Cross Disciplinary Perceptions of the Computational Thinking among Freshmen Engineering Students

Rami J. Haddad and Youakim Kalaani
Electrical Engineering Department,
Allen E. Paulson College of Engineering and Information Technology,
Georgia Southern University

Abstract

In this paper, we analyzed the perception of Computational Thinking among engineering students from three different engineering disciplines (Electrical, Mechanical, and Civil) and correlated their performance with their discipline. The goal of this analysis is to determine whether structuring discipline-specific Computational Thinking courses is more beneficial than the current setting which allows for student multidisciplinary interaction. This analysis was quantitatively verified by assessing the students' performance in over 40 different sections of Computing for Engineers course taught over two years period (2012-2014). Our sample consisted of 861 students (142 Civil, 484 Mechanical, and 235 Electrical). Students' performance was assessed using quizzes, assignments, lab projects, and exams. We statistically analyzed students' performance and presented our findings which are thought to help the structuring of Computational Thinking courses in multidisciplinary engineering programs.

Keywords

Computational thinking, computing in engineering, computing education.

Introduction

Computational Thinking (CT) as a concept was the driving force for inventing computers[1]. Historically, the Computational Thinking was referred to as algorithmic thinking[2]. The term Computational Thinking was first coined in 1996 by Seymour Papert[3]. Ten years later, Jeannette Wing re-emphasized the importance of Computational Thinking as a fundamental skill and went on to define it as a thought processes carried out by an information-processing agent[4]. Information-processing agents or computers have very powerful processing capabilities. However, their structured “Computational” approach to processing information can limit their ability to solve problems. To solve a problem using computers, the problem’s abstract physical framework has to be mapped into a computational framework using a process called modeling. Modeling involves formulating the problem, defining its inputs and outputs, dividing it into basic parts to be individually solved to generate the required solutions. Computational Thinking is comprised of four main skills namely, abstraction, decomposition, recursion, and algorithm design[5,6]. Abstraction is the skill that identifies the underlying laws and principles that governs the physical behavior of a model. Decomposition is the skill that involves breaking the problem into basic parts or components. Recursion is the skill that utilizes a repetitive solution of a simple instance of the problem to solve the more complex problem. Finally, algorithm design is the process of combining the solutions of all the decomposed parts of the problem in logical order. Therefore, Computational Thinking skills are vital for engineering students to solve complex problems using computers[4]. Current technological advancement in computation have simplified the process of modeling however they have actually contributed to diminishing Computational Thinking skills in the post-millennials[7].

Recent studies have indicated that engineering students who decide to change their majors usually do that at the freshman level[8]. One main reason for students to change majors is the low level of success in STEM courses[9,10]. This is mainly associated with their ability to master Computational Thinking early on in their academic careers. Therefore, understanding the students' cognitive learning styles is vital to help improve retention, progression, and graduation rates. Based on our initial observations, we have noticed a difference in how engineering students from different disciplines perceive Computational Thinking. This implies the existence of different cognitive profiles among engineering students from various disciplines. Our hypothesis indicated that perception of Computational Thinking instructed to engineering students from various backgrounds will differ and depend on the instructor background. A recent research study indicated disciplinary variation in student writing skills[11]. This study is one of only few studies that address the disciplinary variation in student skills. However, the disciplinary variation in student perception of computation thinking has been ignored, mainly because the majority of the Computational Thinking based courses are targeted to serve a specific major or discipline. In our case, the Computing for Engineers course we offer has a unique structure in its diversity of students' disciplines which facilitated the study of cross disciplinary perceptions of the Computational Thinking concepts.

Our Model of Integrating Computational Thinking

At Georgia Southern University, Computational Thinking is formally introduced to students in ENGR 1731 Computing for Engineers course at the freshman level. It provides students with the foundations of Computational Thinking coupled with an introduction to the design and analysis of algorithms to solve engineering problems. It is also intended to engage engineering students in a multi-disciplinary environment. Topics discussed in this course include problem abstraction, problem decomposition, fundamental programming concepts, and the practical and theoretical limitations of computation. MATLAB is used as the programming language and was chosen for its simple syntax and relevance to all Engineering disciplines. This 3-credit, problem-based course is exclusively taught by Electrical Engineering faculty. The contact hours are divided equally between lecturing and hands-on application using a problem-based model. A supporting lecture notes book[12] is used which is especially tailored to go hand-in-hand with the course syllabus. MATLAB Programming for Engineers is the textbook used by the students as additional reference[13]. Even-though the lecture notes are customized and offer basic engineering concepts, the instruction and the problems introduced do not seem to equally appeal to students having different engineering backgrounds.

Assessment of Computation Thinking Perception

To test the proposed hypothesis, a quantitative analysis of the variation in students' perception of Computational Thinking across different engineering disciplines was conducted in 40 sections of the Computing for Engineers course offered from Fall 2012 to Spring 2014. Our sample consisted of 861 students (142 CE, 484 ME, and 235 EE). Less than 2% of the students from these engineering disciplines claimed to have prior experience in any computer programming which was mostly found to be very basic knowledge. The student performance was assessed using a set of quizzes, assignments, weekly lab reports, and exams. The students' final grade in the course was used to assess their ability to perceive Computational Thinking. Figure 1 demonstrates the distribution of the students' final grade in those three engineering disciplines.

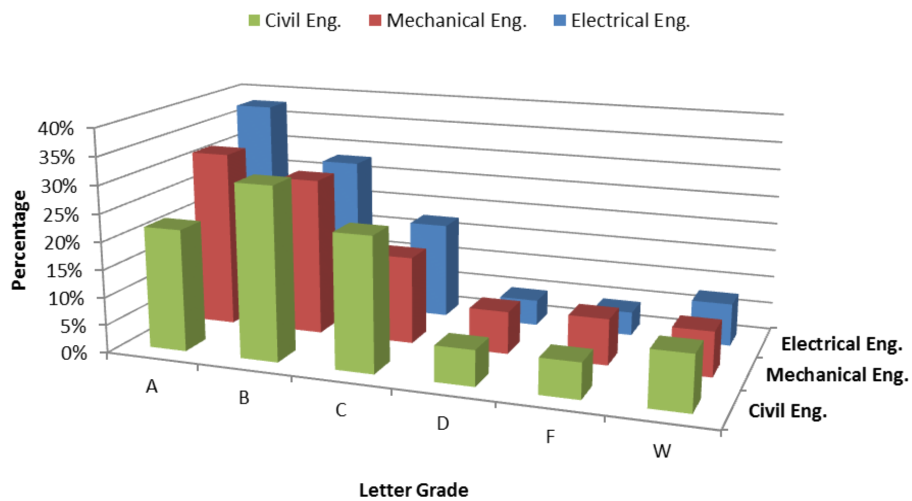

Figure 1- Distribution of Students' Grades in the Computing for Engineering Course

From Figure 1, it is evident that EE students had the lowest D-grade, F-grade, and withdrawal (DFW) rates compared to ME and CE students. In addition, EE had the highest rate for A-grade followed by ME and CE, respectively. The difference in student performance is contributed to how students from different engineering disciplines perceive Computational Thinking when it is instructed by faculty with a specific engineering background. Figure 2 shows the normal distribution fit for the students' grades categorized by disciplines. As shown, there is a difference in the overall grade point average among engineering disciplines. Therefore, we can further claim that there is a strong correlation between Computational Thinking pedagogy and the instructor discipline that ultimately affect the students' perception of Computational Thinking knowledge. Due to the randomization of our student sample, each section had almost similar distribution of engineering disciplines. Therefore, the effect due to the differences among the instructors teaching the course averaged out in this analysis. However, a recent study found that the variation across Computational Thinking instructors can have an impact on the students' long-term academic success[14].

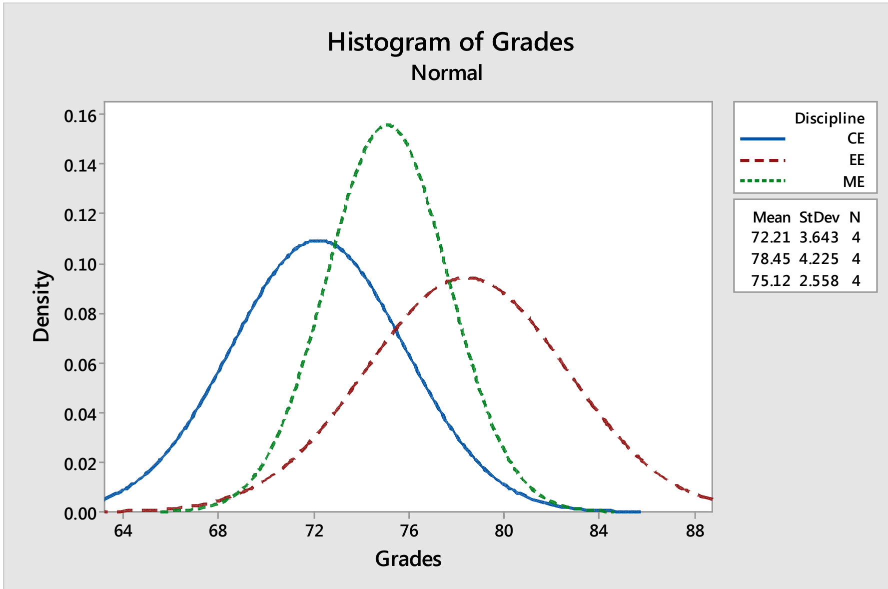

Figure 2- Normal Distribution Fit of Students' Grades Categorized by Discipline

To statistically verify our hypothesis, a thorough statistical analysis using the Minitab statistics software[15] was conducted. The null hypothesis indicates that there are no statistical significant differences in the students' grades across disciplines. To test this hypothesis, the General Linear Model was used to analyze the data with probability criterion for 5% (p=0.05) significance level. The null hypothesis is rejected if the analysis generates a p-value less than the 0.05 significance level. This indicates that the perception of Computational Thinking varies depending on the engineering disciplines. The response variable is the students' average grades categorized by discipline obtained in four academic semesters. As illustrated in Figure 3, there are two main factors to consider.

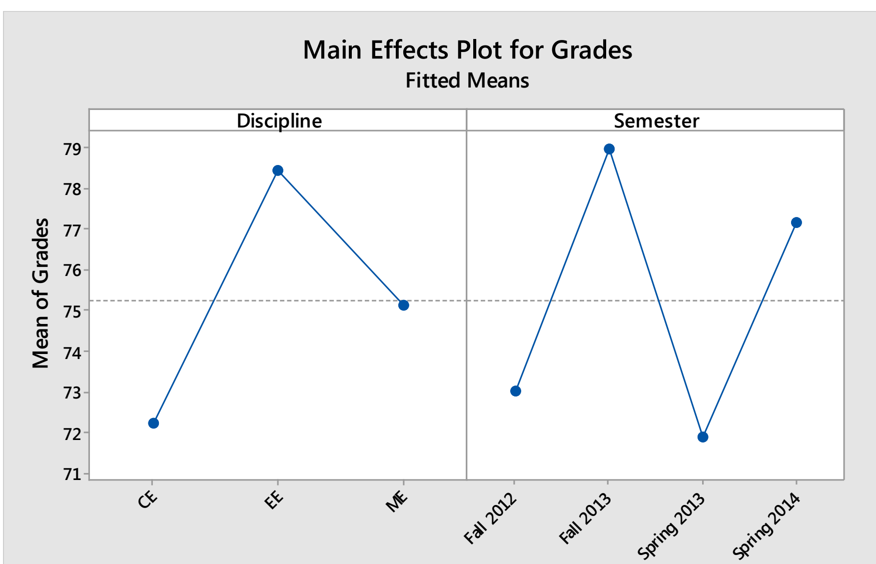

Figure 3 – Main Effect Plot - Treatment Effect (Discipline) and Nuisance Effect (Semester)

The first factor is the treatment effect modeled by the grade average of EE, ME, and CE students' grade average. The three-level treatment used is the effect of Computational Thinking perception among EE, ME, and CE students. The second factor is the semester effect which is modeled as a nuisance or blocking factor to extract the variability due to students and instructors.

The statistical analysis presented in Figure 4 generated a p-value less than **0.002** which is twenty five times less than the **0.05** criterion for significance. Therefore, the null hypothesis is rejected with a confidence level of **99.8%** confirming the existence of a statistically significant difference in grade average among students' from different engineering disciplines. This validates our hypothesis that there is a difference in how Computational Thinking is perceived across different engineering disciplines.

Statistical Analysis Model (General Linear Model: Grades versus Discipline, Semester)

```
Factor      Type    Levels  Values
Discipline  fixed        3  CE, EE, ME
Semester    random       4  Fall 2012, Fall 2013, Spring 2013, Spring 2014


Analysis of Variance for Grades, using Adjusted SS for Tests

Source      DF   Seq SS   Adj SS  Adj MS      F      P
Discipline   2   77.881   77.881  38.941  21.91  0.002
Semester     3  102.341  102.341  34.114  19.20  0.002
Error        6   10.663   10.663   1.777
Total       11  190.885


S = 1.33308   R-Sq = 94.41%   R-Sq(adj) = 89.76%

Grouping Information Using Tukey Method and 95.0% Confidence

Discipline  N   Mean  Grouping
EE          4  78.45  A
ME          4  75.12    B
CE          4  72.21      C

Means that do not share a letter are significantly different.
```

Figure 4- Outcome of the Two-way ANOVA with Blocking Statistical Analysis

To further investigate this conclusion, Fisher comparisons were conducted with a confidence level of **95%** as illustrated in Figure 5. The outcome of the Fisher's comparisons also supported our conclusion that the performance of engineering students from different disciplines is statistically different. Based on these pairwise comparisons, EE students in general were shown to perform better than ME and CE students, followed by ME students who performed better than CE students. These results could be attributed to the correlation between the instructor background and the students' cognitive learning styles in perceiving Computational Thinking knowledge.

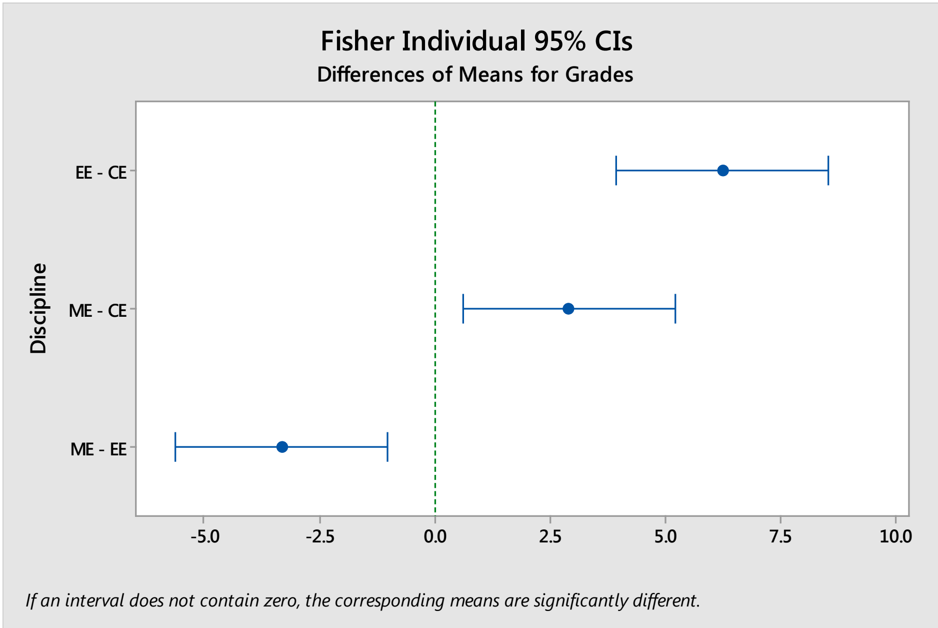

Figure 5- Fisher 95.0% Simultaneous Confidence Intervals

To test the model's goodness of fit, the probability plot of the students' grades based on the discipline was generated as shown in Figure 6. The data points in the CE, ME, and EE figures relatively follow the straight line generating a p-value over 0.05 and a low adjusted Anderson-Darling statistic (AD). This supported our conclusions that all students' grades fit a normal distribution. However, CE students' grades generated the highest p-value of 0.86 and the lowest AD statistic indicating that CE students' grades fit a normal distribution better than ME and EE. The order of the students' grades goodness of fit correlated well with the grade means values illustrating not only the impact on the grade means but also on the students' grade distributions.

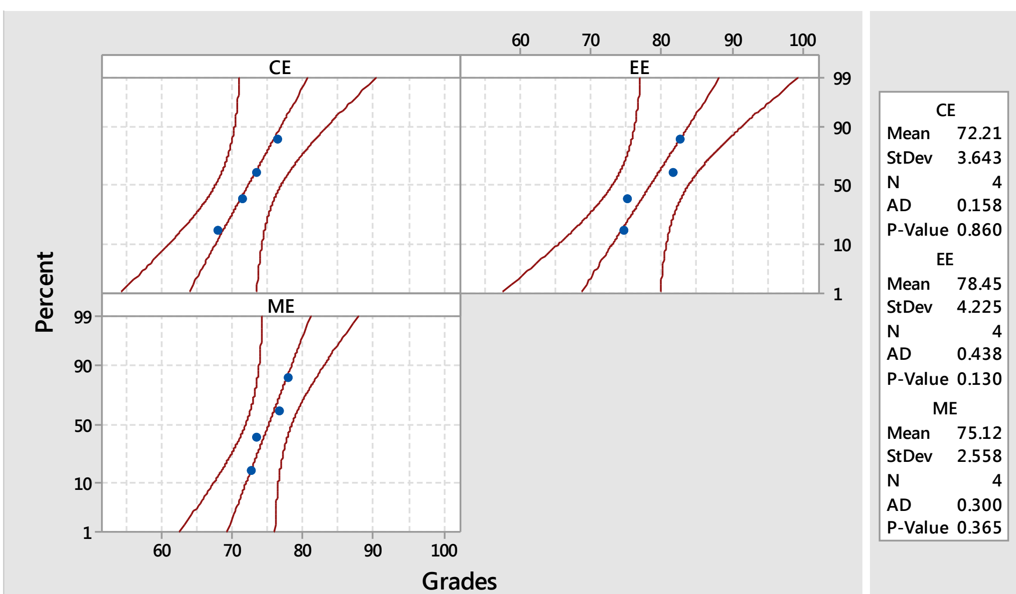

Figure 6- Probability Plot of Grades for EE, ME, and CE Students

Conclusions

Computational Thinking is one of the most essential skills that engineering students should have in order to succeed in their academic and professional careers. However the perception of Computational Thinking can differ among students depending on their discipline, which makes it a challenge to effectively teach Computational Thinking. This paper presented an intensive study of Computational Thinking perception among students in a multidisciplinary engineering course offered at the freshman level. In this study, the students' grade point average was used to assess the relationship between Computational Thinking perception and the students' engineering discipline. It was determined that there is a statistical significant difference among students grades based on their engineering discipline. This conclusion was inferred by statistical analysis with 95% confidence level. To improve the teaching effectiveness, it is recommended that discipline-specific Computational Thinking instruction to be implemented. This would improve the students' perception of Computational Thinking, improve their performance in other engineering courses, and ultimately have a positive impact on the students' retention, progression, and graduation rates.

References

1 Michael S. Mahoney "The history of computing in the history of technology" in Annals for the History of Computing Vol. 10, No. 2, 1988, pp 113-125

2 Denning, P. J. (2009, June). "The profession of IT - beyond Computational Thinking," Communications of the ACM, 52 (6), 28-29.

3 S. Papert, (1996). "An Exploration in the Space of Mathematics Educations," International Journal of Computers for Mathematical Learning, vol. 1, no. 1, pp. 95-123.

4 J. M. Wing, (2006). "Computational Thinking," Communications of the ACM, vol. 49, no.3, pp. 33-35.

5 Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards Computational Thinking for science majors. In Sigcse '09: Proceedings of the 40th ACM technical symposium on computer science education (pp. 183–187). New York, NY, USA: ACM.

6 Eisenberg, M. (2008, July). Recursion - or, better Computational Thinking through laughter. International Journal of Computers for Mathematical Learning, 171-174.

7 Dede, C., P. Mishra, et al. (2013). Working Group 6: Advancing Computational Thinking in 21st century learning, EDUsummIT.

8 Bernold, L. E., Spurlin, J. E. and Anson, C. M. (2007), "Understanding Our Students: A Longitudinal-Study of Success and Failure in Engineering With Implications for Increased Retention", Journal of Engineering Education, 96: 263–274.

9 National Center for Education Statistics (NCES) (2013). "STEM Attrition: College Students' Paths Into and Out of STEM Fields", Statistical Analysis Report.

10 R.M. Marra, K.A. Rodgers, D. Shen & B. Bogue,"Leaving engineering: A multi-year single institution study," J. Eng. Education, 101(1), pp. 6-27, Jan. 2012.

11 Hardy, J. A., & Römer, U. (2013). "Revealing disciplinary variation in student writing: a multidimensional analysis of the Michigan Corpus of Upper-level Student Papers" (MICUSP). Corpora, 8(2), 183-207.

12 Bernadin, S., Haddad, R. (2014). "Computing for Engineers: Course Notes", 2nd Ed., 164 Pages, Kendall Hunt, ISBN-9781465259455.

13 Chapman, S. (2008). "MATLAB Programming for Engineers", 4th Ed., Cengage Learning, ISBN-9780495244493.

14 Haddad, R. J., Kalaani, Y. (2015). "Can Computational Thinking Predict Academic Performance?", Integrated STEM Education Conference (ISEC), 2015 IEEE, vol., no., pp.1-5, 7, March 2015.

15 Minitab 17 Statistical Software (2010). [Computer software]. Minitab, Inc. http://www.minitab.com/en-us/

Rami J. Haddad

Rami J. Haddad is an Assistant Professor in the Department of Electrical Engineering at Georgia Southern University. He received his B.S. degree in Electronics and Telecommunication Engineering from the Applied Sciences University, Amman, Jordan, his M.S. degree in Electrical and Computer Engineering from the University of Minnesota, Duluth, MN, and his Ph.D. degree from the University of Akron, Akron, OH. His research focuses on various aspects of optical fiber communication/networks, broadband networks, smart grid applications, multimedia bandwidth forecasting and engineering education.

Youakim Kalaani

Youakim Kalaani is an Associate Professor of Electrical Engineering in the Department of Electrical Engineering at Georgia Southern University. Dr. Kalaani received his B.S. degree in Electrical Engineering from Cleveland State University (CSU). He graduated from CSU with M.S. and Doctoral degrees in Electrical Engineering with concentration in power systems. Dr. Kalaani is a licensed professional engineer (PE) and an ABET Program Evaluator (PA). He is a member of IEEE and has research interests in distributed power generations, optimization, and engineering education.