2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #36929

Exploring the Viability of Agent-Based Modeling to Extend Qualitative Research: Comparison of Computational Platforms

Samantha Splendido, Pennsylvania State University, University Park

Sam Splendido is a Ph.D. student in Mechanical Engineering at Pennsylvania State University. She is currently a graduate research assistant under Dr. Catherine Berdanier in the Engineering Cognitive Research Laboratory (ECRL). She earned her B.S. in Biomedical and Mechanical Engineering from Pennsylvania State University.

Catherine G. P. Berdanier, Pennsylvania State University

Catherine G.P. Berdanier is an Assistant Professor in the Department of Mechanical Engineering at Pennsylvania State University. She earned her B.S. in Chemistry from The University of South Dakota, her M.S. in Aeronautical and Astronautical Engineering and her PhD in Engineering Education from Purdue University. Her research expertise lies in characterizing graduate-level attrition, persistence, and career trajectories; engineering writing and communication; and methodological development.

Exploring the Viability of Agent-Based Modeling to Extend Qualitative Research: Comparison of Computational Platforms

Abstract

The purpose of this methods paper is to identify the opportunities and applications of agent-based modeling (ABM) methods to interpretative qualitative and educational research domains. The context we explore in this paper considers graduate engineering attrition, which has been a funded research focus of our group for ten years. In attrition research, as with all human research, it is impossible and unethical to imperil real graduate students by subjecting them to acute stressors that are known to contribute to attrition in order to "test" different combinations of factors on persistence and attrition. However, agent-based modeling (ABM) methods have been applied in other human decision-making contexts in which a computer applies researcher-programmed logic to digital actors, invoking them to make digital decisions that mimic human decision-making. From our research team's ten years of research studying graduate socialization and attrition and informed from a host of theories that have been used in literature to investigate doctoral attrition, this paper compares the utility of two programming languages, Python and NetLogo, in conducting agent-based modeling to model graduate attrition as a platform. In this work we show that both platforms can be used to simulate attrition and persistence scenarios for thousands of digital agent-students simultaneously to produce results that agree with both with previous qualitative data and that agree with aggregate attrition and persistence statistics from literature. The two languages differ in their integrated development environments (IDE) with the methods of producing the models customizable to fit the needs of the study. Additionally, the size of the intended agent pool impacted the efficiency of the data collection. As computational methods can transform educational research, this work provides both a proof-of-concept and recommendations for other researchers considering employing these methods with these and similar platforms. Ultimately, while there are many programming languages that can perform agent-based modeling tasks, researchers are responsible for translating high quality, theory-driven, interpretive research into a computational model that can model human decision-making processes.

Introduction and Literature Review

Many fields of research rely on the growth of technology to improve their research capabilities and further their findings within studies. Qualitative research fields, specifically, have benefitted from growing technology, especially relating data collection (e.g. audio/visual recordings, transcription services) and analysis (e.g. statistical software packages, word processing technology) [1]. However, an area that is underutilized by qualitative researchers is artificial intelligence (AI). AI and its sub-fields present a space for qualitative researchers to build upon existing research to enhance future studies through computational methods and modeling. In this paper we will focus on the potential for agent-based modeling (ABM), one such sub-field of AI, to contribute to qualitative research.

AI, as a general field, has a wide range of definitions, but the agreed upon goal is to develop computer programming to perform tasks, make decisions, and understand or perceive situations as a human would [2]. Considering AI as an umbrella, shown in Figure 1, its breadth of focus encompasses multiple subfields that each attempt to understand human-related phenomena through computers. One important sub-field of AI is a computer's ability to enhance its intelligence by learning from previous decisions, actions, consequences, etc. through machine learning (ML) [3]. As indicated by Figure 1, ML is a direct sub-field of AI that most closely aids in developing intelligence through its iterative processes. ML can be a helpful addition to many simulations as a way to consistently update programming, data, or output components and learn from previous iterations to improve accuracy and consistency. Another branch of AI with the potential to expand research of human-related phenomena is natural language processing (NLP). The goal of NLP is to identify and explain nuances within language, with a large expansion into computational methods [4]. With any experiences attempting to identify and explain nuances, NLP can be difficult to perform computationally but could revolutionize research based in interview data or other conversation or text-based qualitative data.

The final AI subfield in Figure 1, and the focus of this paper, is agent-based modeling (ABM). The goal of ABM is to develop a model that can reproduce dynamic phenomena to understand interactions and outcomes from those phenomena, especially those with some time dependence [5]. The information provided to the model is typically from previous research, but also can be expanded using some other AI methods like ML and NLP. The draw towards AI and its subsequent techniques in modern research is the ability to reproduce and create data to understand phenomena, especially in many situations where real-world research cannot be carried out due to various constraints.

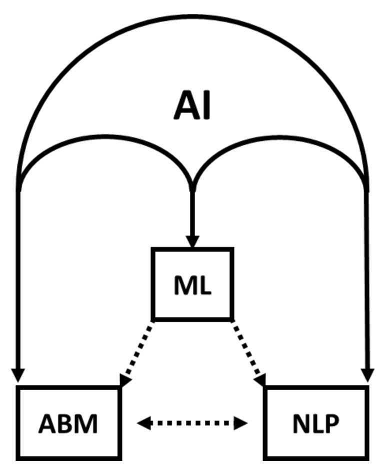

Figure 1: Visualization of umbrella of AI and sub-components' relationships.

ABM is used in a wide range of fields [6]–[11] to predict and study the decision-making of individuals, referred to as agents, within a computational model. An important component of agents is their set of rules that allow them to make decisions based on the situation they are in. These rules are tailored to the phenomena being studied so that the agent in the simulation performs actions similar to an individual outside of the simulation. For models to produce high quality results, the agent's rules are informed by previous thorough data collection, research, and interpretation of findings into rules for the algorithm [12]. The novelty of using ABM in research is that these agents are capable of interacting with both their environments and other agents, depending on the phenomena being simulated. This allows for researchers to tailor the rules to their information on phenomena and continue learning about interactions and situations between individuals in a system.

The growth of computation and simulations began in the 1950's and 60's in technical sectors of natural science research like physics and chemistry intending to perform large-scale predictive computations [13]–[15]. Beyond the natural sciences, computational techniques moved into economics to perform large scale calculations and provide large new data storage options[16]. Many of these computational techniques are the precursors to how we use ABM today to approach problems. ABM has, more recently, found success and been proven reliable in simulations for transmission and event prediction in medicine [11] and immunology [10]. These studies have produced results consistent with previous quantitative data due to the agent's data informed rules that govern their interactions with both their environment and other agents. Studies in immunology, for example, use ABM to model the spread of a disease within a population by programming different rates of the disease (like transmission rate, movement speed, cell death rate, etc.) with components for random occurrences for the disease within the human body[10]. ABM and ML also grown more prominent in social science settings focused on human decision-making, specifically in situations with large amounts of quantitative data [17]–[19].

There are not many applications of ABM, or AI in general, in qualitative studies. This paucity is because ABM typically relies on large amounts of quantitative data. In contrast, qualitative studies do not always provide the necessary scale of data, or quantification of that qualitative data, required for predictive modeling [12]. Applying ABM to qualitative research is challenging because there is a lack of rigorous scholarly guidance governing the process of applying quantitative methods to qualitative research while still maintaining a solid theoretical foundation and adhering to quality in the research process. It is also difficult to characterize the accuracy of the conversion from qualitative to quantitative data, since that process is highly interpretive. There are some qualitative studies that have developed techniques to convert data into quantitative forms. Elsawah et al. [20], for example, created a qualitative methodology to use ABM to model human decision-making based on perceptions of members within socio-ecological systems. Their research studied a farmer's perceptions of market impacts on their vineyards and how those perceptions impacted their decision-making. Their model also accounted for each farmer's cultivation strategy, attributes (how much experience they have, willingness to experiment), and budget in order to understand what contributed to decisions relating to running their vineyard. Outside of this example, there are very few studies that indicate the feasibility of using ABM to model human decision-making.

To this end, the purpose of this paper is to explore and show proof of concept that ABM to qualitative research topics, addressing viability within a qualitative field like engineering education. Many fields of study with meaningful high quality qualitative data could greatly benefit from models that can make sense of and recreate human decision-making. In this paper, we identify a potential application for ABM in qualitative work by understanding experiences and factors that can lead graduate engineering students to depart from their graduate programs.

Background in Graduate Attrition in Engineering Education: Motivating the Context for ABM

Attrition from engineering graduate programs has become an important issue for universities to try to understand. This is not to say that all attrition from graduate school is bad, but understanding the reasons for attrition can help universities better support their students and create environments more conducive to higher learning. Berdanier et al. [21] created the Graduate Attrition Decisions (GrAD) theoretical model, Figure 2, which identified major themes contributing to graduate engineering student attrition. These themes are the graduate student's advisor, support network, quality of life and work, cost, perception by others, and goals. Major findings from this work indicate the importance of recognizing that factors for attrition are not isolated and most choices to depart from graduate programs are due to combinations of factors. In subsequent work, our research team has continued to investigate the mechanisms of graduate engineering attrition, especially related to the costs incurred in graduate school, the relationship between themes, and how goals evolve over time in graduate programs, and the psychological costs incurred by graduate school. Many students who consider departure continue to persist because of the sunk-cost fallacy while in graduate school. The sunk cost fallacy for graduate students occurs when students consider departing their graduate program but choose to remain longer than they should because they feel they have invested too much time and effort to abandon the program at that point, which can contribute to challenges in predicting individual students' intentions to persist or depart [22].

Our research group is uniquely qualified to undertake agent-based modeling in an engineering education context because of our deep topic expertise and methodological commitment to constructivist and person-centered research. In sum, our research team has been working on aspects of doctoral engineering persistence, attrition, and thriving for the past ten years, and have conducted and analyzed over 100 interviews with current and former graduate students from nationwide samples, collecting experiences from those who are persisting, who thrive, who are considering leaving, and those who left their PhD programs either with or without a Master's degree. This expertise and experience, combined with high quality data sets and a deep commitment to thorough constructivist qualitative research methods, provides us the unique expertise to begin to translate these datasets into a computational model.

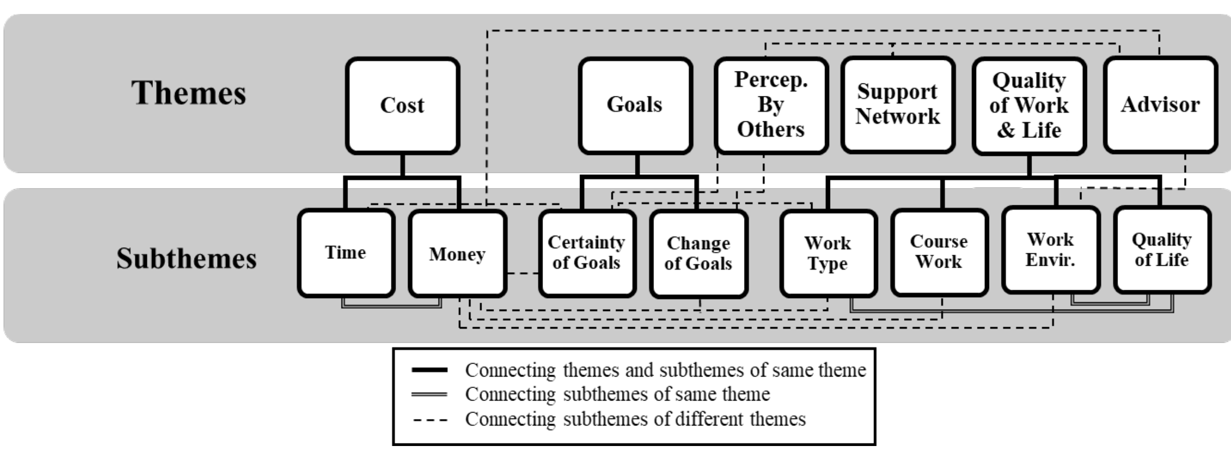

Figure 2: The GrAD[21] *themes and sub-theme visualization*

Understanding that attrition is nuanced and individualized to each graduate student, it is difficult to try to predict what real-world experiences may influence students to decide to depart from their programs. Also, the practices for testing which experiences, or combination of experiences, will cause students to depart would be highly unethical to test on human subjects. Thus, ABM presents a potential solution to studying attrition in graduate engineering students by simulating students' experiences through computer agents. These agents can be programmed with the attributes of graduate students (e.g. when they would consider departure, when they would choose to depart) and put through various graduate school experiences. Based on these attributes programmed agents can respond to positive or negative examples of the attrition themes identified with the same logic as humans and provide researchers greater insight into different attrition phenomena. ABM also allows for a larger scale of students to be studied in a shorter amount of time as well as preventing any negative ethical ramifications on human subjects.

In this paper, we use two different programming languages to apply ABM to qualitative-focused research data to demonstrate the efficacy of ABM in qualitative research. To achieve this goal, we aim to create agent-based models for attrition and compare programming languages and their ability to represent students' decision-making while experiencing graduate school. Additionally, this study allows us to understand and indicate the importance of quality and good practices for developing programs for ABM purposes as well as reducing the gap in qualitative research using simulations and modeling.

Methods

Selecting programming languages. An important aspect of working with different programming languages is identifying the goals of your work and finding the program that is able to achieve those goals. With the multitude of languages available, it can be difficult to understand what built-in components are right for your research goals. In this study, we explore how two programming languages (Python and NetLogo) can apply ABM to understand engineering graduate student attrition. Both programming languages have been used extensively in ABM contexts [10], [23]–[26] and are open-source with vast, free libraries of assistance for coding questions, concerns, and error troubleshooting. The main difference between the two languages is that Python is a general-purpose language while NetLogo is domain-specific. General-purpose programming languages are designed to be multipurpose (e.g. used in app development, web design, statistical calculations) and typically have optional packages that expand the program's abilities beyond the base level [27]. In contrast, domain-specific programming languages are developed for a specific field or purpose, with some optional packages that typically enhance the program's ability to perform its original task [28]. These programming languages, therefore, have characteristically different purposes and applications.

As a program, Python is capable of a wide variety of services and can be applied to a large number of interested domains. The services provided in Python can be uniquely utilized by each domain through data analysis and visualization, automation, web development, and AI and machine-learning. Additionally, Python is considered one of the most user-friendly and beginner friendly programming languages with an efficient and easy to read syntax. The syntax provides the added benefit to programs of fewer lines of code needed to execute a task leading to very efficient execution time [29].

NetLogo, on the other hand, was specifically designed for ABM applications. Many of the key syntax and features of NetLogo are similar to other programming languages, but they are specifically applied to agents within the program. Originally developed at Northwestern University, NetLogo creates an environment and agents tailored to specific parameters designated by the programmer and includes information about interactions between agents and the environment and agents with other agents [25]. Unlike other programming languages, NetLogo visualizes the program output for the agents as it makes its calculations through the integrated development environment (IDE). The IDE creates a window for the environment of interest and the programmer is expected to develop the model visually at the same time as developing the necessary calculations in the window.

Translating theory to Agent-Based Modeling algorithms. To understand the unique differences and advantages of each programming language in a qualitative context, we applied ABM to data from research in graduate engineering student attrition. The agent-based models developed for this paper are part of a larger study understanding graduate attrition. Because this work is still underway and represents an ongoing competitive research thrust, and because the purpose of this paper is to be a proof-of-concept on viability of these methods, the full code in each programming language will not be provided.

Quantifying qualitative data to apply ABM was a difficult task in this study. Understanding that all graduate students' experiences differ in their educational journeys is part of what makes those experiences so difficult to study. In an attempt to understand the changing landscape and difficulties of engineering graduate school, we referred to previous research [21], [30], [31]on graduate student attrition to develop the set of rules our student-agents follow. Transforming previous research information from qualitative data into a programmable quantitative format required multiple steps. First, we used literature to identify the main variables, in this case experiences that can cause attrition from graduate school, that can affect the ABM. Each existing theme in the GrAD model [21]was transferred as a potential variable that can be experienced by each agent and helped inform us of how to represent the variable's weighted effect on the agent's decisions. Further weighting choices are discussed in our implementation section. These weights were additionally informed by interview data collected from with departing graduate students, which allowed us to determine the variables that could have a greater impact on students' decisions to persist or depart from their graduate engineering programs. Some of the GrAD model themes were separated into multiple variables to account for more situations and accommodate the subthemes in the GrAD model. For example, the GrAD model cost theme was separated into a funding variable, for financial costs of graduate school, and an academic program variable, to account for time commitments and expectations that add to non-financial costs of graduate school, in the model. Other themes in the GrAD model were grouped into the same variable to account for the linkages between themes and subthemes. For example, the "perception by others" and "support network" themes both contribute to the theme of school community and indicate the important groups that both judge and support students in graduate school. The interview data also provided additional insights into important factors of graduate school that were not as heavily focused on in the GrAD model, but that were investigated in subsequent work [22]. As such, our model includes a variable for critical events [31] with the potential to completely change a student's career path as well as a modifier to represent sunk cost after each year the student-agent completes.

Implementation of ABM for graduate student experiences. Our method of following each student-agent through their graduate school journey provides each student-agent with a motivation that can be affected by each of these variables. Then, the student-agent moves through their academic year by experiencing this variable. Each variable outputs a randomized value based on their weighted range that can add or subtract from the student-agent's randomly assigned motivation value. These ranges were informed by graduate attrition literature to better represent the magnitude of each variable. In both the Python and NetLogo models, we utilized the "random" function to maximize the variability of each student-agent's experience. This means that students' interactions with each variable was randomly assigned instead of us as researchers assigning experiences to each agent. Randomizing the variables allows researchers to easily customize the number of agents and creates more opportunities for a variety of experiences to be represented.

The model initiates with each student-agent beginning their first academic year with a randomly prescribed motivation value between 10 and 25 that indicates their desire to go to graduate school. This is used to represent observed varying levels of interest in graduate school that students experience and sets a baseline of motivation for the students throughout the academic years. From there, the variables change the motivation of the student-agents, both positively and negatively, based on the randomly generated value. For example, in the first academic year a student can have a bad advisor experience, but develop a good support network, so their motivation value can decrease because of the advisor variable and then increase because of the community or support variables. After each academic year, the model checks the student-agent's motivation values to see if that student-agent will persist or depart from graduate school that year. If a student-agent has a motivation value below -5 at the end of any year, our model assumes that the student-agent leaves the program. As an example of how these decisions manifest: We chose a value of -5 to allow student-agents to experience some difficulty (have a negative motivation value) but choose to "stick it out", a testament to a surviving versus thriving experience in graduate school [30]. Beyond -5, we assume student-agents have reached a psychological threshold and will want to depart, aligning with the trends we have seen in our qualitative data.

For student-agents that do not leave their program after an academic year, we add a random value between 5 and 15 to their motivation to simulate further motivation from the previous year as well as simulate the impact of sunk cost, another aspect of graduate student motivation that has been featured in our prior qualitative work. Then, the motivation value at the end of each academic year becomes the starting motivation value of the following academic year. Therefore, if a student-agent has a motivation of -5 at the end of a year, their sunk cost value will bring them at least back up to 0. Because our models assume a student will complete graduate school within 6 years (an approximation of the US average completion time via the National Center for Science and Engineering Statistics [32]) a student-agent with a motivation value greater than -5 after 6 academic years is assumed to have graduated from their program. However, it is possible for student-agents to drop out of their programs in each academic year, including their last year because literature indicates graduate engineering students consider departing their degree program at many different points in their graduate school journey, including their last year [30].

To properly characterize the impacts of our variables, we developed our ranges of modifier values to reflect on the weight of each factor shown in literature on student experiences [21]. The ranges were developed to reflect the weight of each variable on the students. A larger range with larger values indicates a greater weight of that variable and potential for that variable to impact motivation more than others. One example is that literature indicates that one of the most impactful variables that impacts a graduate student's experience and well-being is the student-advisor relationship [21], [31]. Specifically, we understand that a difficult advisor relationship can severely impact a student's intention to depart from their program. This understanding informed our advisor variable range values to go from -20 to 5, indicating that a bad advisor has the potential to derail a student's graduate school experience, while a good advisor has the potential to improve a student's experience, but not nearly as significantly as a bad advisor. This decision helps indicate the weight of this variable by indicating that there is a much larger opportunity for a negative advisor experience to impact the student, even if they have other positive experiences.

Randomizing the student-agents' experiences with each variable is important for ABM because it allows for these experiences to fluctuate from positive to negative. For example, a good advisor in one academic year has the potential to become a "bad advisor" the next year: Testimonies from our interviews and literature [30], [31] indicate that many students had positive experiences with their advisors, until something changed within the advisor's personal or professional life (becoming department head, children, change in relationship with student, etc.) which led students to feel neglected or uncomfortable in their situation. The opposite is also true, but less likely, where an advisor relationship shifts from negative to positive.

Another example of the impact of our variable ranges is with the graduate school program variable. A sample of the initial motivation value (programmed as desire) and advisor variable Python code are shown in Figure 3. We chose a range from -10 to 10 for this variable because literature [33] indicates that good and bad graduate school programs can potentially have very similar positive and negative, respectively, impact students' motivations. The school program range in the model also indicates the graduate school's ability to have a somewhat significant impact on the student's experience but will most likely not be the sole reason for the student remaining or departing from their school program.

```python
# did they want to go to grad school
def wantgrad(self):

    desire = random.randrange(10, 25)
    #if desire <= 17:
    #   print("Did not have strong feelings about going to grad school " + str(desire))
    #else:
    #   print("Did want to go to grad school " + str(desire))

    return desire


# Year run
def yearrun(self, player, DO, year):
    player += random.randrange(5, 15)
    year += 1
    #print("year: " + str(year) + " Player: " + str(player) )

    # Advisor
    playeradv = random.randrange(-20, 5)
    playerad = 0
    # print("Player adviser num: " + str(playeradv))
    #if playeradv<0:
    #     print("Bad Advisor")
    #else:
    #     print("Good Advisor")
    self.tempplayerad = playeradv + playerad
    player = player + playeradv + playerad
    player, DO, year = self.playercheck(player, year, DO, "result: Advisor")
```

Figure 3: Sample portion of Python code for attrition model for "desire to enroll" and "advisor."

Parameters for each language to run simulations. The main variable information, the ranges for the variables, and the significance of each variable to a student's experience were developed first. This information was used in both programming languages to compare how they performed ABM. In Python, we programmed calculations to output graduation rates and the percentage of student-agents that departed due to each variable. While Python does not have embedded visualization aspects showing real-time progress, there are exportable data components that can be plotted in Excel or otherwise to showcase individual agent-student stories. The NetLogo IDE requires the use of their window to visualize agents in the study, so we programmed our student-agents as people and allowed them to change color based on their motivation within graduate school. In NetLogo, all student-agents begin as green icons and only change color if their motivation drops below 0. At 0, we assume the students are experiencing some difficulty but may not be considering departure just yet, which changes their color to yellow. The student-agent icons change to a red color when their motivation value is below 0, indicating that they are considering departing from their program. At the end of each academic year, NetLogo checks whether the student-agent's motivation value is equal to or below -5, in which case the model assumes the student is departing from their program. If this happens, the student-agent icon becomes blue. These color changes allow the model to calculate the graduation rate within the graduate program. Both programming languages began their modeling with 10,000 random student-agents in their first year of graduate school and determined the graduation rate for each after 6 academic years to compare their performance for large sample sizes to each other.

Results

Through our development of an agent-based model for graduate attrition in two programming languages, we found the unexpected benefit of result confirmation through comparison of the two models, especially in their outcomes. The example outputs from the two employed programming languages are shown in Figure 4. The results from the each of the platforms agreed well with each other and with the real statistics on graduate engineering attrition: The Python model currently generates a graduation rate for agent-students of 57.45% and NetLogo of 59.68%, were similar to one another and comparable to the engineering graduation rates after 6 years according to The Council of Graduate Schools (57%) [34].

More promising still is that repeated runs of the ABM simulations in each programming language produced graduation rates in the low ~60% every time, even with large sample sizes. The fact that the models developed through the methods discussed in this paper produce accurate results even with much of the model randomizing student-agents' experiences is a testament to the validity of this method.

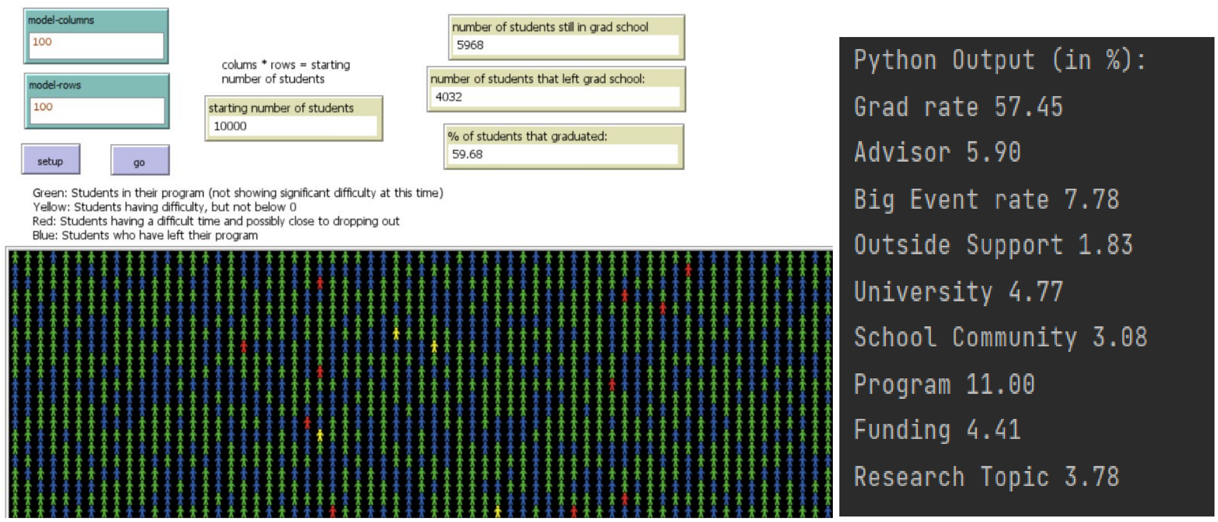

Figure 4: NetLogo (left) and Python (right) sample model outputs.

The two platforms described in this study both validate the ability of ABM to begin to describe decision making in engineering education and offers an understanding of the affordances of the two platforms that may be advantageous depending on the research applications, especially for ABM purposes. Through our application of the general-purpose and domain-specific programming languages, we found benefits to both for model development and execution. General-purpose programming languages, like the Python model we developed, tend to have more developed internal computation methods and are capable of running larger data sizes much faster. It is important to note, however, that computation time depends on a variety of factors including number of lines of code and printed outputs, so the speed with which general-purpose programs create models can be highly individualized based on inputs in a given study. The Python model is also beneficial because it provides clearer information on errors within the code, exporting data, and data analysis.

The NetLogo model offers different benefits. For example, the NetLogo model provided interesting visual information during and after the models were calculated. While it did take longer to run this model for large sample cases compared to the Python model, being able to visualize students staying, considering departure, and departing from the program as the model was running can provide a different depth to the model and serves as a reminder that the agents in this model are more than just numbers in a computer. Another benefit of NetLogo is that it contains more built-in functions to look at specific agents during the computation, giving researchers the opportunity to focus on an individual agent's experiences and decisions with much simpler syntax. However, determining what is the proper way to visualize the data from the model requires additional effort and time and may not be beneficial in all studies.

Discussion and Cautionary Notes on Using ABM to Describe Human Behavior

We do not assert that any one programming language is superior in applying ABM, but there are different affordances granted by each. Therefore, it is up to the researcher to understand the important differences within these programming languages to make an informed decision on which one will help them achieve their goals in the most effective, efficient, and ethical way. In future development of our research, some additional packages within programs may be utilized within both programming languages to potentially compare further modeling techniques. While it is important to select a programming language based on the goals of the study, any application of ABM can provide opportunities to develop adjustable and readily computable models when grounded in theoretical understandings.

While ABM provides a unique opportunity for studies to be conducted to understand phenomena and potentially predict human decision-making for a variety of experiences, the potential negative impacts of models predicting human decisions and outcomes cannot be overlooked. Many models must account for randomness components in human decision-making and, therefore, will not always provide the same statistically significant outputs. This knowledge allows us to recognize that human decision-making does not always follow predictable patterns. The model's randomness also reminds us that even though the student-agents in our study were created in this computer program, these models are intended to understand the experiences of real students. Therefore, researchers must remain committed to the individuals their study was intended to understand and their real-world experiences.

The necessity to weight different contributing variables in a model requires researchers to study human decision-making and lived experiences for the particular phenomena being studied well before the model can be developed and used. Studying the more unique "what if" scenarios of a phenomena can be tempting to some researchers, but those scenarios become unimpactful to the given community if not based in theory, literature, or experience. Attempting to weight information that is not well-rooted in theory or data can create outputs that will not accurately represent phenomena and will produce unhelpful or incorrect results that contribute to the saying "bad data in equals bad data out." It is especially important to base models on a theoretical understanding of the phenomena because ABM is just beginning to be considered *as a tool* in human-subjects research. These models should not be used as a means to see how much stress students or people in general can handle before they break or to completely replace real-life human studies. The goal of agent-based models in human-subjects research is, instead, to supplement understanding of human decisions in situations where sample sizes may be an issue or where true studies cannot be performed due to ethical concerns.

Some components of programming languages can pose ethical concerns to our community, especially relating to syntax, that should be addressed by studies intending to use ABM. These languages have many built-in functions that, when taken out of context, can raise questions about the programmer's research principles. For example, the NetLogo code used in this paper has many built-in functions and names for components or calculations that can be confusing or offensive when describing human-subjects related models. Agents in NetLogo models are created and identified as a certain "breed". This is meant in no way to dehumanize the agents in model; it is just the syntax used by the software developers who created NetLogo for biological models (e.g. predator-prey simulations, virus-cell interactions) [25]. Another example of potentially offensive syntax is that an agent that is deleted and recreated "dies" in the programming code. Out of context, one may assume that we are trying to play "god" with students' careers and lives or the experiences of agents in other contexts. This is not the case as the syntax cannot be changed by the researcher, so it is important for researchers to be transparent about their programming practices and indicate potential philosophical or ethical concerns upfront. Also, it remains important for researchers to remain steadfast in their commitment to interpretative research, commitments to qualitative and educational research to understand and helping the human subjects they are researching, and to remember the importance of not playing "god" when it comes to simulated experiences.

Conclusion

There is potential opportunity for agent-based modeling (ABM) to be incorporated into qualitative research. As AI and subsequent programming techniques continue to advance, developing human decision-making models from qualitative data can provide substantial support data and greater insights into many phenomena. With the large pool of information on modeling and programming languages available, researchers who are interested in expanding into ABM will need a clear understanding of their goals and a solid foundation for their model to be developed and used correctly. This includes understanding important functions, syntax, and capabilities of programming languages before the models are developed. Additionally, researchers will need significant understanding of relevant theories, literature, and real-world data related to the given phenomena to properly translate experiences into programmable functions, variables, loops, etc.

Through this comparison of computational platforms, we identified some differences between programming languages that may help researchers determine what programming language is most suitable when looking to utilize ABM techniques in their studies. In languages like Python that are part of the general-purpose category, computation time and interpretable data information and analysis techniques are great advantages. Languages like NetLogo which are domain-specific have more readily available visualization techniques that may provide more insights into phenomena for specific studies. General-purpose and domain-specific programming languages also have many similar qualities, like downloadable packages for further modeling techniques, that can be further explored by individual researchers. Applying both programming languages to develop a model for engineering graduate student attrition provided an additional quality assessment for the accuracy of ABM models. Both models were able to provide graduation rates similar to one another after multiple runs and comparable to current literature findings on attrition from engineering graduate school in qualitative studies.

Finally, to maintain the goals of human-subjects research, we again emphasize the importance of understanding the intentions of our research while developing the models. The purpose of developing agent-based models is not to test random scenarios or perform screenings on policy changes to see which students make it through a program. Instead, ABM can supplement thorough and well-designed qualitative research and can develop models that understand a given phenomenon without risking the lives or well-being of human subjects.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant #1844878. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the National Science Foundation.

References

[1] C. A. Moylan, A. S. Derr, and T. Lindhorst, "Increasingly mobile: How new technologies can enhance qualitative research", doi: 10.1177/1473325013516988.

[2] P. H. Winston, *Artificial Intelligence*, 3rd Edition. Addison-Wesley Longman Publishing Co., Inc., 1992.

[3] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science (1979)*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/SCIENCE.AAA8415/ASSET/AB2EF18A-576D-464D-B1B6-1301159EE29A/ASSETS/GRAPHIC/349_255_F5.JPEG.

[4] N. Ranjan, K. Mundada, K. Phaltane, and S. Ahmad, "A Survey on Techniques in NLP," *Int J Comput Appl*, vol. 134, no. 8, pp. 975–8887, 2016.

[5] C. M. Macal and M. J. North, "Agent-based modeling and simulation," *Proceedings - Winter Simulation Conference*, pp. 86–98, 2009, doi: 10.1109/WSC.2009.5429318.

[6] S. Bouarfa, H. A. Blom, R. Curran, and M. H. Everdij, "Agent-based modeling and simulation of emergent behavior in air transportation," *Complex Adaptive Systems Modeling*, vol. 1, no. 1, pp. 1–26, Dec. 2013, doi: 10.1186/2194-3206-1-15/FIGURES/10.

[7] E. Z. Berglund and M. Asce, "Using Agent-Based Modeling for Water Resources Planning and Management," *J Water Resour Plan Manag*, vol. 141, no. 11, p. 04015025, May 2015, doi: 10.1061/(ASCE)WR.1943-5452.0000544.

[8] Q. Huang, D. C. Parker, T. Filatova, and S. Sun, "A review of urban residential choice models using agent-based modeling," vol. 41, pp. 661–689, 2014, doi: 10.1068/b120043p.

[9] M. W. Macy and R. Willer, "FROM FACTORS TO ACTORS: Computational Sociology and Agent-Based Modeling," *Annu. Rev. Sociol*, vol. 28, pp. 143–66, 2002, doi: 10.1146/annurev.soc.28.110601.141117.

[10] F. Chiacchio, M. Pennisi, G. Russo, S. Motta, and F. Pappalardo, "Agent-based modeling of the immune system: NetLogo, a promising framework," *Biomed Res Int*, vol. 2014, 2014, doi: 10.1155/2014/907171.

[11] B. Ambale-Venkatesh *et al.*, "Cardiovascular Event Prediction by Machine Learning: The Multi-Ethnic Study of Atherosclerosis," *Circ Res*, vol. 121, no. 9, pp. 1092–1101, Oct. 2017, doi: 10.1161/CIRCRESAHA.117.311312/-/DC1.

[12] W. Rand and R. T. Rust, "Agent-based modeling in marketing: Guidelines for rigor," *International Journal of Research in Marketing*, vol. 28, no. 3, pp. 181–193, Sep. 2011, doi: 10.1016/J.IJRESMAR.2011.04.002.

[13] D. M. Auslander, "Distributed System Simulation With Bilateral Delay-Line Models," *Journal of Basic Engineering*, vol. 90, no. 2, pp. 195–200, Jun. 1968, doi: 10.1115/1.3605079.

[14] D. K. Lilly, "On the numerical simulation of buoyant convection," *Tellus*, vol. 14, no. 2, pp. 148–172, May 1962, doi: 10.1111/J.2153-3490.1962.TB00128.X.

[15] R. N. Foster and J. B. Butt, "A computational model for the structure of porous materials employed in catalysis," *AIChE Journal*, vol. 12, no. 1, pp. 180–185, Jan. 1966, doi: 10.1002/AIC.690120132.

[16] R. Agrawal, A. Borgida, and H. v. Jagadish, "Efficient management of transitive relationships in large data and knowledge bases," *ACM SIGMOD Record*, vol. 18, no. 2, pp. 253–262, Jun. 1989, doi: 10.1145/66926.66950.

[17] L. Hannaford, X. Cheng, and M. Kunes-Connell, "Predicting nursing baccalaureate program graduates using machine learning models: A quantitative research study," *Nurse Educ Today*, vol. 99, p. 104784, Apr. 2021, doi: 10.1016/J.NEDT.2021.104784.

[18] L. Tan, J. B. Main, and R. Darolia, "Using random forest analysis to identify student demographic and high school-level factors that predict college engineering major choice," *Journal of Engineering Education*, vol. 110, no. 3, pp. 572–593, Jul. 2021, doi: 10.1002/JEE.20393.

[19] J. Kleinberg *et al.*, "Human Decisions and Machine Predictions," *Q J Econ*, vol. 133, no. 1, pp. 237–293, Feb. 2018, doi: 10.1093/QJE/QJX032.

[20] S. Elsawah, J. H. A. Guillaume, T. Filatova, J. Rook, and A. J. Jakeman, "A methodology for eliciting, representing, and analysing stakeholder knowledge for decision making on complex socio-ecological systems: From cognitive maps to agent-based models," *J Environ Manage*, vol. 151, pp. 500–516, Mar. 2015, doi: 10.1016/J.JENVMAN.2014.11.028.

[21] C. G. P. Berdanier, C. Whitehair, A. Kirn, and D. Satterfield, "Analysis of social media forums to elicit narratives of graduate engineering student attrition," *Journal of Engineering Education*, vol. 109, no. 1, pp. 125–147, Jan. 2020, doi: 10.1002/JEE.20299.

[22] G. M. Sallai, M. Bahnson, K. Shanachilubwa, and C. G. P. Berdanier, "Persistence at What Cost? How Graduate Engineering Students Consider the Costs of Persistence within Attrition Considerations," *Journal of Engineering Education*, 2023, doi: 10.1002/jee.20528.

[23] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," *Proceedings - Winter Simulation Conference*, vol. 2005, pp. 2–15, 2005, doi: 10.1109/WSC.2005.1574234.

[24] C. Harwick, "Helipad: A Framework for Agent-Based Modeling in Python," *SSRN Electronic Journal*, Jun. 2021, doi: 10.2139/SSRN.3870501.

[25] S. Tisue and U. Wilensky, "NetLogo: Design and implementation of a multi-agent modeling environment," *Conference on Social Dynamics: Interaction*, 2004.

[26] J. Kazil, D. Masad, and A. Crooks, "Utilizing Python for Agent-Based Modeling: The Mesa Framework," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12268 LNCS, pp. 308–317, 2020, doi: 10.1007/978-3-030-61255-9_30/FIGURES/3.

[27] G. Berry, "Real time programming : special purpose or general purpose languages," 1989. Accessed: Feb. 06, 2023. [Online]. Available: https://hal.inria.fr/inria-00075494

[28] M. Mernik, J. H. Cwi, A. M. Sloane, J. H. Nl, and ; A M Sloane, "When and How to Develop Domain-Specific Languages When and How to Develop Domain-Specific Languages 317," *ACM Comput Surv*, vol. 37, no. 4, pp. 316–344, 2005.

[29] K. R. Srinath, "Python-The Fastest Growing Programming Language," *International Research Journal of Engineering and Technology*, 2017, Accessed: Feb. 06, 2023. [Online]. Available: www.irjet.net

[30] E. Zerbe, G. M. Sallai, and C. G. P. Berdanier, "Surviving, thriving, departing, and the hidden competencies of engineering graduate school," *Journal of Engineering Education*, 2022, doi: 10.1002/JEE.20498.

[31] E. Zerbe, G. M. Sallai, K. Shanachilubwa, and C. G. P. Berdanier, "Engineering graduate students' critical events as catalysts of attrition," *Journal of Engineering Education*, vol. 111, no. 4, pp. 868–888, Oct. 2022, doi: 10.1002/JEE.20481.

[32] National Center for Science and Engineering Statistics, "Doctorate Recipients from U.S. Universities 2021," 2021. [Online]. Available: https://ncsesdata.nsf.gov/rdas

[33] C. M. Golde, "The Role of the Department and Discipline in Doctoral Student Attrition: Lessons from Four Departments," *J Higher Educ*, vol. 76, no. 6, pp. 669–700, 2016, doi: 10.1080/00221546.2005.11772304.

[34] Council of Graduate Schools, "Ph.D. completion and attrition: Analysis of baseline data," *Council of Graduate Schools*, 2008.