WIP: Detecting Computing-Enabled Interdisciplinary Domains Using the MIDFIELD Data Set

Tim Ransom
Engineering and Science Education
Clemson University
Clemson, United States
tsranso@clemson.edu

Dr. Stephanie Ashley Damas
Engineering and Science Education
Clemson University
Clemson, United States
damas@clemson.edu

Dr. D. Matthew Boyer
Engineering and Science Education
Clemson University
Clemson, USA
dmboyer@clemson.edu

***Abstract*—This WIP paper discusses the significance of interdisciplinary programs in computing, highlighting their role in addressing complex issues and attracting students to universities. It introduces the MIDFIELD project, which collects course records from multiple universities in the USA, enabling the analysis of course overlaps between interdisciplinary degrees and their parent fields. The study aims to inform curriculum design and promote interdisciplinary majors, particularly between computing and other STEM fields, by identifying shared courses. Such initiatives are crucial for diversifying the computing workforce and bridging the gap between students" experiences and computer science fundamentals.**

***Index Terms*—Education; Curriculum development**

I. Introduction

A. Purpose

This work analyzes the robust data set provided by the Multiple-Institution Database for Investigating Engineering Longitudinal Development (MIDFIELD) [1] to identify overlaps between interdisciplinary degree course requirements and their "parent" degree course requirements. The set of courses shared between curricula can indicate to curriculum designers the set of classes that other institutions have deemed sufficiently relevant to include and the courses offered through different departments. Identifying specialized courses is also valuable in evaluating efficient plans to create new departments at institutions with established "parent" programs. We conducted our analysis guided by the following research question: What are the most overlapping degree programs in the computing disciplines?

B. Study Motivation

Interdisciplinary studies integrate concepts, theories, and methods from multiple disciplines. In interdisciplinary majors, there is a deliberate effort to create a new perspective or approach that is not limited to any single discipline. This results in a deeper understanding of complex issues. Even further, interdisciplinary majors create space for impossible solutions within the confines of a single discipline [2]. While multidisciplinary majors have also been used to collaborate across disciplines, interdisciplinary majors use a combination of tools interchangeable from the distinct disciplines.

This study helps lay the groundwork for promoting interdisciplinary majors between computing and other STEM fields. Research has demonstrated how a diverse team of computing-enabled professionals contributes significantly to solving complex problems [3]. Implementing interdisciplinary majors can make computer science programs more attractive to students with marginalized identities who seek to bridge the gap between their lived experiences and computer science fundamentals [4] [5]. As such, this study aims to identify fundamental computer science courses that other programs can use and extend to create interdisciplinary majors.

C. Background

1) Computing Field: The computing domain has historically created new interdisciplinary programs as it engages with other knowledge branches [6]. These specialized programs offer a targeted approach to solving complex issues and act as a point of attraction for students to enroll in one University over another. These interdisciplinary programs' curricula often include courses from their respective "parent" fields. For example, a bioinformatics curriculum can consist of courses from both computer science departments and biology departments. Specialized courses are often also created in the "child" field to meet the needs of interdisciplinary students.

2) Knowledge, Skills, Abilities: Knowledge, skills, and abilities (KSAs) are derived from human resource management and organizational development to assess and describe the attributes required for a particular job or role [7]. Knowledge refers to the theoretical or practical understanding of a subject. Skills are the practical abilities and competencies that enable individuals to perform tasks effectively. Abilities are inherent or acquired traits that influence how effectively individuals can perform tasks or solve problems. Together, KSAs determine the outcome-based standard of individuals in a particular field [7] [8].

Each field develops KSAs widely accepted as baseline competencies for individuals who belong in said field [8]. In

the context of degree attainment, degree program curricula are motivated by providing students with the standardized KSAs in that field. For the computing field, this concept is highly applicable. Computing was originally developed to be leveraged to achieve a wide range of goals among other fields [6] [9]. At the onset of computing education, courses were created to ensure students attained the knowledge, skills, and abilities to be prepared for the workforce [6]. If one course is offered in multiple degree programs, then KSA transference claims the KSAs attained by students who take that course are included in both programs.

3) Need for Convergence: Many students matriculate through courses to develop the KSAs that constitute degrees in their major. By attaining a degree, students are expected to be prepared to enter the workforce and contribute to solutions to real-world problems. The National Science Foundation has identified several BIG Problems of our generation that require complex solutions guided by contributions from various fields. The NSF proposed growing convergence research as one of the top 10 BIG Ideas to address today's grand challenges [10]. The nation's grand challenges "will not be solved by one discipline alone" [11]. They require convergence: merging ideas, approaches and technologies from widely diverse fields of knowledge to stimulate innovation and discovery. To prepare our students to take on the grand challenges of our generation, we must evaluate curriculum development to consider the creation of more interdisciplinary studies.

4) Summary: This study analyzes courses offered in different majors to identify areas of overlap among existing computing majors. In identifying the extent to which majors overlap, institutions can begin to consider how to develop interdisciplinary degree programs from existing disciplinary degree programs seamlessly. We consider the curriculum level our unit of analysis to interpret this set of courses as representing the knowledge, skills, and abilities (KSAs) of the students who graduate with a given degree.

II. Methods

A. Data Source

The Multiple-Institution Database for Investigating Engineering Longitudinal Development (MIDFIELD) [1] project has collected complete course records of every student's path toward a degree from eighteen Universities across the United States of America. This data allows us to construct sets of courses that students choose to complete to fulfill their institution's requirements for graduation.

B. Data Analysis

For each CIP code in the MIDFIELD dataset, a set of courses is identified that any student who completed that degree took while progressing toward graduation. These sets include every course students took, including general education courses, duplication across students with more than one major, and courses students took that were not required for graduation. The MIDFIELD dataset has information that dates back to the 1980s, but we filter to data from 2000 onward to address the potential overrepresentation of older degree curricula. We have chosen for the time being to leave in these shortcomings of the data filtering process because of the subjectivity of deciding which courses develop skills that are less relevant to the curricula students are navigating.

In the future, we will be keeping a scalar value associated with each course to represent how often a course was taken. For example, and with randomly chosen courses: CPSC 1010 being taken by 99% of the computer science majors while ENG 3010 is only 10% indicates that CPSC 1010 is more core to the computer science curriculum.

We have designed a heuristic to correlate course codes and titles across various institutions over time for correlation detection. The MIDFIELD dataset provides an interpreted data course title that we use first. For courses that do not have a MIDFIELD title, we rely on the combination of department code and course number to indicate course content. For example, CPSC 2920 at a given institution is likely the same course two students took at the same institution referred to by CPSC 2920. Courses with the same name and number at different institutions are assumed to be different classes, and so were not counted as similar classes.

We then computed a percent similarity between these sets, determined by the degree of the set intersection divided by the set union's degree.

Due to the size of the MIDFIELD dataset, the authors implemented several performance optimization techniques with the R programming language to enable these comparisons on consumer-grade computing platforms.

III. Preliminary Results

We present three heatmaps indicating the rate of course overlaps between STEM programs in the MIDFIELD dataset. First, we show a section of computing-related programs and note a general trend of low course overlap (typically less than 10%). Next, we show a set of engineering degrees and note a generally higher degree of overlap, with a notable exception being software engineering. The final set of courses presented is selected to represent general stem courses, including programs such as Mathematics instead of more specialized degrees available.

A. Computing Programs

Initial results in figure 1 indicate a wide range of overlapping courses between computing degree programs. Overall, each computing-related degree has fairly low course overlap with the other degrees, with fewer than 10% shared courses. Information sciences and software engineering both have very low course overlap in particular. This could be for several reasons, including the students who enrolled in these programs adhered to structured/suggested curricula, did not complete secondary majors, or these majors have implemented their courses mostly in-department.

Several degree programs have low course overlaps as a trend, with one or two outstanding hot spots of shared courses. For example, computer science has very low crossover to

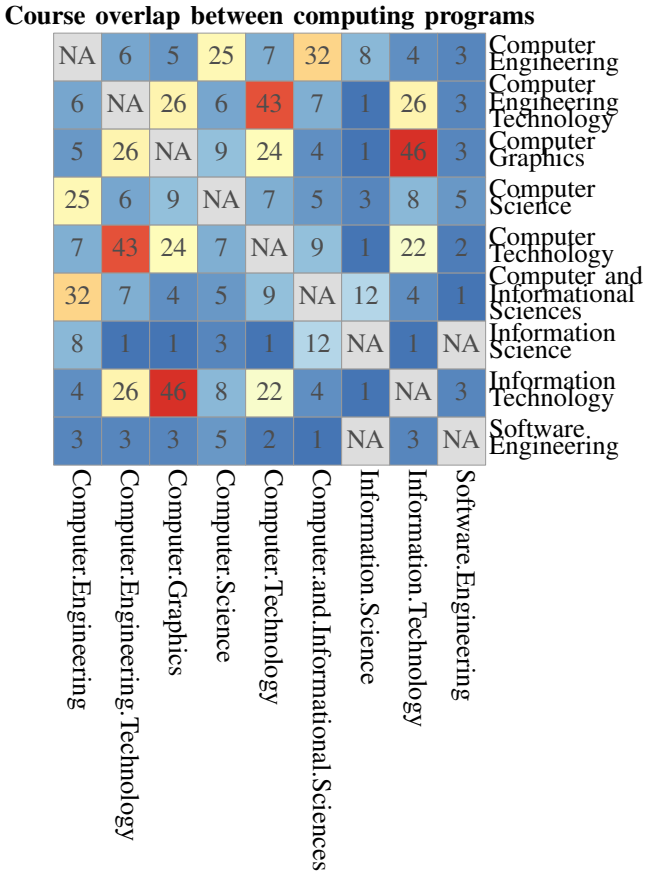

Fig. 1. Percent overlap of courses students completed majors in computing

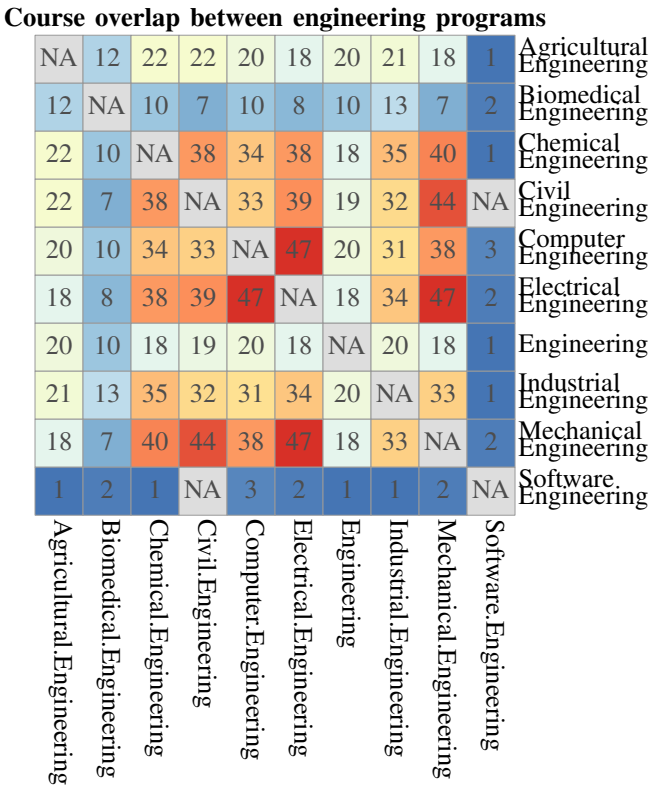

Fig. 2. Percent overlap of courses students completed majors in engineering

other computational programs except for computer engineering with 25% course overlap. Computer engineering also shares a relatively large course overlap with Computer and Informational Sciences at 32%. This could indicate that these particular programs share a common base set of KSAs and have an institutional organization to allow students to explore these with minimal resistance. For example, computer science students may have similar interests as computer engineering students, leading to a high cross-pollination between these two degree programs. The high course overlap shown in figure 1 indicates that this particular intersection is likely to hold interesting connections.

The largest course overlaps we detected among computing programs were the pairings of Computer Graphics / Information Technology (46%) and Computer Technology / Computer Engineering Technology (43%). These degrees are less frequently completed (in MIDFIELD data) than the more general computer science degree but nonetheless indicate a high correlation propensity for cross-reaching KSA development. High course overlap allows us to investigate the contexts that these programs create and are within in search of transferable lessons on curriculum development.

B. Engineering Programs

We first note the generally higher rates of course overlap in figure 2 between the engineering degrees compared to the computing degrees. This could likely be caused by implementing general engineering courses that students must take before matriculating into their chosen engineering field. It is unlikely that this high overlap rate is caused by general education requirements because the same general education requirements would apply to computing programs that do not exhibit this property.

Exceptionally high rates of courses are shared between computer engineering and electrical engineering in figure 2. When combining this with the high rate of course overlap between computer engineering and computer science in figure 1, we can infer that there is a shared set of KSAs between computer science, computer engineering, and electrical engineering represented through the shared courses.

Notably, software engineering shares very little similarity with other engineering programs, even less than biomedical or agricultural engineering, which have (comparatively) low rates of conferred degrees. This could be due to software engineering having fewer shared KSAs than other engineering fields, differences in software engineering classification in the universities, or an unknown characteristic of the specific departments represented in MIDFIELD.

C. General STEM Programs

Finally, we present a selection of STEM degrees indicated to be general through their name from the National Center for Education Statistics (e.g., ”Mathematics, General”) in figure 3. Overall, we note that the course overlap between these

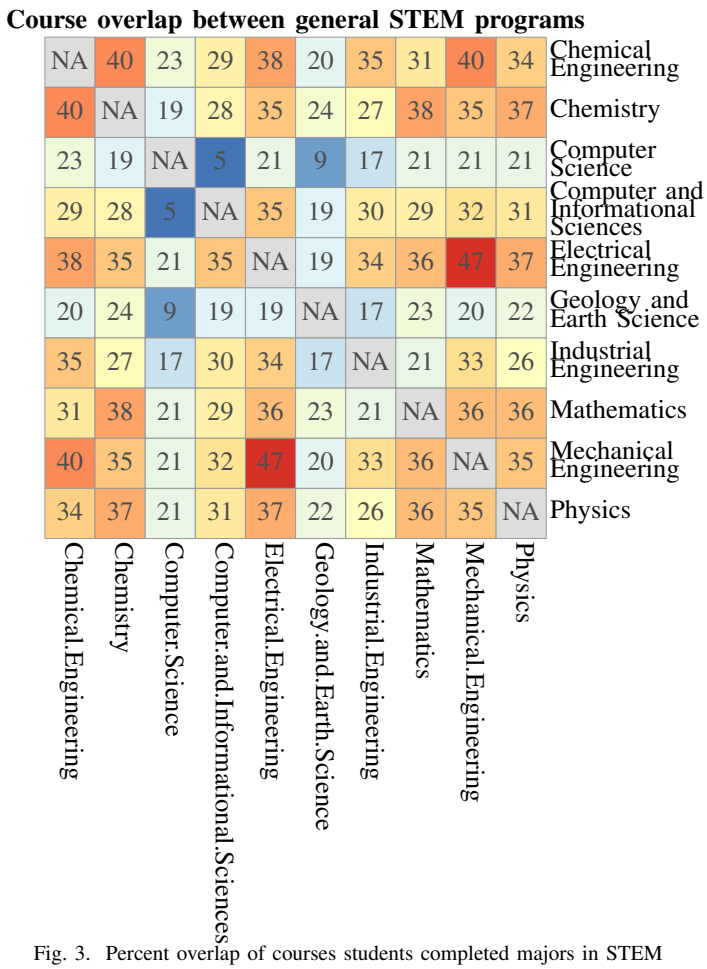

Fig. 3. Percent overlap of courses students completed majors in STEM

programs is between the computing and engineering domains. Computer science has an unexpectedly low course overlap with the other STEM fields, and computer and informational sciences have less deviation of overlap from the other STEM fields. This could indicate an underlying difference between computer science and computer and informational science that is preventing full convergence between computing and other domains. Correlations between engineering domains in this heatmap maintain the generally higher trend among engineering but show that the trend only selectively applies to other domain fields. For example, physics programs have high rates of course overlap with electrical and mechanical engineering, while these fields have significantly lower match rates with computer science. Variations across wide domains like this could indicate a possible university organizational difference facilitating convergent course enrollment or a significant variation in the availability of these programs at different universities.

IV. Future Directions

Our results identify that there is already significant course enrollment overlap between college majors. The outliers in these correlations point us to specific degree programs to compare in search of more in-depth explanations. Highly correlative degree programs indicate which specific programs are currently being integrated by either the curriculum designers themselves or through the efforts of the students navigating the systems.

These results further emphasize the feasibility of developing interdisciplinary majors. This proof of concept lays the groundwork for further investigation between majors. Identifying high cross-pollination courses between majors helps institutions identify which courses to create or advertise to develop interdisciplinary majors.

A greater granularity of analysis is possible through the MIDFIELD dataset as well. A more accurate heuristic to identify matching courses could be developed by integrating a time component (does the Programming 101 class develop the same KSAs in 2020 as in 2000?). Along with a better heuristic, more accurate results could be computed by considering the frequency of courses overlapping between programs. This would possibly be able to distinguish between common engineering and general education requirements.

References

[1] S. M. Lord, M. W. Ohland, M. K. Orr, R. A. Layton, R. A. Long, C. E. Brawner, H. Ebrahiminejad, B. A. Martin, G. D. Ricco, and L. Zahedi, "Midfield: A resource for longitudinal student record research," *IEEE Transactions on Education*, vol. 65, no. 3, pp. 245–256, 2022.

[2] J. T. Klein, *Interdisciplinarity: History, theory, and practice*. Wayne state university press, 1990.

[3] K. Marquardt, I. Wagner, and L. Happe, "Engaging girls in computer science: Do single-gender interdisciplinary classes help?," in *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 128–140, IEEE, 2023.

[4] P. O. Franco, J. S. Ramírez-Lugo, H. O. Zuazaga, M.-E. P. Hernández, L. R. Pericchi, and J. E. García, "Enhancing undergraduate education and curriculum through an interdisciplinary and quantitative initiative to broaden participation in big data," in *Proceedings of the 18th LACCEI International Multi-Conference for Engineering, Education, and Technology: Engineering, Integration, And Alliances for Ai Sustainable Development*, 2020.

[5] J. Cooper and L. Dierker, "Increasing exposure to programming: A comparison of demographic characteristics of students enrolled in introductory computer science programming courses vs. a multidisciplinary data analysis course," *International Research in Higher Education*, vol. 2, no. 1, 2017.

[6] P. J. Denning and M. Tedre, *Computational thinking*. MIT Press, 2019.

[7] J. Hlavac, "Knowledge, skills and abilities (ksas) as a metric to re-conceptualise aptitude: a multi-stakeholder perspective," *The Interpreter and Translator Trainer*, vol. 17, no. 1, pp. 29–53, 2023.

[8] M. J. Stevens and M. A. Campion, "The knowledge, skill, and ability requirements for teamwork: Implications for human resource management," *Journal of management*, vol. 20, no. 2, pp. 503–530, 1994.

[9] T. Chakraborty, "Role of interdisciplinarity in computer sciences: quantification, impact and life trajectory," *Scientometrics*, vol. 114, pp. 1011–1029, 2018.

[10] "NSF 10 Big Ideas - Special Report — NSF - National Science Foundation — nsf.gov." https://www.nsf.gov/news/special_reports/big_ideas/convergent.jsp. [Accessed 14-05-2024].

[11] "Learn About Convergence Research — new.nsf.gov." https://new.nsf.gov/funding/learn/research-types/learn-about-convergence-research. [Accessed 14-05-2024].