Analyzing Methods for Promoting Inclusion and Diversity in Computer Science Education

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Adam Emerson Marcus

Spring 2021

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Sean M. Ferguson, Department of Engineering and Society

According to the U.S. Bureau of Labor Statistics, the number of computer science jobs available is expected to grow 15% from 2019-2029, much faster than the average compared to other fields (Computer and Information Research Scientists: Occupational Outlook Handbook, 2020). Given the presence of technology around us in nearly every aspect of life, it is essential that the future generations of computer scientists be inclusive of a greater variety of people from different backgrounds. Despite the rapid growth of the field, the next group of students looking to study computer science is still extremely misrepresentative of the population. This inequity can be attributed to the fact that several groups such as women, Black, and Hispanic do not continue on to upper level computer science courses at the same rate as other demographics (Ericson & Guzdial, 2014). There have been several different attempts at promoting inclusion in computer science education which have had varying levels of success. In this context, diversity refers to the demographics of the students in the field, and inclusion is defined as increasing the opportunities and reach for potential students.

It is a widespread belief that "different sectors should be represented in the process to define and develop technology" (Garcia-Holgado et al. 2019). While it would be ideal if there was interest in pursuing careers in technology from every person, that is simply not the case. According to a study by Tai, Liu, Maltese, and Fan, "students with expectations for a science related career were 3.4 times more likely to earn physical science and engineering degrees that students without similar expectations" (2006). Due to this expectation-driven divergence, the question becomes one of how to engage students that are pursuing other fields so that people similar to them might be represented in the design and development of future technologies. There are currently some technologies with this in mind such as Arduino, a company that produces microcontrollers and microcontroller kits that enable users to do many different things such as create wearable technologies by sewing the microcontroller and sensors into fabrics (Beuchley et al., 2008). This concept can also be expanded into other fields in the arts besides clothing design, such as drawing or creating music.

My STS research focuses on the issues of diversity and inclusion in computer science education. Several case studies are examined to determine the relationships between the actors in each. These case studies surrounding diversity and inclusion in computer science education can be broken into three main types: a systematic approach to change all courses, an individualized approach to reach underrepresented students, and a mix of the two that examines single classes aimed at promoting inclusion at the individual level.

I explore different examples and methods for inclusion in computer science education, analyzing several interactions in each case study reviewed by applying Actor Network Theory. This theory states that any person or piece of technology forms connections with others, and can be used to highlight existing networks comprised of these actors. One key point of actor network theory is that all people and technologies interact on the same level (Sismondo, 2010). This means that one can look at a person-person interaction in the same way as a person-technology interaction to better understand why certain methods succeed. After reviewing the three methods, a recommendation is made on the best approach to promoting diversity and inclusion in computer science education.

Systematic Approach to Inclusion

The problem of creating more diversity and inclusion in computer science can be approached in several different ways. One of the more common and widely studied approaches is a systematic one which aims to promote diversity at several different schools rather than one individual program. One organization, Technology Education and Literacy in Schools (TEALS), connects schools with “trained technology industry professionals who volunteer their time throughout a school year” (Ibe et al., 2018). TEALS focuses on adding actors in the form of professionals with relevant technological experience to the existing network of each school they work with. The TEALS program specifically created a Diversity, Equity and Inclusion Working Group (DEIWG) with the goal of identifying and assessing goals and changes they could make to the program to achieve those goals (Ibe et al., 2018). The DEIWG identifies five key program areas: school recruitment and selection, student recruitment and engagement, volunteer recruitment and placement, instructional support, and curriculum. Each of these areas centers around a different set of actors: school administrators, students, volunteers, teachers, and experts respectively; however, the interactions between all of the groups with one another and the overall effect of this strategy on diversity in the schools studied are noted.

School recruitment and selection must be considered by the DEIWG at the highest level, as increasing their reach can be an excellent route to becoming more diverse. In particular, they can target schools with higher proportions of underrepresented groups in computer science. However, identifying such schools is only half of the battle. Administrators whose schools are already a part of the TEALS program are essential to helping the program grow. They act on behalf of their own students when initially applying to be a part of TEALS, but afterwards they must remain active to ensure its success (Granor, DeLyser, & Wang, 2016). The key interaction that can help the program grow to more areas is between administrators of schools that are already in TEALS and those that are not. This communication can help to expand the size of the TEALS network and circulate ideas of diversity in computer science amongst those with the power to bring about positive change.

Shifting the scope down, once a school has joined TEALS the next consideration deals with student recruitment and engagement. The breakdown of students in computer science courses is the key metric for measuring the diversity of the TEALS program. Each year the goal is to have a demographic distribution closer to that of the US national average in high schools (or higher proportions for the historically underrepresented groups). Student recruitment within a school can take several forms. Among the students, social factors can be a driving force that prompts each to choose computer science classes; however, one main indicator on if a student will enroll in a computer science course is their level of interest in the subject. As a result, teachers and administrators can take matters into their own hands. Teachers can have an immense impact on student interest levels by promoting awareness of computer science classes and resources with the students. Such resources are provided by TEALS to the schools for this reason. Administrators on the other hand can affect which courses students take in a more direct manner: by choosing the students' classes each year. At the high school level, administrators can ensure that underrepresented students are those who are placed in computer science courses to increase their exposure to the subject. This was the case in TEALS affiliated New York schools, which has seen their computer science gender diversity rise from under 40% female to nearly an even split in just a few years, much quicker than other states' schools are closing the gap. In regards to racial diversity, the underrepresented groups show a preference towards administrator, teacher, and counselor recommendation as the main reason they enroll in a computer science course (Ibe et al., 2018). Each of these scenarios relies on supervisors in a different manner. Teacher and counselor input influences student interest and participation in computer science courses, whereas administrator intervention can bypass interactions with students completely. Seeing as both approaches have seen success, the best practice for improving diverse student recruitment is a combination of targeted recommendations from teachers and counselors to their students, coupled with administrative oversight on which students are placed in computer science classes.

The next step once the right students are in the classroom is to ensure that the instruction is clear, effective, and engaging. In the TEALS system, a group of volunteers works in the classroom with one teacher for two years. Since the teacher does not need any prior computer science experience, they progressively transition the course from volunteer taught to instructor taught. Volunteers initially take the role of the teacher during class time, emphasizing the importance of three actor interactions: volunteer-student, teacher-student, and volunteer-teacher. As with any classroom setting, the teacher-student relationship is crucial, but in the method used by TEALS the volunteer-student relationship takes the same importance. Volunteers for the program are usually software developers without any kind of educational training, meaning they need to learn the pedagogy for teaching their students. As a result, the volunteer-teacher interactions are the most essential to the success of the TEALS program at a school. The teachers and volunteers begin as the two halves of one fully equipped computer science teacher: one with expertise in the subject matter and one with expertise in disseminating such information. They must combine their knowledge of pedagogy and computer science respectively, and must work well together to achieve their goal of transitioning a teacher from any subject to be a computer science teacher. In order to speed this process up, volunteers can be given a crash course in teaching and teachers can be given one in computer science, but since they will be working together as a team in the classroom for at least two years, the relationship and interactions between each of these actors needs to be positive in order to be effective. In addition, a strong partnership between a volunteer and teacher not only prepares the teacher better for teaching their own computer science courses in the future, it also helps the volunteers to be better prepared for helping teachers who are new to the program. Their chemistry is considered by TEALS when vetting their candidates for each position, and pairings remain subject to change if it is clear that a partnership will not be successful. In addition, partnerships that are deemed to be successful will often be expanded. Offering another computer science course with the same teacher-volunteer pairings and restarting the two year process, while the teacher simultaneously takes over the first computer science class on their own (Granor, DeLyser, & Wang, 2016).

In analyzing a specific, systematic approach to promoting diversity, equity, and inclusion in high school level computer science, certain actor-actor interactions stand out as keys to a strong program. Specifically, administrator connections across schools, school faculty involvement with students, and volunteer synergy with their partnered teacher should all be heavily emphasized in order to cast the widest, most effective net using a systematic approach like that used by TEALS.

Individualized Approach to Inclusion

Another way that inclusion can be promoted in computer science is through a more individualized approach. Rather than attacking the lack of diversity by going through the underlying emphasis across all computer science courses, a single class or even a single person can be targeted in an attempt to increase diversity. A case study by Dennehy and Dasgupta in 2017 employed the tactic of one on one peer mentoring in order to study retention rates and self-efficacy for female students pursuing engineering degrees. The methods used in this study rely on human to human relationships as the method of promoting diversity directly rather than through organizational goals. The results of this study indicate higher levels of both retention and self-efficacy among students with female peer mentors. These increases make the individualistic approach more appealing as there is a positive relationship between levels of self-efficacy in education and GPA for a diverse group of students (Majer, 2009).

In Dennehy and Dasgupta's case study, there are only two types of actors who are involved: the student and the mentor. One key distinction is that their results indicated higher retention and self-efficacy for female students who had other female mentors, and they make sure to point out that this was not necessarily the case for women who had male peer mentors (2009). This disparity introduces the idea of gender and race matching when creating mentor-mentee pairs. According to a study by Blake-Beard, Bayne, Crosby, and Miller, "having a mentor of one's own gender or race was felt to be important by many students, especially women and students of color" (2011). This lends itself to the belief that the mentor and mentee should share characteristics for the individualized approach to be more effective. Rather than underscoring their interaction from a purely academic standpoint, each actor should share some cultural and/or social similarities, highlighting comfort and confidence in the STEM community for underrepresented groups as they work closely with a mentor who exemplifies success.

Peer mentoring is the most common individualized approach that lends itself to inclusion of underrepresented groups due to its ease of implementation. All that is needed to form these relationships are people who have gone through similar experiences that a new student is about to embark on. Based on the discussions of effective peer mentoring pairings, I would recommend that universities assign every incoming student a peer mentor based on background with strong consideration to gender and race. More effective peer mentoring programs for underrepresented students will in turn increase retention and self-efficacy among those groups leading to increased diversity in computer science programs.

Hybrid Approach to Inclusion

One more method to be examined is the promotion of diversity through a combination of direct person to person communication within the more systematic setting of an individual computer science course. This method emphasizes the connection between a person and a piece of technology rather than other people or organizations. Many schools now have courses focused around student-technology interactions by using hands on tools such as LEGO Mindstorms robots or the LilyPad Arduino to promote interest from all different groups of students to enroll (Beuchley et al., 2008).

Computer science education is quite abstract in the sense that you rarely have anything tangible for students to interact with. Rather they can simply see outputs on their screen, becoming somewhat monotonous. The LEGO Mindstorms kit aims to simplify the extremely complex task of building a robot into an issue of assembling their blocks however you like and customizing actions using code. It is extremely flexible by allowing you to customize what your robot does in countless different ways with included motion, light, and touch sensors. The most important quality for our purposes however is user-friendliness in the form of a programming interface which "models programming as a process of dragging puzzle pieces (representing program steps) together to produce a chain (complete program)" (Klassner & Anderson, 2003). The design of LEGO Mindstorms is focused around its interaction with the user over the extent of its capabilities. Since it is an open-ended tool, rather than one with a single purpose to serve, its possibilities are primarily limited by what the user can imagine. In addition, its descendance from LEGOs, an extremely common toy used by children everywhere, increases its appeal to an extremely wide audience. The philosophy here for promoting inclusion of underrepresented groups is seeded in the audience it can reach, but is augmented by its "chameleonesque adaptivity which... permits the acknowledgement and embracing of different learning styles and epistemologies, engendering a convivial environment in which students can concretize their ideas and projects with intense personal engagement" (Blikstein, 2008).

Another technology, the LilyPad Arduino, represents a different approach for reaching underrepresented groups. Instead of capitalizing on the reach of the product, the LilyPad Arduino is designed to attract interest from non-STEM sectors. Similar to the LEGO Mindstorms kit, the LilyPad Arduino is an open ended technology. The Arduino kit includes a microcontroller in the form of a fabric patch and several different sensors that can be sewn into fabric to create e-textiles – wearable technology that can be controlled by user written code (Apiola, Lattu, & Pasanen, 2010; Beuchley et al., 2008). The versatility of the kit provides a creative outlet that can spark intrigue from those interested in fashion to try their hand in a computer science program or course centered around the Arduino. Its stake in fashion makes the Arduino kit an excellent candidate to increase diversity in computer science programs as the fashion industry is dominated by females at nearly 84.4% of fashion designers (Fashion Designers, 2019). This disparity represents an opportunity to expose women to an application of computer science that some would not have considered otherwise. It stands out in stark contrast with the traditional use of technology in schools, which "fosters students who are consumers of software and not constructors" (Blikstein, 2008). The Arduino's appeal to the fashion sector is a perfect example of its ability to expand the possibilities for people who usually use computers as a means of designing the visuals rather than the functionalities of their work.

The LEGO Mindstorms and LilyPad Arduino kits are similarly flexible, consisting of a computer to program and several sensors to read and control, but they relate to inclusion and diversity separately. The LEGO Mindstorms increases the widespread appeal to all types of people, which in turn can help include more people from underrepresented groups in computer science education. The LilyPad Arduino on the other hand looks to make computer science more appealing to those interested in fashion, a sector dominated by one of computer science's underrepresented groups. These two technologies unrestricted and creative nature makes them the perfect tools for the hybrid approach to promoting diversity as professors can design an entire course around them that can effectively teach computer science in an environment that appeals more to people outside of STEM.

Conclusion

The three different approaches to promoting diversity and inclusion each center around interactions with students, but from various actors and with various levels of depth. The systematic approach, which is highly structured and relies much more heavily on actors who are not students, seems to be the best approach for greater inclusion in the earlier stages of computer science education due to its wide reach. Simply exposing more students to computer science at an earlier time will provide the foot in the door that underrepresented groups need to continue into computer science with greater numbers. This approach is limited, however, due to its reliance on volunteers working in the field coming forward. The individualistic approach of peer mentoring is extremely promising due to the rate at which it can expand, with additional consideration needed when creating mentor-mentee pairs. The increased retention rates that result from peer mentoring make it an excellent option for widespread implementation at the college level in order to ensure that students from marginalized groups continue their studies in the field. The hybrid approach seems the most effective tool for inclusion, as it can appeal specifically to students who do not typically study computer science; however, it is much more difficult to expand quickly due to the fact that there are very few technologies with the capabilities of the LEGO Mindstorms and LilyPad Arduino kits. In addition, such technologies are not free like simple human-human interactions are. This introduces one of the main limitations of the hybrid approach in terms of accessibility. While a well funded university may be able to provide the students in such a course with the accompanying technology, the price tag becomes an issue for schools with less money.

References

Apiola, M., Lattu, M., & Pasanen, T. A. (2010, June). Creativity and intrinsic motivation in computer science education: experimenting with robots. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 199-203).

Blikstein, P. (2008). Travels in Troy with Freire: Technology as an agent of emancipation. *In Social Justice Education for Teachers* (pp. 205-235). Brill Sense.

Blake-Beard, S., Bayne, M. L., Crosby, F. J., & Muller, C. B. (2011). Matching by race and gender in mentoring relationships: Keeping our eyes on the prize. *Journal of Social issues*, *67*(3), 622-643.

Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008, April). The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 423-432).

Computer and Information Research Scientists: Occupational Outlook Handbook. (2020, September 01). Retrieved November 04, 2020, from https://www.bls.gov/ooh/computer-and-information-technology/computer-and-information-research-scientists.htm

Dennehy, T. C., & Dasgupta, N. (2017). Female peer mentors early in college increase women's positive academic experiences and retention in engineering. *Proceedings of the National Academy of Sciences*, *114*(23), 5964-5969.

Ericson, B., & Guzdial, M. (2014, March). Measuring demographics and performance in computer science education at a nationwide scale using AP CS data. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 217-222).

Fashion designers. (n.d.). https://datausa.io/profile/soc/fashion-designers.

Garcia-Holgado, A., Vázquez-Ingelmo, A., Verdugo-Castro, S., González, C., Gómez, M. C. S., & Garcia-Peñalvo, F. J. (2019, April). Actions to promote diversity in engineering studies: a case study in a Computer Science Degree. In *2019 IEEE Global Engineering Education Conference (EDUCON)* (pp. 793-800). IEEE.

Granor, N., DeLyser, L. A., & Wang, K. (2016, February). Teals: Teacher professional development using industry volunteers. *In Proceedings of the 47th acm technical symposium on computing science education* (pp. 60-65).

Ibe, N. A., Howsmon, R., Penney, L., Granor, N., DeLyser, L. A., & Wang, K. (2018, February). Reflections of a diversity, equity, and inclusion working group based on data from a national CS education program. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 711-716).

Klassner, F., & Anderson, S. D. (2003). Lego MindStorms: Not just for K-12 anymore. *IEEE robotics & automation magazine*, 10(2), 12-18.

Majer, J. M. (2009). Self-efficacy and academic success among ethnically diverse first-generation community college students. *Journal of Diversity in Higher Education*, *2*(4), 243.

Sismondo, S. (2010). *An introduction to science and technology studies* (2nd ed). Wiley-Blackwell.

Tai, R. H., Liu, C. Q., Maltese, A. V., & Fan, X. (2006). Planning early for careers in science. *Life sci*, *1*(0.2).