ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #38405

Computational Thinking in the Formation of Engineers: Year 2

Noemi V Mendoza Diaz (Assistant Professor)

Dr. Mendoza is a faculty member of Technology Management in the College of Education-Engineering at Texas A&M University. She has worked as electrical engineering professor in Mexico. She recently obtained funds from NSF to investigate enculturation to engineering and computational thinking in engineering students. She is the co-advisor of the Society for Hispanic Professional Engineers at TAMU and is interested in computing engineering education and Latinx engineering entrepreneurship.

Russ Meier

Dr. Russ Meier teaches computer architecture at Milwaukee School of Engineering. His funded research explores how first year students develop computational thinking. He received the Iowa State University Teaching Excellence Award, the Iowa State University Warren B. Boast Award for Undergraduate Teaching Excellence, and the MSOE Oscar Werwath Distinguished Teacher Award. He belongs to IEEE and its HKN, Computer and Education Societies, as well as the American Society for Engineering Education and its Electrical and Computer Engineering, Educational Research and Methods, and First Year Programs divisions. In these groups, he helps deliver engineering education conferences, webinars, and certificate programs. He leads teams accrediting engineering degrees as an Engineering Area Commissioner in ABET. IEEE elevated him to Fellow for contributions to global online engineering education. And, the International Society for Engineering Education bestowed International Engineering Educator Honoris Causa for outstanding contributions in engineering education.

Deborah Trytten

Dr. Deborah A. Trytten is a Professor of Computer Science and Womens' and Gender Studies at the University of Oklahoma. Her main research focus is diversity in engineering education and introductory software engineering education.

Computational Thinking in the Formation of Engineers: Year 2

Introduction

In the United States, engineering students spend four or more years studying mathematics, science, and engineering topics that provide breadth and depth in a field of study. The structure of the engineering curriculum is not nationally standardized but accreditation requirements, professional society guidelines, and input from industrial advisory committees all help universities develop robust curricula that continue to produce graduates prepared to design systems around multiple realistic constraints using modern tools and practices [1] [2]. Central to this modern design practice is the use of computers to collect and analyze data, as well as to calculate solutions to numerical problems or automate control processes. Not all engineers use computers in the same way. In some engineering fields, the use of spreadsheets and statistical software may be sufficient, while in other fields proficiency in software development must be established. A unifying theme, however, is successful development of a set of computational thinking skills that enable innovative design [3] [4].

A computational thinker can ***decompose*** problems into smaller and more manageable subparts, create ***abstract models*** for each subpart, choose appropriate ***data representations*** to support the models, ***write algorithms*** to manipulate the models and automate solutions, and ***understand the impact*** of the computing solution on the society of users. At many institutions, first-year engineering courses introduce these ideas in computational thinking and prior work discovered these courses can be a barrier to enculturation as an engineer [5]. Students find computational thinking topics difficult to master. And social identity as well as inequities in high-school preparation and technology access all add variables to the first-year experience that can impede the progress of students [6] [7] [8].

This mixed-methods research project seeks improvements in the way computational thinking is taught in college level engineering courses by understanding the multiple factors that affect computational thinking development. The overarching goal is helping students from a wide variety of social identities succeed in enculturation as an engineer. The research is guided by three fundamental research questions:

1) How does the integration of computing into the foundational engineering courses affect the formation of engineers?
2) In what ways do social identities (e.g., gender, ethnicity, first-generation status, socioeconomic status), choices (e.g., major, transfer status), and other factors impact the engineering student experience with computational thinking?
3) In what ways do computational thinking skills develop over time in engineering students?

Instrument Validation

Over the past two years, the research project sought answers to questions one and two while gathering longitudinal data for question three. The **first major research result from year two** is the successful validation of a quantitative multiple-choice instrument that can measure pre-post computational thinking skills in students. As reported previously, the initial effort in the first project year focused on the development of this Engineering Computational Thinking Diagnostic (ECTD) [9] [10]. Initial validation attempts on two beta versions with fifteen questions each resulted in statistical factor analysis suggesting the need to edit question and distractor text. A new version with the questions that performed best in factor analysis was formed with edited question and distractor text. The new ECTD was tested in Spring 2021. Exploratory factor analysis was again conducted by the statistical consultant. All eigenvalues and correlations between items confirmed the instrument validated against a single factor called ***computational thinking*** [11].

An example ECTD question testing algorithmic skill is shown in Example 1 with the correct answer in bold.

> A computer program counts the number of times each word appears in a text file. The program converts all words to lowercase and prints an alphabetized table of words and counts. For example, if the word "the" appears 20 times, and the word "The" appears 34 times, then the final table of words must include the word "the" and the count 54. Which set of ordered steps represents a correct solution?
>
> A. count words, sort words, print table
> B. sort words, count words, convert words to lower case, print table
> C. count words, convert words to lower case, print table
> **D. convert words to lower case, sort words, count words, print table**
> E. sort words, count words, print table

Example 1: An instrument question designed to test algorithmic skill

The key requirement in this question is an alphabetically sorted list of words without variation in capitalization. This requires an algorithm that includes both capitalization uniformity and sorting; only answer item D contains both in the proper order – along with the other requirements of counting and printing a table.

An example social impact of technology question from the validated ECTD is shown in Example 2 with the correct answer in bold.

> Many companies are competing to bring self-driving cars to the marketplace. Which of the following impacts on society best motivates development of this technology?
>
> A. Self-driving cars allow passengers to have more meaningful face-to-face conversations.
> B. Self-driving cars eliminate the need for ridesharing services like Uber or Lyft.
> **C. Self-driving cars will decrease accidents since software will not get distracted like humans.**
> D. Self-driving cars will cost significantly more than cars driven by humans.

Example 2: An instrument question designed to test social impact understanding

The ethics of engineering and its impact on society are an important part of cognitive development and enculturation. While the distractors in Example 2 have degrees of correctness, the best answer is the life-saving advantage of automated technology to society.

Implementation of Position-of-Stress Surveys

The **second major research result from year two** is the implementation of a quantitative three-question survey administered to students as they complete activities that are judged to be positions of academic stress. The goal was identification of student cohorts that were gaining, losing, or remaining the same in confidence about major choice as they study computational thinking. Study participants answered these questions at each stress point:

1) How many minutes did you take to complete this exercise?
   Answered using a free-form text box

2) How difficult was the set of questions you just submitted?
   Answered using a sliding-scale from 0 to 100 with scale text “Very Easy” at 0 and scale text “Very Difficult” at 100.

3) How confident do you feel about your choice of major in engineering or computing after completing this exercise?
   Answered using a sliding-scale from 0 to 100 with scale text “Having Serious Doubts” at 0 and “Very Confident” at 100.

The research plan identified the pre-test ECTD, the midterm exam, and the post-test ECTD as stress points where student confidence may be changing. The ECTD and position-of-stress survey was given to all students enrolled in a first-year introductory engineering course at a large Southwestern university. The difference between initial confidence and midterm confidence was calculated to identify students with increasing, decreasing, and unchanged confidence about major choice.

Semi-structured Interviews Reveal Privilege

The position-of-stress surveys provided a strong data set about confidence in major choice. The research team used stratified sampling on this data to choose participants to invite to semi-structured interviews to gather more nuanced qualitative data about how computational thinking impacts engineering enculturation. Five stratified subpopulations were formed: confidence level and race, confidence level and gender, confidence level and first-generation status, confidence level and identification as academically talented, and drastic changes in confidence level. Invitations to participate were sent by email in three phases to a total of 70 students and 27 semi-structured interviews were conducted.

The interview protocol required the interviewer to gain permission for audio-only recording and to then ask a series of prompts designed to encourage twenty minutes of discussion about pre-university technology access, pre-university course access, the level of pre-university computing experience, attitudes about engineering and computing, influence from family or friends with engineering or computing background, comfort with mathematics and science, the pathway taken to engineering matriculation, social identity and its impact on current confidence, and the impact of the current engineering course on confidence. The interviewer supplemented the protocol with additional questions as necessary to understand participant responses and allow participants to augment their initial response. The research team includes faculty members at multiple institutions and these faculty members each conducted interviews with students that were not at their own institution in all but two cases. In these two outlier cases, the interviewer was from the same institution but taught courses in a college other than engineering – preserving the anonymity of the participants.

A complete record of the qualitative research methodology has been published [12]. In summary, an iterative and inductive methodology was used by five trained coders to establish a rich set of qualitative data from automatically transcribed recordings with human editing. Categorization of participants was completed based on coding similarity and one pathway narrative for each category was created. While computational thinking involves skills beyond programming, programming emerged as a theme in categorization. The categories used to write pathway narratives were:

1) Increasing confidence with prior programming success
2) Increasing confidence without prior programming success
3) Decreasing confidence with prior programming success
4) Decreasing confidence without prior programming success

The **third major research result from year two** is this set of pathway narratives that confirm well-known stereotype, grandstanding, and inequity effects in classroom learning and enculturation [7] [8] [13] [14] [15] [16] [17]. The pathway narratives also revealed that students with the privilege of prior programming experience excelled in the first-year engineering courses while students without the same privilege were struggling and questioning the choice of major. This hidden computing privilege was demonstrated in the narratives as access to advanced pre-university mathematics and computing coursework as well as access to computing technology. Participants with the hidden computing privilege matriculated to engineering with a set of skills that were not institutional entry requirements and many did not recognize that they were better prepared than classmates. When a participant did recognize the privilege, the consensus was that knowing programming in advance helped greatly with the course [12].

Underrepresented groups often experience inequity in pre-university computing coursework and access to computing technology. This lack of privilege is a hazard that creates a barrier to enculturation as an engineer, and may work against efforts to improve diversity, equity, and inclusion in Colleges of Engineering. When institutions add computing in the first year but do not provide a curricular mechanism to address this hazard, it leaves people marginalized and unsupported.

Future Work and Broader Impact

In year two, data collection was piloted at one of the three institutions funded by the National Science Foundation grant. In year three, mixed-method data collection will expand to all three institutions and continue gathering longitudinal data about the first-year experience. The data will allow exploration of how institutional type – with the associated differences in curriculum, student support, residential life, and privileges (including but not limited to the privilege of having prior computer programming experience) – impacts first-year student success in acquiring computational thinking and enculturation as an engineer. It will also seek to distill common student pathways with the hope of broadly informing curriculum design so that the resulting curriculum supports the inclusion of any student lacking prior computing privilege. The broader impact is an informed academy preparing an inclusive community of diverse engineers to solve today's engineering problems through computational thinking and computer-assisted design.

Acknowledgements

This work is possible through the support of the NSF:RFE award #1917359 "Collaborative Research in Improving Computational Thinking in the Formation of Engineers, A Multi-Institutional Initiative."

Bibliography

[1] "Criteria for Accrediting Engineering Programs, 2021 – 2022 | ABET." https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2021-2022/ (accessed Jul. 09, 2021).
[2] "Computing Curricula | IEEE Computer Society." https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/curricula (accessed Aug. 30, 2021).
[3] C. L. Isbell *et al.*, "(Re)defining computing curricula by (re)defining computing," *SIGCSE Bull.*, vol. 41, no. 4, pp. 195–207, Jan. 2010, doi: 10.1145/1709424.1709462.
[4] P. J. Denning, "The profession of IT: beyond computational thinking," *Commun. ACM*, vol. 52, no. 6, pp. 28–30, Jun. 2009, doi: 10.1145/1516046.1516054.
[5] N. V. Mendoza Diaz, S. Y. Yoon, J. C. Richard, and T. D. Wickliff, "Exploring enculturation in the first-year engineering program (Year II)," presented at the 2018 ASEE Annual Conference & Exposition, 2018. Accessed: Jul. 01, 2020. [Online]. Available: https://peer.asee.org/29860
[6] J. Margolis and A. Fisher, *Unlocking the clubhouse: women in computing*, Revised edition. Cambridge, Mass: The MIT Press, 2003.
[7] J. Margolis, R. Estrella, J. Goode, J. J. Holme, and K. Nao, *Stuck in the shallow end: education, race, and computing*. Cambridge, MA: The MIT Press, 2010.
[8] R. Varma, "Making computer science minority-friendly," *Commun. ACM*, vol. 49, no. 2, pp. 129–134, Feb. 2006, doi: 10.1145/1113034.1113041.
[9] N. V. Mendoza Diaz, D. A. Trytten, R. Meier, and Y. S. Yoon, "An Engineering Computational Thinking Diagnostic: A Psychometric Analysis," presented at the Frontiers in Education (FIE), Lincoln, Nebraska, USA, 2021.
[10] N. V. Mendoza Diaz, R. Meier, D. A. Trytten, and S. Yoon Yoon, "Computational Thinking Growth During a First-Year Engineering Course," in *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, Oct. 2020, pp. 1–7. doi: 10.1109/FIE44824.2020.9274250.

[11] N. V. M. Diaz *et al.*, “Computational Thinking in the Formation of Engineers (Year 1),” presented at the 2021 ASEE Virtual Annual Conference Content Access, Jul. 2021. Accessed: May 12, 2022. [Online]. Available: https://peer.asee.org/computational-thinking-in-the-formation-of-engineers-year-1

[12] N. V. Mendoza Diaz, D. A. Trytten, R. Meier, and J. M. Moore, “The Impact of Prior Programming Experience on Computational Thinking in First-Year Engineering Experience.,” presented at the 2022 CoNECD (Collaborative Network for Engineering & Computing Diversity), Feb. 2022. Accessed: May 12, 2022. [Online]. Available: https://peer.asee.org/the-impact-of-prior-programming-experience-on-computational-thinking-in-first-year-engineering-experience

[13] J. B. Main and C. Schimpf, “The Underrepresentation of Women in Computing Fields: A Synthesis of Literature Using a Life Course Perspective,” *IEEE Transactions on Education*, vol. 60, no. 4, pp. 296–304, Nov. 2017, doi: 10.1109/TE.2017.2704060.

[14] M. Meyer and S. Marx, “Engineering dropouts: a qualitative examination of why undergraduates leave engineering,” *Journal of Engineering Education*, vol. 103, no. 4, pp. 525–548, 2014, doi: 10.1002/jee.20054.

[15] D. A. Trytten, A. W. Lowe, and S. E. Walden, “‘Asians are Good at Math. What an Awful Stereotype’ The Model Minority Stereotype’s Impact on Asian American Engineering Students,” *Journal of Engineering Education*, vol. 101, no. 3, pp. 439–468, 2012, doi: 10.1002/j.2168-9830.2012.tb00057.x.

[16] C. E. Foor, S. E. Walden, R. L. Shehab, and D. A. Trytten, “‘We weren’t intentionally excluding them...just old habits’: Women, (lack of) interest and an engineering student competition team,” in *2013 IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, OK, USA, Oct. 2013, pp. 349–355. doi: 10.1109/FIE.2013.6684846.

[17] C. M. Steele, “A threat in the air: How stereotypes shape intellectual identity and performance.,” *American Psychologist*, vol. 52, no. 6, pp. 613–629, 1997, doi: 10.1037/0003-066X.52.6.613.