

Work in Progress: Analyzing a Distributed Expertise Model in an Undergraduate Engineering Course

Ms. Sara Willner-Giwerc , Tufts University

Sara Willner-Giwerc is currently pursuing a Ph.D. in mechanical engineering at Tufts University. She is a National Science Foundation Graduate Research Fellow, which supports her research at the Tufts Center for Engineering Education and Outreach (CEEEO) on technological tools and pedagogical approaches for facilitating hands on learning in educational settings.

Dr. Kristen B Wendell, Tufts University

Kristen Wendell is Assistant Professor of Mechanical Engineering and Adjunct Assistant Professor of Education at Tufts University. Her research efforts at at the Center for Engineering Education and Outreach focus on supporting discourse and design practices during K-12, teacher education, and college-level engineering learning experiences, and increasing access to engineering in the elementary school experience, especially in under-resourced schools. In 2016 she was a recipient of the U.S. Presidential Early Career Award for Scientists and Engineers (PECASE). <https://engineering.tufts.edu/me/people/faculty/kristen-bethke-wendell>

Work in Progress: Analyzing a Distributed Expertise Model in an Undergraduate Engineering Course

Abstract:

This work in progress presents an analysis of a *distributed expertise* approach to teaching computational thinking in a first-year undergraduate engineering course. Using student artifacts, surveys, interviews, and class observations, this mixed methods comparative case study explores the following two research questions: (1) In a course that uses a distributed expertise model, in what ways do students demonstrate knowledge and competency in computer science fundamentals, data collection methods, data analysis techniques, and data communication, and how does this compare to students taught in a traditional model? and (2) How does the complexity, solution diversity, functionality, and emotional investment in students' final projects compare between a distributed expertise model and a traditional instructional model?

Introduction:

As computer science became ubiquitous and important in almost every field in the 1950s, educators realized that learners in many subjects outside of computer science could benefit from thinking like computer scientists. As a result, the term “computational thinking” emerged as a way to refer to the thought processes involved in solving a complex problem [1]. Between the 1950s and the early 2000s, computational thinking has been defined in numerous different ways by both industry professionals and educators alike. While an exact definition of the term is not widely agreed upon, the one consensus is on the importance of teaching computational thinking concepts and practices to pre-college and college students as they prepare for careers in a highly technical world [2]. In 2006, the CS-for-All movement and other educational initiatives determined that computational thinking was a fundamental skill that all students, regardless of their intention to pursue a career in a technical field, should learn [3]. Many universities, including Carnegie Mellon, followed suit and began offering computational thinking courses for non-computer science majors [4].

Many of the computational thinking courses that emerged are taught using a lecture based approach, where students spend the majority of class time listening to an instructor talk about computational thinking concepts [5]. This type of learning is referred to as passive learning, and has been shown to be less effective than active learning, which involves instruction where students are engaged in the learning process through participating in discussions, creating representations, or otherwise working with the course content [6]. One such active learning pedagogy is called *jigsaw learning*, which was first proposed by Elliot Aronson [7]. In jigsaw learning, different groups of students are assigned different pieces of a larger content area. Each group is responsible for constructing knowledge in one sub-domain of the larger area and is responsible for teaching it to the rest of the class so that together everyone learns all of the content [7]. Jigsaw learning is widely used in K-12 education and at the college level for a variety of subjects [8,9], and has been applied in some cases to engineering courses [10].

Distributed expertise is similar to jigsaw learning. However in a distributed expertise model, the goal is not for students to directly teach each other the content after they have specialized in one area. Instead, the distributed expertise model embraces the idea that different students will develop different knowledge sets [11]. In this study, we are leveraging the distributed expertise

model for teaching computational thinking in a first year engineering course. We argue that this approach could have a large positive learning impact for students and alleviate some of the current challenges facing computational thinking education for first year engineering students. We hypothesize that using a distributed expertise model will create a more authentic collaboration between students, enable students to construct deeper and more applicable knowledge, and allow for the completion of more technically sophisticated projects.

Background:

Introduction to Computing in Engineering is a course required for all 200 first-year engineering students at a research university in the northeastern United States. Since it was first introduced, the course has undergone several iterations, and recently the course coordinator concluded that the current approach of gathering all 200 students for one large-enrollment course was not meeting all students' needs. As a result, the university approved four different new proposals for the format of the course and allowed all four versions to run in the Spring 2019 semester. This decision created a fruitful context for a comparative case study across different instructional approaches. Two of the professors are using a lecture and lab based instructional model, where students attend two, 75-minute lectures each week and learn about computational concepts using MATLAB. Students then practice these concepts during one 75-minute lab section each week run by a teaching assistant (TA), and they also complete weekly homework assignments using MATLAB. The course culminates in a final project of students' own choosing to demonstrate their knowledge of both MATLAB as a tool and computational thinking concepts more generally. Each of these lecture-based sections contained approximately 50 students each.

The third section uses a mixed lecture and project based approach using Python and the BBC micro:bit. The BBC micro:bit is a low-cost, hand-held microprocessor that can be programmed in Python. Students in this section attend three, 75-minute classes each week. In each class, the professor first offers a mini-lecture on computational concepts with live coding demonstrations. Then the bulk of the class period is devoted to students collaborating on projects and homework while the professor and TA's circulate to answer questions. This section also culminates in a final project where all students apply content knowledge developed throughout the course. Approximately 30 students are enrolled in this section.

The fourth section uses a distributed expertise approach. Students attend three, 75-minute class sessions per week. These classes are mostly spent working in small groups on computational thinking projects using Raspberry Pi's and Python with the support of the professor and TAs. Raspberry Pi is a small, low cost programmable computer. Occasionally the professor conducts short demonstrations. Students spent the beginning of the semester learning as one large group and then broke into specialty groups for the middle portion of the semester. There are four specialty groups: computer vision and image processing, sensing and actuating, data acquisition and processing, and data analytics and visualization. Students spent the remainder of the semester working in final project groups that consisted of one student from each of the specialty groups. The final project involves constructing a warehouse robot that can: receive remote instructions and broadcast findings wirelessly, visually identify a location, plan an efficient path, navigate an unknown warehouse layout, identify a product, and coordinate all of these actions between multiple devices. This project is designed to require knowledge developed in each of the specializations, thus leveraging and requiring the expertise distributed throughout the group. Approximately 70 students are enrolled in this section.

Methods:

Using data from surveys, interviews, student artifacts, and class observations, we are conducting a mixed methods comparative case study to address two research questions: (1) In a course that uses a distributed expertise model, in what ways do students demonstrate knowledge and competency in computer science fundamentals, data collection methods, data analysis techniques, and data communication, and how does this compare to students taught in a traditional model? and (2) How does the complexity, solution diversity, functionality, and emotional investment in students' final projects compare between a distributed expertise model and a traditional content delivery model [12]? Table 1 describes the purpose of each data set collected over the course of the semester.

Table 1. Data Types Collected and Justification

<i>Data Type</i>	<i>Reason Collected</i>
Pre/Post Surveys	<ul style="list-style-type: none">• Demographic information• Measure shifts in attitudes and opinions about computational thinking• Measure growth in computational content knowledge
Collaboration Surveys	<ul style="list-style-type: none">• Indicate whether or not the distributed expertise model had any impact on group dynamics.
One-on-one Student Interviews	<ul style="list-style-type: none">• Hear detailed accounts of learning• Better understand how students perceive computational thinking and engage with the content.
Student Artifacts	<ul style="list-style-type: none">• All student coursework (homework assignments, projects, lab reports/submissions, exams, etc.) was collected across all sections• Artifacts will be analyzed using standard metrics to try and gain insights into the types of learning that occurred throughout the semester
Classroom Observations/Videos	<ul style="list-style-type: none">• Understand the logistical setup of the classroom environment• Witness the ways learning manifested in each of the different sections.

Preliminary Results and Future Work:

Data processing and analysis are ongoing. However, early analysis of student artifacts has revealed that students in the distributed expertise course showed competency in computer science fundamentals, data collection methods, data analysis techniques, and data communication just as successfully or more successfully than students taught using more traditional approaches. The final projects from the distributed expertise model showed high levels of complexity, solution diversity, and student emotional investment. Additionally, student interviews and surveys revealed that using a distributed expertise model had a positive impact on group dynamics, lead to a more authentic group experience, and had a positive impact on students' confidence in their computational thinking abilities. In the presentation we will share student artifacts and findings from the analysis of the collected data sources and illustrate evidence of the positive impact the distributed expertise model had on learning outcomes.

Acknowledgements:

This material is based upon work supported by the National Science Foundation (NSF) Graduate Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References:

- [1] P. J. Denning, "Viewpoint Remaining Trouble Spots with Computational Thinking," pp. 33–39, 2016.
- [2] U. Ilic, H. I. Haseski, and U. Tugtekin, "Publication Trends Over 10 Years of Computational Thinking Research," *Contemp. Educ. Technol.*, vol. 9, no. 2, pp. 131–153, 2018.
- [3] C. Concepts, *A Framework for K-12 Science Education*. 2012.
- [4] J. M. Wing, "Computational Thinking : What and Why ?," no. November, pp. 1–6, 2010.
- [5] T. T. Yuen and K. A. Robbins, "A Qualitative Study of Students' Computational Thinking Skills in a Data-Driven Computing Class," *ACM Trans. Comput. Educ.*, vol. 14, no. 4, pp. 1–19, 2014.
- [6] R. Maceiras, A. Cancela, S. Urréjola, and A. Sánchez, "Experience of cooperative learning in engineering," *Eur. J. Eng. Educ.*, vol. 36, no. 1, pp. 13–19, 2011.
- [7] E. Aronson, *The jigsaw classroom*. Sage, 1978.
- [8] S. Holloway, S. Tilleman, and R. Macy, "Active Learning in Entrepreneurship: Applying the Jigsaw Method to Entrepreneurship Instruction," *USASBE 2008 Proc.*, no. August 2014, pp. 1–6, 2008.
- [9] M. N. Gömleksiz, "Effectiveness of cooperative learning (jigsaw II) method in teaching English as a foreign language to engineering students (Case of Firat University, Turkey)," *Eur. J. Eng. Educ.*, vol. 32, no. 5, pp. 613–625, 2007.
- [10] N. M. Tahir, K. A. Othman, and F. H. Yahaya, "Case study of jigsaw cooperative learning effect within Electrical Engineering courses," in *2011 International Conference on Business, Engineering and Industrial Applications*, 2011, pp. 20–23.
- [11] A. L. Brown, D. Ash, M. Rutherford, A. Gordon, and J. C. Campione, "Distributed expertise in the classroom," *Distrib. Cogn. Psychol. Educ. considerations*, pp. 188–288, 1991.
- [12] S. B. Merriam, *Qualitative Research and Case Study Applications in Education. Revised and Expanded from " Case Study Research in Education."*. ERIC, 1998.