice knowledge of computer programming at different levels of abstraction, *Ergonomics*, **39**(3), pp. 461–481, 1996.
13. University of California, Berkeley, https://www2.eecs.berkeley.edu/Courses/EECS151/
14. S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, 1980.
15. M. Wing, Research notebook: Computational thinking – What and why, *The Link Magazine*, pp. 20–23, 2011.
16. A. Gero and I. Levin, Computational thinking and constructionism: Creating difference equations in spreadsheets, *International Journal of Mathematical Education in Science and Technology*, **50**(5), pp. 779–787, 2019.
17. S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, P. Kampylis and Y. Punie, Developing computational thinking in compulsory education, *JRC Science for Policy Report*, European Commission, 2016.
18. ISTE and CSTA, *Operational Definition of Computational Thinking for K–12 Education*, 2011.
19. C. Hu, Computational thinking: What it might mean and what we might do about it, in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, pp. 223–227, ACM, 2011.
20. J. P. Monat and T. F. Gannon, Applying systems thinking to engineering and design, *Systems*, **6**(3), article 34, 2018.
21. 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.
22. G. Spöttl, Development of "Industry 4.0" – are skilled workers and semi-engineers the losers? in *7th World Engineering Education Forum*, 2017, pp. 851–856.
23. J. Piaget, *The Origin of Intelligence in Children*, International University Press, 1952.
24. O. Hazzan, Reflections on teaching abstraction and other soft ideas, *ACM SIGCSE Bulletin*, **40**(2), pp. 40–43, 2008.
25. P. C. H. Cheng, Electrifying diagrams for learning: Principles for complex representational systems, *Cognitive Science*, **26**(6), pp. 685–736, 2002.
26. E. G. Offerdahl, J. B. Arneson and N. Byrne, Lighten the load: Scaffolding visual literacy in biochemistry and molecular biology, *CBE – Life Sciences Education*, **16**(1), es1, 2017.
27. J. Sanguinetti, Abstraction and standardization in hardware design, *IEEE Design and Test of Computers*, **29**(2), pp. 8–13, 2012.
28. A. Shekh-Abed, O. Hazzan and A. Gero, Promoting systems thinking and abstract thinking in high-school electronics students: Integration of dedicated tasks into project-based learning, *International Journal of Engineering Education*, **37**(4), pp. 1080–1089, 2021.
29. 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.
30. 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.
31. H. Koppelman and B. van Dijk, Teaching abstraction in introductory courses, in *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, pp. 174–178, ACM, 2010.
32. 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*, **46**(5), pp. 735–749, 2021.
33. Ö. Korkmaz, R. Çakir and M. Y. Özden, A validity and reliability study of the computational thinking scales (CTS), *Computers in Human Behavior*, **72**, pp. 558–569, 2017.
34. W. J. Dally and R. C. Harting, *Digital Design: A Systems Approach*, Cambridge University Pres, 2012.
35. J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Elsevier, 2011.
36. R. B. Johnson and A. J. Onwuegbuzie, Mixed methods research: A research paradigm whose time has come, *Educational Researcher*, **33**(7), pp. 14–26, 2014.
37. H. F. Hsieh and S. E. Shannon, Three approaches to qualitative content analysis, *Qualitative Health Research*, **15**(9), pp. 1277–1288, 2005.
38. M. J. Rosenberg and C. I. Hovland, Cognitive, affective and behavioral components of attitudes, in M. J. Rosenberg, C. I. Hovland, W. J. McGuire, R. P. Abelson and J. W. Brehm (eds), *Attitude Organization and Change: An Analysis of Consistency Among Attitude Components*, Yale University, CT, pp. 1–14, 1960.
39. S. Fishbein, *Evaluation of Interdisciplinary Learning in Science and Technology*, Technion – Israel Institute of Technology, Haifa, Israel, 2003.
40. E. Ç. Altuntaş and S. L. Turan, Awareness of secondary school students about renewable energy sources, *Renewable Energy*, **116**(A), pp. 741–748, 2018.
41. L. Festinger, *A Theory of Cognitive Dissonance*, Stanford University Press, 1957.
42. J. Airey and C. Linder, A disciplinary discourse perspective on university science learning: Achieving fluency in a critical constellation of modes, *Journal of Research in Science Teaching*, **46**(1), pp. 27–49, 2009.
43. S. Ainsworth, DeFT: A conceptual framework for considering learning with multiple representations, *Learning and Instruction*, **16**(3), pp. 183–198, 2006.
44. A. Gero, Y. Stav and N. Yamin, Increasing motivation of engineering students: Combining "real-world" examples in a basic electric circuits course, *International Journal of Engineering Education*, **32**(6), pp. 2460–2469, 2016.
45. A. Gero, Interdisciplinary program on aviation weapon systems as a mean of improving high school students' attitudes towards physics and engineering, *International Journal of Engineering Education*, **29**(4), pp. 1047–1054, 2013.
46. L. Ivanitskaya, D. Clark, G. Montgomeryand and R. Primeau, Interdisciplinary learning: Process and outcomes, *Innovative Higher Education*, **27**, pp. 95–111, 2002.
47. A. Gero, Students' attitudes towards interdisciplinary education: A course on interdisciplinary aspects of science and engineering education, *European Journal of Engineering Education*, **42**(3), pp. 260–270, 2017.
48. M. Borrego and L. K. Newswander, Definitions of interdisciplinary research: Toward graduate-level interdisciplinary learning outcomes, *The Review of Higher Education*, **34**(1), pp. 61–84, 2010.

49. 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.
50. Y. Zhang, Y. Zhao, L. He and Q. Bai, Interdisciplinary teaching design and practice based on STEM, *Modern Distance Education Research*, **1**, pp. 10–11, 2017.
51. 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.
52. J. Straub, R. Marsh and D. Whalen, The impact of an interdisciplinary space program on computer science student learning, *Journal of Computers in Mathematics and Science Teaching*, **34**(1), pp. 97–125, 2015.
53. A. Gero, H. Essami, O. Danino and L. Kornblum, Students' attitudes toward interdisciplinary learning: A high-school course on solar cells, *International Journal of Engineering Education*, **38**(4), pp. 1130–1140, 2022.
54. P. Rivera-Reyes and L. C. Pérez, Abstraction and problem solving in an undergraduate electrical engineering circuits course. in *Proceedings of Frontiers in Education Conference*, pp. 1–7, IEEE, 2016.

Appendix A – Open-Ended Questionnaire

Below is a sample of the open-ended questions mentioned in Section 5.3:

- Describe the most interesting lesson in the course?
- What do you think about the combination of digital systems and computer structure? explain.
- If it were up to you, would you rather study the current course or two separate courses (a course in digital systems and a course in computer structure)? explain.

Appendix B – Interview

Below is a sample of the interview questions mentioned in Section 5.3:

- What do you think about the course?
- Would you change anything in the course? explain.
- What do you think about the combination of digital systems and computer structure? explain.

Mohammed Ali Hadish holds a BSc in Technology and Science Education from the Technion – Israel Institute of Technology. Currently, Mr. Hadish is a graduate student in the Department of Education in Technology and Science at the Technion.

Shahar Kvatinsky is an Associate Professor at the Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology. Shahar received the BSc degree in Computer Engineering and Applied Physics and an MBA degree in 2009 and 2010, respectively, both from the Hebrew University of Jerusalem, and the Ph.D. degree in Electrical Engineering from the Technion – Israel Institute of Technology in 2014. From 2006 to 2009, he worked as a circuit designer at Intel. From 2014 and 2015, he was a post-doctoral research fellow at Stanford University. Kvatinsky is a member of the Israel Young Academy. He is the head of the Architecture and Circuits Research Center at the Technion, chair of the IEEE Circuits and Systems in Israel, and an editor of Microelectronics Journal and Array. Kvatinsky has been the recipient of numerous awards: 2021 Norman Seiden Prize for Academic Excellence, 2020 MDPI Electronics Young Investigator Award, 2019 Wolf Foundation's Krill Prize for Excellence in Scientific Research, 2015 IEEE Guillemin-Cauer Best Paper Award, 2015 Best Paper of Computer Architecture Letters, Viterbi Fellowship, Jacobs Fellowship, ERC starting grant, the 2017 Pazy Memorial Award, the 2014, 2017 and 2021 Hershel Rich Technion Innovation Awards, 2013 Sanford Kaplan Prize for Creative Management in High Tech, 2010 Benin prize, and seven Technion excellence teaching awards. His current research is focused on circuits and architectures with emerging memory technologies and design of energy efficient architectures.

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. 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 16 times. He received the Israeli Air-Force Commander's Award for Excellence (2006), the Yanai Prize for Excellence in Academic Education (2016) and the Technion's Award for Continuing Excellence in Teaching (2022). Dr. Gero is an Associate Editor of IEEE Transactions on Education.