Undergraduate Students' Attitudes Toward an Engineering Course that Integrates Several Levels of Abstraction

Aharon Gero, Mohammed Ali Hadish, and Shahar Kvatinsky

Technion – Israel Institute of Technology, Haifa, Israel
gero@technion.ac.il,hadish16@campus.technion.ac.il,
shahar@ee.technion.ac.il

Abstract. Abstract thinking is considered a higher-order thinking that allows one to solve problems while switching between levels of abstraction, i.e., moving between multiple levels of complexity. The Faculty of Electrical and Computer Engineering (Technion – Israel Institute of Technology) combined two undergraduate courses, focusing on different levels of abstraction, into a single course "Digital Systems and Computer Structure" integrating multiple levels of abstraction, i.e., logical, micro-architecture and architecture levels. The present study aimed to characterize students' attitudes toward the course. This mixed-method study involved 103 undergraduate students. According to the findings, students hold positive attitudes toward a course that combines multiple levels of abstraction. Students claim that the course is interesting, provides a complete picture of computer systems, imparts higher-order thinking skills, but at the same time imposes a heavy load.

Keywords: Abstract Thinking, Students' Attitudes, Digital Systems, Computer Structure.

1 Introduction

One type of thinking relevant to engineering is abstract thinking, namely, focusing on meaningful information for the current phase and temporarily ignoring the unnecessary details [1-2]. Abstract thinking is considered a higher-order thinking that allows one to solve problems while switching between levels of abstraction, i.e., moving between multiple levels of complexity [3].

In view of the above, the Faculty of Electrical and Computer Engineering (Technion – Israel Institute of Technology) combined two undergraduate courses, dealing with different levels of abstraction, into a single course "Digital Systems and Computer Structure" integrating multiple levels of abstraction. This unique course is mandatory for beginning electrical and computer engineering (hereinafter: ECE) students and computer science (hereinafter: CS) students. The present study aimed to characterize students' attitudes toward the course.

The article begins with a concise review of abstract thinking. Next, the course "Digital Systems and Computer Structure" is described. Then, the research goal is formulated and the methodology is presented. Finally, the main findings are discussed.

2 Abstract Thinking

Abstract thinking, as an important element of computational thinking [4-5], is relevant in many fields, and currently holds a major role in engineering [6]. This is due to the increased complexity of engineering systems [7] and the Industry 4.0 concept, based, inter alia, on big data [8]. Therefore, compared to the past, engineers must deal with growing intricacy, especially in the hardware and software industry [9].

As mentioned earlier, a key concept in abstract thinking is a level of abstraction. It is possible to characterize an arbitrary number of levels, ranging from the highest level, where the problem is observed from a general point of view, to the lowest level at which close attention to detail is needed [10].

For instance, 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, focusing on the highest level of implementation of the features defined at the previous level. The next level, logic gates, deals with the logical properties of the components defined at the micro-architecture level. Lower levels address the electronic devices that construct the logic gates and their physical structure [11].

It is widely agreed that the major cognitive skills of the so-called abstract thinker are recognizing the level of abstraction appropriate for a particular phase and maneuvering between levels of abstraction [12-13].

3 "Digital Systems and Computer Structure" Course

As noted earlier, the Faculty of Electrical and Computer Engineering (Technion – Israel Institute of Technology) combined the course "Digital Systems", which dealt with Boolean algebra and the analysis and design of combinational and sequential systems, and the course "Logic Design and Introduction to Computers", focusing on advanced logic design and computer structure, into a single undergraduate-level course entitled "Digital Systems and Computer Structure". The latter covers three levels of abstraction, i.e., logical, micro-architecture and architecture levels, switches between them, and therefore provides a more complete picture of the computer.

The course "Digital Systems and Computer Structure" is compulsory for ECE (second semester) and CS (third semester) students. The aim of the course is to provide students with analysis and design skills at the three levels of abstraction mentioned above. A prerequisite for the course is the course "Introduction to Computer Science", which deals with the algorithmic approach to problem solving and programming in C language.

The course (frontal instruction) lasts 13 weeks, and consists of four hours of lectures and two hours of tutorials every week. The teaching staff is comprised of lecturers (PhD in ECE) and assistants (senior ECE students and graduate students). The contents of the course overlap approximately with the contents of the preceding courses, and include: Boolean algebra, the digital model, combinational systems, sequential systems, state machines, von Neumann machine, hardware description language, addressing modes, branches, stack, routines, traps, pipelined processor, control and data hazards [14].

The assessment is based on computer exercises (simulations in VERILOG), a mid-semester test and a final examination.

4 Research Goal and Methodology

4.1 Goal

The aim of the study was to characterize students' attitudes toward the course "Digital Systems and Computer Structure".

4.2 Participants

One hundred and three undergraduate students (53 ECE students; 50 CS students), who took the course "Digital Systems and Computer Structure", participated in the study. The students have not previously been exposed to learning that combines multiple levels of abstraction.

4.3 Methodology

At the end of the course, students filled out two anonymous questionnaires, a closed-ended and an open-ended. In addition, nine semi-structured interviews were conducted. All instruments were aimed to characterize students' attitudes toward the course.

The quantitative data were analyzed by statistical means, and a possible difference between ECE and CS students was examined. The qualitative data underwent a directed content analysis based on the tri-component attitude model.

4.4 Tools

The closed-ended questionnaire was a five-level Likert-like scale, extending from "strongly disagree" to "strongly agree". The scale was based on an instrument developed in [15]. This self-reporting questionnaire consisted of 17 statements expressing attitudes toward an engineering course that combines multiple levels of abstraction. Some of the statements represented positive attitudes and others – negative ones. The statements were validated by two engineering education experts. Cronbach's alpha (α = 0.78) pointed to acceptable internal consistency. Some statements are given in Table 1.

In the open-ended questionnaires and interviews, students were asked, among other things, about their position toward the course and their preference, i.e., courses that incorporate multiple levels of abstraction or courses that focus on a few levels.

Table 1. Closed-ended questionnaire – sample statements.

Polarity	Statement
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.

5 Findings

5.1 ECE and CS Students' Attitudes

Table 2 presents the mean score (5≥M≥1) and standard deviation (SD) of students' attitudes toward the course (closed-ended questionnaire). According to *t*-test (equal variances), there is no significant difference between ECE and CS students (t_{101}=1.41, p>0.05). Both students hold positive attitudes toward an engineering course that integrates several levels of abstraction.

Table 2. Closed-ended questionnaire – descriptive statistics.

Group	M	SD
ECE	3.71	0.70
CS	3.89	0.58

5.2 Attitude Components

Content analysis of the open-ended questionnaires and interviews reveals various aspects in respondents' attitudes toward the course. Since there is no significant difference between ECE and CS students' attitudes (Section 5.1), there is no distinction between the two groups below.

Cognitive Domain. Cognitively, the vast majority of respondents (90%) argue that the course provides a complete picture of computer systems:

> *Educationally, it [the course] is well built. You start from the simplest thing... start from Boolean variables... then Boolean expressions, logic gates, combinational components… you progress until you build a complete processor!* (interview)

About one-third of the respondents (35%) claim that the course is appropriate for beginning students:

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

But it is accompanied by a heavy cognitive load, stemming from the broad scope of content (32%):

The load was too high… there were lots and lots of content. (questionnaire)

According to about a quarter of the respondents (24%), the course imparts higher-order thinking skills, due to the combination of several levels of abstraction:

It [the course] 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)

Affective Domain. In the affective domain, 28% of the respondents think that the course is interesting because it maneuvers between several levels of abstraction:

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

Behavioral Domain. Finally, from the behavioral perspective, the vast majority of respondents (83%) prefer courses that incorporate multiple levels of abstraction over courses that focus on a few levels:

I prefer to study the current [new] course over the two separate [preceding] courses. (questionnaire)

6 Discussion and Conclusions

According to the results, students hold positive attitudes toward an engineering course that combines multiple levels of abstraction, with no significant difference between ECE and CS students. This lack of difference may be due to the fact that the respondents are beginning students. Thus, the specific program (ECE vs. CS) has not yet affected, if at all, students' attitudes.

Students argue that the course imparts higher-order thinking skills, due to the combination of several levels of abstraction. This claim is in line with research findings suggesting that representation that combines multiple levels of abstraction promotes learning, especially when dealing with complex systems, such as physical [16] or biochemical systems [17]. Moreover, incorporating multiple levels of abstraction allows learners to maneuver between levels of abstraction and thus contributes to their cognitive development [16].

Students claim that the load characterizing a course that incorporates multiple levels of abstraction is very high. Indeed, as described in Section 3, course instruction is much more extensive than is common in university courses.

Affectively, students find interest in combining digital systems with computer structure. The behavioral aspect in their attitude is congruent with the cognitive and affective components described above. According to it, the vast majority of students prefer courses that incorporate multiple levels of abstraction over courses that focus on a few levels.

The study faced a limitation: the number of students who took part in the research was relatively small. To diminish the impact of this limitation, both quantitative and qualitative tools were used.

The study's main contribution lies in the first analysis (to the best of our knowledge) of students' attitudes toward an engineering course that integrates multiple levels of abstraction. This contribution is important in view of the inadequate abstract thinking skills of beginning engineering students [18]. The authors recommend developing such courses. However, they should be scheduled in a semester characterized by a relatively moderate load and assigned with highly-qualified faculty.

References

1. Denning, P.J., Comer, D.E., Gries, D., Mulder, M.C., Tucker, A., Turner, A.J., Young, P.R.: Computing as a discipline. Computer 22(2), 63–70 (1989).
2. Gero, A., Shekh-Abed, A., Hazzan, O.: Correlation between systems thinking and abstract thinking among high school students majoring in electronics. In: The Impact of the 4th Industrial Revolution on Engineering Education, pp. 541–548. Springer, Cham (2020).
3. Mishali, O., Dubinsky, Y., Maman, I.: 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).
4. Hu, C.: 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).
5. Gero, A., Levin, I.: Computational thinking and constructionism: Creating difference equations in spreadsheets. International Journal of Mathematical Education in Science and Technology 50(5), 779–787 (2019).
6. Kramer, J.: Is abstraction the key to computing? Communications of the ACM 50(4), 37–42 (2007).
7. Monat, J.P., Gannon, T.F.: Applying systems thinking to engineering and design. Systems 6(3), 34 (2018).
8. Jazdi, N.: Cyber physical systems in the context of Industry 4.0. In: IEEE International Conference on Automation, Quality and Testing, pp. 1–4. IEEE (2014).
9. Spöttl, G.: Development of "Industry 4.0" – are skilled workers and semi-engineers the losers? In: 7th World Engineering Education Forum, pp. 851–856. (2017).
10. Hazzan, O.: Reflections on teaching abstraction and other soft ideas. ACM SIGCSE Bulletin 40(2), 40–43 (2008).

11. Sanguinetti, J.: Abstraction and standardization in hardware design. IEEE Design and Test of Computers 29(2), 8–13 (2012).
12. Gero, A., Shekh-Abed, A., Hazzan, O.: Interrelations between systems thinking and abstract thinking: The case of high-school electronics students. European Journal of Engineering Education 46(5), 735–749 (2021).
13. Shekh-Abed, A., Hazzan, O., Gero, A.: 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), 1080–1089 (2021).
14. Digital Systems and Computer Structure, https://students.technion.ac.il/local/technionsearch/course/44252/202102?lang=en, last accessed 2022/5/1.
15. Fishbein, S.: Evaluation of interdisciplinary learning in science and technology. Technion - Israel Institute of Technology, Israel (2003).
16. Cheng, P.C.H.: Electrifying diagrams for learning: Principles for complex representational systems. Cognitive Science 26(6), 685–736 (2002).
17. Offerdahl, E.G., Arneson, J.B., Byrne, N.: Lighten the load: Scaffolding visual literacy in biochemistry and molecular biology. CBE–Life Sciences Education 16(1), 1 (2017).
18. Rivera-Reyes, P. Pérez, L.C.: Abstraction and problem solving in an undergraduate electrical engineering circuits course. In: Proceedings of Frontiers in Education Conference, pp. 1–7. IEEE (2016).