The Journal of Computing Sciences in Colleges

Papers of the 30th Annual CCSC Midwestern Conference

October 20th-21st, 2023
University of Indianapolis
Indianapolis, IN

Baochuan Lu, Editor
Southwest Baptist University

Saleh Alnaeli, Regional Editor
University of Wisconsin-Stout

Volume 39, Number 4 **October 2023**

The Journal of Computing Sciences in Colleges (ISSN 1937-4771 print, 1937-4763 digital) is published at least six times per year and constitutes the refereed papers of regional conferences sponsored by the Consortium for Computing Sciences in Colleges.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

Following is a listing of the contact information for the members of the Board of Directors and the Officers of the Consortium for Computing Sciences in Colleges (along with the years of expiration of their terms), as well as members serving CCSC:

Serving the CCSC: These members are serving in positions as indicated:

Bin Peng, Associate Editor, bin.peng@park.edu, Department of Computer Science and Information Systems, Park University, Parkville, MO 64152.

Brian Hare, Associate Treasurer & UPE Liaison, hareb@umkc.edu, School of Computing & Engineering, University of Missouri-Kansas City, Kansas City, MO 64110.

George Dimitoglou, Comptroller, dimitoglou@hood.edu, Department of Computer Science, Hood College, Frederick, MD 21701.

Megan Thomas, Membership System Administrator, mthomas@cs.csustan.edu, Department of Computer Science, California State University Stanislaus, Turlock, CA 95382.

Karina Assiter, National Partners Chair, karinaassiter@landmark.edu, Landmark College, Putney, VT 05346.

Deborah Hwang, Webmaster, hwangdjh@acm.org.

CCSC National Partners

The Consortium is very happy to have the following as National Partners. If you have the opportunity please thank them for their support of computing in teaching institutions. As National Partners they are invited to participate in our regional conferences. Visit with their representatives there.

Gold Level Partner

Rephactor
ACM2Y
ACM CCECC

Welcome to the 2023 CCSC Midwestern Conference

On behalf of the conference committee, I welcome you to the 30th annual Consortium for Computer Science Conference (CCSC) for the Midwest region, held on the campus of the University of Indianapolis, in Indianapolis, Indiana. As always, we are looking forward to another great conference with something to offer everyone, regardless of whether you are a returning or new attendee, or faculty or a student. The conference program includes refereed paper sessions, panels, speakers, workshops, tutorials, nifty assignments, and student activities. The submitted papers were of high quality and the acceptance rate was 64%. All accepted papers will be presented and published in the Journal of Computing Sciences in Colleges. We will also be hosting a programming contest for our undergraduate students as well as different activities focused on students including the Student Showcase (which includes undergraduate student research projects and creative works), and a popular panel session titled "What Students Need to Know About Industry."

We also have a pre-conference workshop planned about no-code development, and have exciting panels and tutorial sessions. We are also honored to have Mr. Jonathan Sweeny as our keynote speaker. Mr. Sweeny is with the FBI, and will be speaking on "Cybersecurity Arms Race." Emily Kitterman with TechPoint will deliver our dinner address on "Opportunities for Talent in Indiana's Digital Innovation Economy: A Careers in Tech Presentation."

I truly appreciate the many people who make conferences like this one possible. All these people worked hard behind the scenes for a year to bring this conference to life. First, a big thank you to the members of the Conference Committee for their hard work and dedication in making this conference the worthwhile event it is. In particular, I thank the Site Chair, Paul Talaga, for ensuring that everything is in place for the conference at the University of Indianapolis. I also thank those who served as peer reviewers during the paper review process. We greatly appreciate their careful attention to detail, and their thoughtful and candid critique of each paper. The authors who submitted papers and other works for consideration also deserve a big thanks. Without colleagues willing to share their work, there would be no reason to hold this conference. I strongly encourage everyone to consider submitting a paper, panel, tutorial, or workshop for next year's conference, and bring your students along as well! We sincerely thank Rephactor and ACM2Y/ACM CCECC for their support of CCSC as National Partners.

We hope you enjoy the conference and will tell us how we can make it better. We are always looking for new people to join the Conference Committee, and sites to host future conferences. Please contact any committee member for more information. The 2024 Conference Committee will meet on Saturday immediately after the conference lunch to start planning for next year. Come join us!

Thank you for being part of the 30th CCSC: MW annual conference.

David L. Largent
Ball State University
CCSC Midwest Regional Representative

2023 Steering Committee - CCSC Midwestern Region

Imad Al Saeed, Registrar (2022) ... Saint Xavier University, Orland Park, IL
Saleh Alnaeli, Editor (2021) . University of Wisconsin-Stout, Menomonie, WI
Stefan Brandle, Webmaster (2023) .... Taylor University, Upland, IN
Mary Jo Geise, Treasurer (2023) .... University of Findlay, Findlay, OH
Sean Joyce, At-Large (2022) .... Heidelberg University, Tiffin, OH
Kris Roberts, At-Large (2021) Ivy Tech Community College, Fort Wayne, IN
David Largent, Regional Representative (2022) Ball State University, Muncie, IN
David Largent, Conference Chair (2023) ... Ball State University, Muncie, IN
Saleh Alnaeli, Past Conference Chair .... University of Wisconsin-Stout, Menomonie, WI

2023 CCSC Midwestern Conference Committee

Lucy La Hurreau, Conference Chair .... Ivy Tech Community College, IL
David Largent, Conference Co-Chair ...... Ball State University, Muncie, IN
Zaid Althaha, Vice-Chair ..... University of Wisconsin-Parkside, Somers, WI
Paul Talaga, Site Chair .... University of Indianapolis, Indianapolis, IN
Saleh Alnaeli, Authors .... University of Wisconsin-Stout, Menomonie, WI
Ahmed Elmagrous, WIP, Nifty Tools/Assignments .... University of Wisconsin-Stout, Menomonie, WI
Cathy Bareiss, Panels, Tutorials, Workshops .. Bethel University, Mishawaka, IN
Imad Al Saeed, Papers .... Saint Xavier University, Orland Park, IL
Grace Mirsky, Past Conference Chair .... Benedictine University, Lisle, IL
Paul Talaga, Programming Contest Co-Chair ..... University of Indianapolis, Indianapolis, IN
Md Haque, Programming Contest Co-Chair ...... University of Indianapolis, Indianapolis, IN
David Largent, Publicity .... Ball State University, Muncie, IN
Deborah Hwang, Registrar .... University of Evansville, Evansville, IN
Stefan Brandle, Speakers Chair .... Taylor University, Upland, IN
Mary Jo Geise, Treasurer (2023) .... University of Findlay, Findlay, OH
Matt Green, Student Showcase Chair .. Waukesha County Technical College, Pewaukee, WI
Kris Roberts, Two-year College Liaison Co-Chair ...... Ivy Tech Community College, Fort Wayne, IN
Takako Soma, Vendors .... Illinois College, Jacksonville, IL
Stefan Brandle, Webmaster .... Taylor University, Upland, IN

Cybersecurity Arms Race*

Keynote

Jonny Sweeny
Indianapolis, IN

In the ever-evolving landscape that is the Internet, there is a constant arms race between the miscreants that pursue self interest whilst harming others, and those that enforce accountability for those actions. In addition to highlighting some of the interesting aspects of this arms race, Sweeny will share examples of these from recent FBI cases and forecast some of what lies ahead

Opportunities for Talent in Indiana's Digital Innovation Economy: A Careers in Tech Presentation*

Banquet Address

Emily Kitterman
Senior Manager of Careers at TechPoint

TechPoint is the industry-led growth initiative working to inclusively grow Indiana's digital innovation economy and overall tech ecosystem. Housed as an initiative of the Central Indiana Corporate Partnership, TechPoint is uniquely positioned as a catalytic convener of key stakeholders from across the Hoosier state in academics, industry, community organizations, and government. Together, we aim to achieve our mission by expanding the digital skill talent pipeline, enhance connectivity through storytelling, and elevate industry success and innovation. In order to expand Indiana's talent pipeline, TechPoint offers various programs, beginning with our Careers in Tech web platform. Careers in Tech outlines and demystifies Indiana's tech landscape using local data from industry partners and highlights areas for opportunity for employment seekers. With careers outlined into specific roles and possible pathways, we break down the skills needed, both technical and foundational; the necessary training and experience based on your point of entry and desired job moves; salary expectations; and highlight stand-out professionals currently in the given field. Careers in Tech is more than just an information resource, it is also a marketplace of the variety of education and training programs available to Hoosiers, including some free virtual learning experiences to try on different roles. Altogether, Careers in Tech serves as the doorway for prospective talent for Indiana's growing digital innovation economy.

Enhancing Mouse Dynamics for Continuous Secure Authentication Using Machine Learning*

Braxton Fair, Cole Harp, Mahlet Asmare
Mohammed Khalil, Rushit Dave, Mounika Vanamala
Computer Information Science
Minnesota State University, Mankato
Mankato, MN 56001
{Braxton.fair,cole.harp,mahlet.asmare}@uwec.edu
{mohammedahnaf.khalil,rushit.dav,vanamalm}@uwec.edu

Abstract

This paper explores the potential use of mouse dynamics as a continuous authentication method. Our study collects data from 15 participants playing the video game Roblox and analyzes 3 machine learning models where we perform a comparative analysis between them. Our results show promising accuracy rates of upwards of 97.51%, with small false negative rates as low as 1.32%, with k-nearest neighbors on average having the best performance. While further research is needed, our study suggests that mouse dynamics could be a useful tool in user authentication tasks. The novel contributions of our paper include providing a new mouse dynamics dataset and a comprehensive comparison of 3 machine learning models for continuous authentication.

1 Introduction

As online platforms and services grow, secure authentication is increasingly vital. Passwords and security questions can be vulnerable to cyber-attacks. Continuous authentication is another layer on top of passwords that verifies a user's identity throughout a session. It monitors behavior for signs of unusual activity or unauthorized access. Machine learning-based methods analyze user behavior patterns, identifying any anomalies indicating a security breach. These algorithms learn and adapt to new patterns in real-time, detecting threats before they cause damage. This prevents security incidents and data breaches. In this paper, we present a comparative study of machine learning-based continuous authentication methods. Our study is based on data collected from a popular online game, Roblox. We extracted features from the raw data and used them to train and evaluate our machine learning models. The models used in our study include K-Nearest Neighbors (k-NN), Decision Trees (DT), and Support Vector Classifier (SVC). The novel contributions of this paper are as follows:

- Introduce a novel mouse dynamics dataset containing data from 15 users while they played the video game in Roblox. Dataset is available at: `https://github.com/taxborn/mauth-research-project`.
- Develop secure authentication models using k-Nearest Neighbors, Decision Tree, and Support Vector Classifiers, and comparing the results from these classifiers.

The remainder of this paper is organized as follows. Section 2 describes some related research for CA and AD. Section 3 provides a description of the data collection used in this research and the features extracted. Section 4 describes our approaches and classification techniques. Section 5 provides implementation and experimental results. Section 6 presents the discussion and analysis. Section 7 has limitations of this paper. Section 8 gives ideas for future works. Section 9 discusses the conclusions.

2 Literature Review

Our literature reference consisted of 18 papers. Of which, we as a team, read the first 2 papers and the other 16 papers were divided evenly among the team with 4 papers for each person. These papers later helped us create literature review documents for reference when doing the project. These papers are in the area of continuous authentication schemes based on machine learning. Below are some of the papers out of the 18 that we as a team went through.

In one of the papers S. Fu, D. Qin [1] presented a combined neural network model (CNN-RNN) for mouse behavior-based user authentication. They used raw sequential mouse data as input. Moreover, they evaluated it on a real dataset which consisted of 15 users with 300 trials each. The result from their approach yields a 3.16% EER and a 99.39% AUC, with an authentication delay of 6.11 seconds on average. Wu, G. [5] talks about in his paper about getting

2.9% EER in their method. This shows the effectiveness of applying deep learning techniques for static mouse behavior-based user authentication.

In another paper L. Gao et al. [2] talked about two classifiers that are fused at the decision-level, which lessens their heavy reliance on training data. They used freely accessible Balabit dataset. Fifteen users' dynamic mouse data made up this dataset. Marcus Tan, Y. X. [6] wrote in their paper about the usage of the Balabit dataset as it is considered to have less occurrences of anomalies. The team breaks down the user's operation events into features like motion length, curvature, curvature, etc., and derives the user's behavior characteristics from the curve characteristics. 60 groups of mouse curves were used to extract 24.3% of the EER. It is possible to increase performance by 11.2% by using 3600 sets of mouse curves. Their approach combines the support vector machine optimized by genetic algorithm with the k nearest-neighbor algorithm to provide an error rate that is lower than that produced by the two approaches alone. Bours, P. [7] wrote about their results were action specific and are optimized using genetic algorithm.

Mondal, S. [3] talked about how they concentrated on developing a context-free continuous authentication system that responds to each individual action a user takes. They provide a reliable dynamic trust model algorithm that is adaptable to any continuous authentication system, regardless of the biometric modality. In another paper [7] Bours, P. described in the trust model that, they see each and every action leading to a change in trust and therefore, it is possible that each particular action led to a lock out of the current user because one such action dropped the trust value below the lock out threshold. This dataset was gathered from 53 users using their data collection software in completely uncontrolled conditions. In order to avoid a scenario in which an attacker avoids detection by restricting to one input device because the system only checks the other input device, they considered both keystroke and mouse usage behavior patterns. The best finding of this study is that 50 out of 53 genuine users never have their accounts accidentally locked out by the system, while the remaining 3 genuine users ($\sim 5.7\%$) occasionally have their accounts locked out, typically after 2265 actions. In addition, only three out of 2756 impostors have evaded detection, or 0.1% of all impostors. After 252 actions on average, imposters are discovered.

3 Dataset

Our dataset consists of raw mouse dynamics data collected during the gameplay of a tower defense game on Roblox. The raw data collected includes the following features: ID, cursor X and Y positions, the button pressed, and button press duration.

4 Data Cleaning

In the data preprocessing step, we utilized a feature set that included various statistical, dynamic, and trajectory-based features. The extracted feature set used in our study consisted of the following features: The mean, standard deviation, minimum, and maximum of X speed, Y speed, and overall speed, jerk, and angle; X acceleration, Y acceleration, and overall acceleration; speed or acceleration squared over distance.

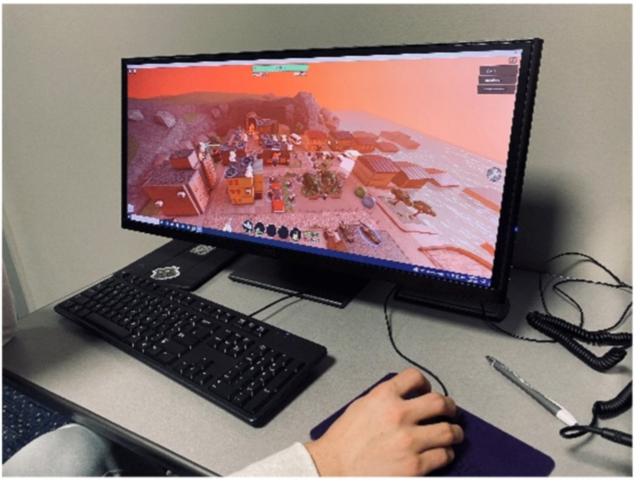


Figure 1: One of our subjects playing the game 'Tower Defense Simulator' from Roblox

To preprocess the data, we first organized the raw data into sequences of 128 rows each, representing roughly 1 second of consecutive mouse dynamics data points on average per subject. Then, for each sequence, we calculated the above-mentioned features using appropriate formulas.

After calculating the features for each sequence, we further processed the data by normalizing or scaling the features, as needed, to ensure that they were on the same scale and to avoid any potential bias introduced by differences in feature magnitudes. This ensures that all features were treated equally during the subsequent analysis and modeling steps.

5 Proposed Approach and Methodology

This paper proposes a novel approach that involves testing the suitability of three machine learning algorithms: K-Nearest Neighbors (KNN), Decision Tree

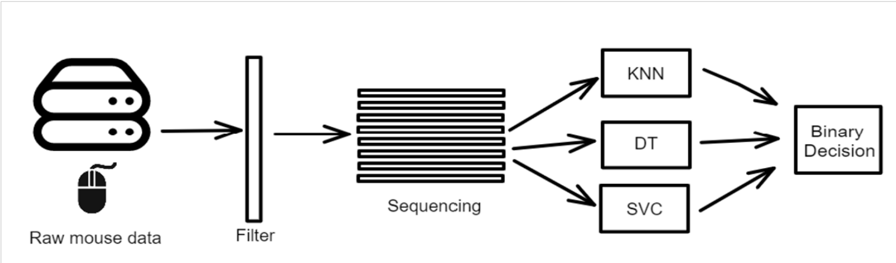


Figure 2: Model Diagram

(DT), and Support Vector Classifier (SVC). Our primary objective is to evaluate each respective model and its potential viability in a continuous authentication model. Along with that perform a comparative analysis of our models against other papers.

5.1 K-nearest Neighbors

The KNN algorithm was implemented using the scikit-learn library's K-Nearest Neighbors classifier, with hyper parameter tuning being done using GridSearch from the same library. It was found that using a k value of 3 and the Manhattan Distance metric were the best hyperparameters for this classifier.

5.2 Decision Tree

The Decision Tree algorithm was implemented using the scikit-learn library's DecisionTreeClassifier class. Hyperparameter tuning was again performed using GridSearch where the maximum depth of 50 seemed to produce the best results. It was noted that Decision Tree has a fast prediction time at the expense of a slow fit time.

5.3 Support Vector Machine (SVC)

The Support Vector Machine (SVC) algorithm with a radial basis function (RBF) kernel was used as a classifier in this study to authenticate. The SVC algorithm was implemented using the scikit-learn library's SVC class, and hyperparameter tuning was performed using a grid search where the penalty parameter C was set to 100 and the kernel coefficient gamma was set to "auto", where with the 55 features we used, evaluates to 0.018.

6 Results

In this section we will discuss a comparative analysis of the 3 models we have trained. We have extracted a total of 55 features from the mouse for analysis, as mentioned in section. Our dataset contained a total of 1,246,577 events genuine user events, for an average of 83,105 events per subject.

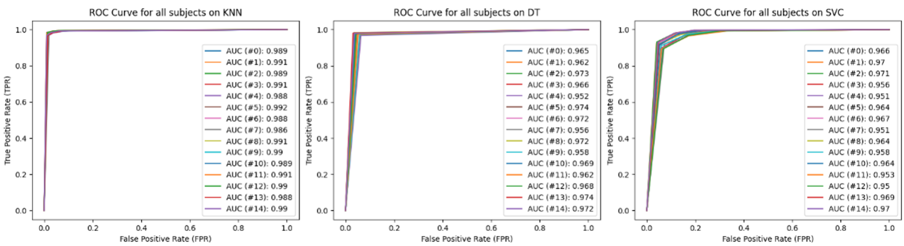


Figure 3: ROC curves for KNN, DT, and SVC models respectively

For evaluation, we split the training dataset into two different sets, one for training and one for testing. We trained the models on the training data, and tested using the testing set. We split the data using a standard 70-30 split, where 70% of the binary classified data was used for training, and 30% was used for testing. To measure the performance of each individual model, we used the following evaluation metrics: accuracy, F1-score, precision, recall, false positive rate (FPR), false negative rate (FNR).

Table 1: Continuous authentication results for our models

Classifier	KNN	DT	SVC
Accuracy	97.49%	96.61%	95.51%
Recall	98.61	97.58	98.61
Precision	96.44	95.73	92.83
F1-Score	97.5	96.6	95.64

Accuracy is the simplest metric to evaluate a model and is solely the percentage of correctly guessed sequences of events. According to Table 1, we were able to achieve accuracies ranging from 95.51% for SVC and up to 97.49% for KNN. False positive rate (FPR) is the rate in which an imposter user is authenticated as a genuine user, and false negative rate (FNR) is the classification of a genuine user as an imposter user. While aiming to reduce both FPR and FNR, we should aim to reduce FPR, since typically authenticating an imposter user is much more detrimental to a secure system than locking out a genuine user [4].

We achieved a FPR of 3.68%, 4.51%, and 8.08% for KNN, DT, and SVC

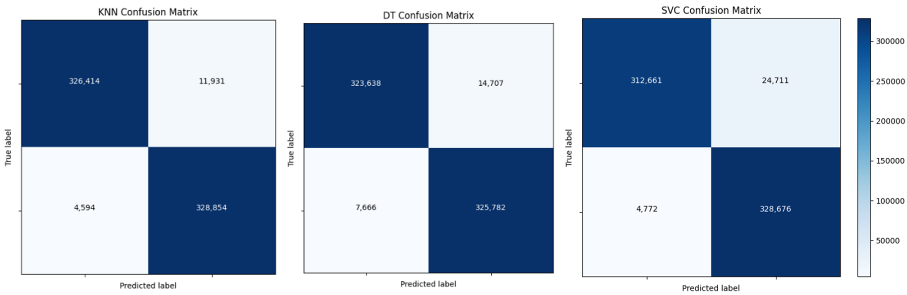


Figure 4: Confusion matrix metrics for KNN, DT, and SVC respectively

respectively. For FNR, we achieved 1.37%, 2.26%, and 1.32% for KNN, DT, and SVC respectively. In our models, KNN performed the best. For our KNN model, we were able to achieve an accuracy as high as 98% for user 5 in our model, and as low as 96.6% for user 7.

7 Discussion and Analysis

The papers reviewed by the team suggest that continuous authentication systems based on mouse dynamics can achieve relatively low error rates with acceptable authentication delays. For example, the CNN-RNN model presented in [1] achieved a 3.16% EER and a 99.39% AUC with an authentication delay of 6.11 seconds on average. In Figure 3, we showcase an AUC ranging from 95% to 99%. Similarly, the continuous authentication system proposed in [2] achieved an error rate of 24.3%, which can be improved by using a larger dataset.

Our research shows that the mouse dynamics method for access control has a false rejection rate that nearly meets the European standard of 1%, indicating that it is effective at discriminating between users. However, the false acceptance rate is still above the required 0.001%, making the method unreliable as a standalone authentication method. It is important to note that authentication methods not meeting the European standard should not be solely relied upon, but combining mouse dynamics with other authentication methods could be valuable for user discrimination tasks.

In conclusion, the literature review highlights continuous authentication using mouse dynamics as a promising research area that can lead to practical and effective authentication systems. Further research is required to develop more robust algorithms and feature sets that improve the accuracy and reliability of these systems.

Table 2: Comparative analysis of our model evaluations versus other papers' model evaluations. When other papers have multiple results, ranges of their results are mentioned.

Paper	Method	Results	Contribution
[1]	CNN-RNN Model with Mouse Movement	EER = 3.16%, AUC = 99.39%	Effective authentication time
[7]	SVM model with Mouse Movement	FRR = 0.86%	Fine-grained mouse movement metrics
[9]	NN with Keyboard Dynamics	EER = 2.13%	High accuracy in a far lower processing time
[10]	Naïve Bayes and SVM with Mouse and Keyboard	FPR = 2.10%, FRR = 2.24%	Multi-modal sensors with feature comparison
[11]	KNN, DT, and SVC with Mouse Movement	Accuracy = 54% – 95%	Lightweight mouse dynamics set to perform authentication
Our Research	KNN, DT, and SVC with Mouse Movement	Accuracy = 95.5% - 97.5%, FPR = 3.68% - 8.08%, FRR = 1.32% - 2.26%	ML model comparison with a novel mouse dynamics dataset

8 Limitations

Some of the limitations that we faced are Our dataset only contains mouse movements from a mouse-intensive video game, and our results may not translate to day-to-day administrative tasks. For KNN, we noticed small False Acceptance Rates (FAR) and False Rejection Rates (FRR), of 0.0368 and 0.0137 respectively, however we would want to minimize FAR, as false positive authentications are much more detrimental to secure systems than false rejections. Our models were trained on data that all came from identical hardware, and our models may not perform the same if different subject data came from different computers.

9 Future Work

Future work could investigate the use of ensemble methods or other sets of dynamic features beyond mouse movement biometrics to further enhance system accuracy and robustness. Additionally, interpreting the decision-making process of the models through techniques such as feature importance analysis or model visualization could provide insights into the factors that contribute to the models' predictions and improve their decision making. It is also worth noting that in smaller tested feature sets the respective models' decision-making varied widely in the tens of percents. This suggests that there could be obvious variations in how some features map to a user differently based on their biometrics, demonstrating that specific features may be more helpful to specific subjects.

10 Conclusion

In conclusion, our research presents a comparative study of machine learning-based continuous authentication methods using a novel mouse dynamics dataset. Our study extracted features from the raw data and used it to train and evaluate machine learning models. Our results show that our k-Nearest Neighbor classifier is the most effective model for continuous authentication using mouse dynamics for our dataset and features selected.

In addition to the technical aspects of our research, it is important to consider the broader implications of our findings. Machine learning-based continuous authentication has the potential to significantly improve the security of online systems by providing near real-time authentication without disrupting the user's experience. Our research contributes to the development of these systems by providing a comparative analysis of three different machine learning models that can be used for continuous authentication based on mouse dynamics.

References

[1] Shen Fu et al. "RUMBA-mouse: Rapid user mouse-behavior authentication using a CNN-RNN approach". In: *2020 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2020, pp. 1–9.

[2] Lifang Gao et al. "Continuous authentication of mouse dynamics based on decision level fusion". In: *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2020, pp. 210–214.

[3] Soumik Mondal and Patrick Bours. "A study on continuous authentication using a combination of keystroke and mouse biometrics". In: *Neurocomputing* 230 (2017), pp. 1–22.

[4] Yi Xiang Marcus Tan et al. "Adversarial attacks on remote user authentication using behavioural mouse dynamics". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–10.

[5] Guannan Wu et al. "A continuous identity authentication scheme based on physiological and behavioral characteristics". In: *Sensors* 18.1 (2018), p. 179.

[6] Yi Xiang Marcus Tan et al. "Adversarial attacks on remote user authentication using behavioural mouse dynamics". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–10.

[7] Patrick Bours and Soumik Mondal. "Performance evaluation of continuous authentication systems". In: *Iet Biometrics* 4.4 (2015), pp. 220–226.

Further Reading

[8] Margit Antal and Elöd Egyed-Zsigmond. "Intrusion detection using mouse dynamics". In: *IET Biometrics* 8.5 (2019), pp. 285–294.

[9] Nyle Siddiqui, Rushit Dave, and Naeem Seliya. "Continuous User Authentication Using Mouse Dynamics, Machine Learning, and Minecraft". In: *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE. 2021, pp. 1–6.

[10] Nan Zheng, Aaron Paloski, and Haining Wang. "An efficient user verification system via mouse movements". In: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011, pp. 139–150.

[11] Ahmed A Ahmed and Issa Traore. "Biometric recognition based on free-text keystroke dynamics". In: *IEEE transactions on cybernetics* 44.4 (2013), pp. 458–472.

[12] Lex Fridman et al. "Multi-modal decision fusion for continuous authentication". In: *Computers & Electrical Engineering* 41 (2015), pp. 142–156.

[13] Chien-Cheng Lin, Chin-Chun Chang, and Deron Liang. "A new non-intrusive authentication approach for data protection based on mouse dynamics". In: *2012 International Symposium on Biometrics and Security Technologies*. IEEE. 2012, pp. 9–14.

Examine the Usability and Interaction of a Next Generation Virtual World Education Grid Model*

Imad Al Saeed
Computer Science department
Saint Xavier University
Chicago, IL 60655
alsaeed@sxu.edu

Abstract

The demand and supply for online courses vs traditional classroom-based counterparts are on the rise. As new technologies emerge, evolve, and advance, more tools become available to educational institutions, thus enabling them to enhance and enrich learning models to achieve new performance levels. Virtual classes over the Grid could take advantage of these technologies in order to exchange information and ideas between instructors and their students by enabling the use of as many of the senses as possible. Virtual World Grid for Education (VWEG) is a clear example of the commitment towards making the learning experience accessible anytime and from anywhere, pertinent and convenient to the global community of learners. This research study is a continuation of the three research papers (as the future work) that were previously accepted and published through the CCSC MW conferences [9, 10, 8]. The study examines the creation of a next generation of a standalone educational environment model for students and educators to provide new paradigms for learning.

1 Introduction

Over the years, studies show students' performances in virtual classes as well as in face-to-face classes [10, 3, 4]. Facts such as instructional method, prior knowledge, learning style, time on task, ability, media, and teachers' effects make education methods more successful. Several studies have been done to determine significant differences between traditional face-to-face learning methods and Virtual learning. One advantage of online courses over face-to-face classes is that online classes are more effective for students because in these kinds of courses, students can leave the class anytime he/she wants to re-energize [4].

Traditional courses rely on the use of available resources at hand [7]. For example, a traditional course could be as limiting as reading a textbook to remember the facts for a later examination [8]. In a virtual world grid, the course content would be delivered in such a way that it is experienced by students through using various resources that could be virtually created as needed to enhance the learning experience. These resources would be tailored to support some of the basic learning processes like visualization, assimilation, retention, reasoning, processing, comprehension, deduction/deliberation, and speculation among others [5, 2]. Throughout the interaction and dynamics that usually takes place during virtual class sessions, active students would contribute to the class body of knowledge with their own retrospection of the concepts being unveiled. Ultimately, the students could use some of that new collaboratively made knowledge to complete their assignment (case studies, projects, role play, labs, discussions, simulations, interactive games, and if any other learning techniques).

To have an effective online course, special technology and tools are needed. Students need a learning environment that supports their needs and is an easy-access course. In other words, the designers of these courses are walking a fine line between course options and tools. Students need centric courses and interoperable and invisible technology designs. Course designers need to design courses not only to connect students to the internet but also to the materials of online classes. They need to offer easy-to-use tools with great communication. All these would be provided by the Virtual World Education Grid environment.

The Virtual World Grid (VWEG) for Education provides a public utility for digital media that combines interactive 3D graphics, commercial game and simulation technology, text, and voice chat (Voice over IP/VoIP) for the academic community. The Grid design provides the structure and tools for developing collaborative online courseware. It also offers the capability to design rich digital environments for student exploration, and a ready environment for institutions, teachers, and students to collaborate and share their skills and experiences for increased levels of innovation and creativity. The VWEG is a window of future learning. The explosion in the use of social media, the technological advances in graphics, and the communications power of the Internet bring the world to the fingertips of anyone ready to venture into the vast sea of information.

The 21st Century is a world vastly different from the preceding century. In order to remain competitive, educational institutions need to deliver content in a format their patrons are comfortable with. The next generation of students, and even some of the current generation, are becoming more and more involved in the online community as evidenced by the explosion in social media websites such as Facebook, Twitter, Whatsapp, etc. Additionally, virtual worlds such as those in Linden Labs Second Life and, even more prevalent, the massively multiplayer online gaming industry are becoming more and more populated.

Based on new and emerging distributed computational grid technologies, the VWEG builds upon existing Internet and web standards to create a unique network optimized for digital media delivery, storage, and processing. The Grid is an open and extensible platform enabling immersive education that can be both local to an institution and geographically dispersed, bringing together a diverse set of students.

The implementation and use of the Grid will have a positive effect on student learning that shall be directly noted in increased student motivation and participation. This increase could be achieved by studying real-world situations in the resource-rich virtual world environment and promoting self-discovering learning techniques while using a mentor/coach approach. This approach effectively shifts the learning role of the student from passive to active. Mentors, instructors, and coaches have the responsibility of encouraging various learning techniques, monitoring progress, and providing appropriate feedback via synchronous and asynchronous interaction. The use of the Grid will enable the students to explore concepts and information using virtually created models, objects, and behaviors from the real world that will give way to structured knowledge that might be based on the students' own analogies to produce knowledge that is applicable to the real world. The VWGE provides private and secure learning environments for teachers and students of all ages, where every avatar is assigned to a unique name/entity. It maintains strong security measures through group association limiting communications to recognized group members (both current and past). Yet, the VWEG provides an architecture that allows avatars to cross multiple nodes located across the Internet. Each node maintains security, offering secure access across the cloud to other platforms in line with our documented Governance structure. Security is multilayered, and the content is peer-reviewed to minimize features that are not within the governance standards set by educational policies developed by stakeholders, and content is designated pertinent and permissible within the learning environment.

2 Virtual Education Grid features

The main features of the actual configuration are:

- Invisible technology helps students have access to courses easily. Details and complexity of configuration and installation of any integrated are hidden by these invisible technologies. Great courses will take advantage of these technologies to consolidate interaction, information, activities, and feedback.
- The pedagogy used in the new course design instead of traditional learning models is more about memorization. Pedagogy is learning the facts by associating them with the subject domain. In other words, it shifts from learning to applying the facts.
- It supports different learning styles. For example, some spatial and visual students need to read textbooks, watch movies, and see images to comprehend the idea, kinesthetic students need physical interaction to understand, and tactile students need to touch and interact with the environment.

3 The Implementation Plan of VWEG

Baseline Requirements:

- Open Access - The VWEG should be accessible to all organizations and individuals through open usage policies and procedures, but also preserve the right to make the server private for events.
- Open APIs and interfaces - VW Education Grid will be equipped with open Application Programmer Interfaces (APIs), protocols, and interfaces to provide software developers with comprehensive, flexible, and open access to the Education Grid.
- Open and interoperable file formats - Software configuration should be readily transferable to other hardware/software platforms and environments.
- Open hosting with conformance and compatibility - Conformance and compatibility are achieved with relevant education technological and operational standards.
- Support for multiple asset types and content formats - It should revolve not only around 3D content but also many other asset types and content formats (e.g., text, images, audio, video, mini-games, etc.).

- Quality control - educational content (i.e., learning experiences and any digital media content assets such experiences are comprised of, such as 3D objects, images, videos, mini-games, and so forth) must be adequately reviewed, categorized and tagged with metadata prior to being formally accepted into the VW education Grid.
- Multitiered metadata architecture - The metadata model will include provisions for qualitative analysis, raking/rating, and tagging by domain experts (subject matter experts), educators, students (learners), and all other users of the system. Content search services must support searching by any tier of metadata and by any combination of metadata tiers. Protected learning environments - It requires the establishment of learning environments with varying levels of protection. Public environments will impose the fewest restrictions and enable the widest distribution of learning assets and virtual classroom experiences. Private learning environments may also be created with restricted access only to authorized users in order to provide safe learning environments for example, should only engage in multi-user collaborations that involve their fellow classmates and their teacher(s).

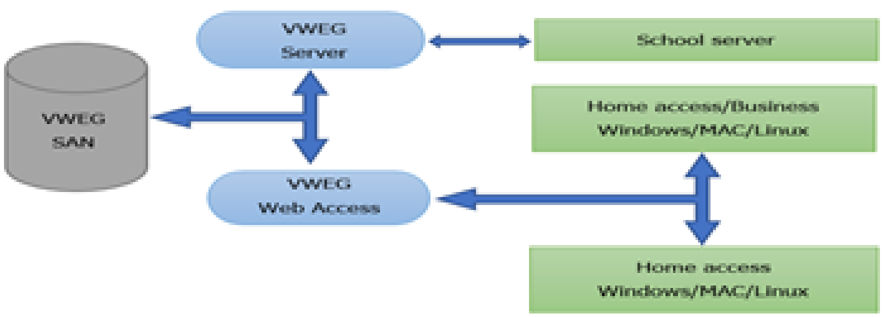


Figure 1: VWEG core system

Figure 1 shows how the VWEG core system is comprised of the VWEG Server, VWEG SAN, and VWEG Website. Since Phase 1 of the VWEG implementation is limited to the infrastructure and governance, it primarily focuses on the VWEG Server and SAN. The VWEG Server is the control structure for the educational environment. As processing nodes and storage devices are added to the VWEG, they are registered with the Grid Server. The Server is set up as the host entity for the grid operating structure and controls where parts of the grid are actively engaged in the operation of the VWEG at any given time.

4 Solution and Business Analysis

1. Scope - The adoption of existing physics engines and OpenSim resources will provide a strong foundational architecture for building the Grid Blocks project. To facilitate educational venues a vehicle to store and retrieve digital learning objects is created using existing techniques with visible enhancements. The project can help in identifying and searching for a suitable or optimum characterization of the VWEG to be more efficient and can lead to the more effective usage of shared resources. This project will consist of four Phases:
   - Phase I - Concept of Operations.
   - Phase II - Content Development.
   - Phase III - Multi-Lingual Support.
   - Phase IV - Adaptive Learning Behaviors.
2. Limitation - The project does not include the design of a new client software tool. A couple of clients are available, including Hippo.
3. Industry Analysis - There are several hundred universities, colleges, and government entities with significant presence and operations in Second Life. Developing a low-cost or even open-source alternative to Second Life is only one part of the needed solution. For an alternative solution to be viable, there needs to be a data/object transfer application or import function so that content can be transferred from Second Life to the new virtual environment. Some of the educational and government entities are likely to have spent considerable time and, in some cases, money on their in-world content. Losing this content is not likely to be a viable option. This project utilizes existing servers at the educational base dedicated to islands for the educational organization. Each island utilizes the security existing for their population's access to the Grid. Client software is compatible with the in-house servers and the bridging between servers Grids at other educational organizations utilizes existing bridge software systems.
4. The reputation of this Grid needs to be built up by bringing attention to the ongoing research and development of the partner organizations. Recognizing outside agencies with similar developments can help to develop trust in this Grid. Through this trust, Grid can draw from corporate and government sponsorship to minimize the costs of operations.
5. Growth Potential - What is the base value of your business? Here you should describe how industry growth will affect your business potential. Figure2 showed data for the Educational Software Market Size from 2019 to 2025.
6. Competitive Analysis - One of the virtual world education grid competi-

tors is OpenSim. OpenSim is an open-source multi-platform, multi-user 3D application server built by reverse-engineering Second Life's published APIs and portions of SL code that Linden Lab has open source [9]. Currently, OpenSim is alpha testing [1]. One of the strengths can be used to create a virtual environment that can be accessed through a variety of clients, on multiple protocols. Open Simulator allows virtual world developers to customize their worlds using the technologies they feel work best - we've designed the framework to be easily extensible [6]. One of the weaknesses of this type of virtual world is its susceptibility to attacks by any web novice users. It is easier to copy assets with a web-based client. The weakness is that asset servers are connected to the public internet, and the protocol for interacting with them is public. The VWEG will result in an improvement over the OpenSim because it has great potential in Language learning and teaching, projects turn out to be motivating both for the students and instructors, and task- based activities are favored over other methodologies. Additionally, OpenSim has high technological requirements and a hard user interface compared to VWEG. On the other hand, the World of Warcraft (WoW) can be seen as a competitor to the Virtual World Education Grid. There are more than 12 million players currently using WoW (Blizzard, 2010). The existence of the tremendous user base represents a strength in that users would not have to switch to a new unfamiliar system in case WoW is used to host educational events and content. But the new users could be discouraged based on WoW's gaming focus. The Virtual World Education Grid will result in an improvement over the WoW worldwide virtual world because it will be dedicated to academia. This will give it a professional reputation, which will potentially attract additional credible participants.

7. Marketing Strategy - Incorporate Virtual World Educational Grid (VWEG) as a nonprofit venture for educational research/development by targeting universities and expanding into the public school system as development allows VWEG to exist as a separate legal organization in order to own its own property and its own bank account. VWEG can be incorporated by filing articles of incorporation or other charter documents with the appropriate local state office. But to do so, a board of directors will be required. As a profit-seeking venture the leveraging for the VWEG project could be handled using the Freemium model but only focusing on educational institutions with basic capabilities freely given and charges for premiums.

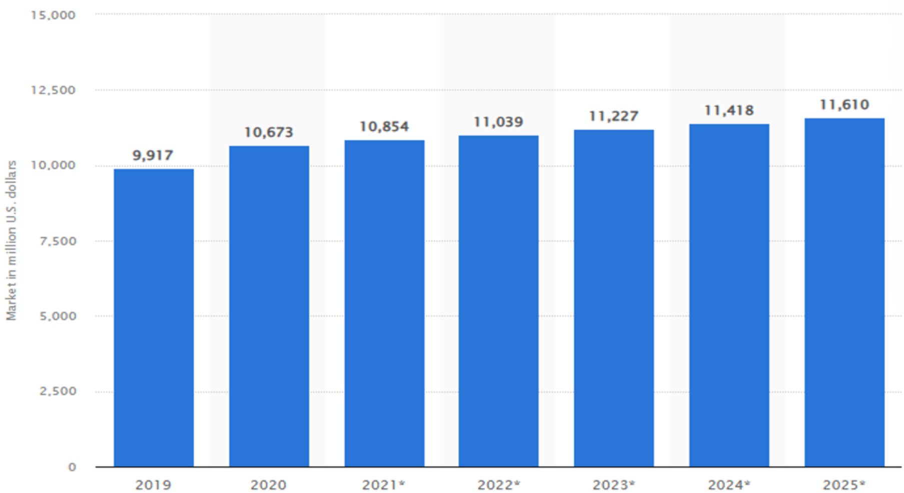


Figure 2: Educational Software Market Size

5 Proposed VWEG Modeling Activities

The purpose of the modeling activities is to offer a framework that could be used for supporting a larger collaborating modeling build process to explore, imagine, and create an innovative approach to simulate a 3D teaching and learning model for testing purposes (see Figure 3). This modeling framework could be used to construct an action plan for implementing the concept design. Figure 3 shows the general steps that could be followed and implemented to perform any design-build to be implemented into any educational environment.

The general steps should consist of the conceptual phase, testing phase, and implementation Phase. The Conceptual Phase consists of the initial concept where all requirements are captured and derived further into a low-fidelity prototype. After the Conceptual Phase, the designer or design group moves into the Testing Phase. In the Testing Phase, the system or its interface should be tested, revised, and redesigned based on the result and users' feedback. The last phase "the Implementation Phase" should be performed and implemented in a real-world environment.

6 Teaching and Learning Environments Setup

There is a big need for establishing a framework for designing and implementing secure teaching and learning models used by students to enhance the availabil-

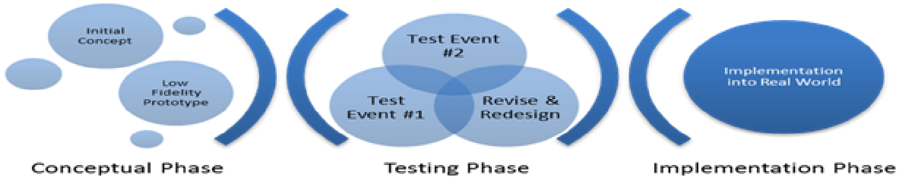


Figure 3: Virtual modeling process

ity of information and overall learning and teaching quality. This also will lead to establishing another framework that could be used to exchange valuable information between universities with similar interests. In this way, the consumers' awareness of the data practices will be increased and provided an easy and cost-efficient way to compare the data practices between them to improve the learning and teaching best practices. The Conceptual Phase consists of simulating a learning model through second life including designing a computer network equipped with a full graphical user interface (GUI) system and simulating the roles of individuals and the responsibilities of those who are responsible for exchanging information through a network (See Figure 4 and Figure 5). A low-fidelity prototype was completed over four separate phases:

Phase 1 - An in-depth analysis of materials available including communication literature and other analysis issues like the real system in the real world.
Phase 2 - Establishing the privacy, policies, and security issues and incorporating them into the design process of the learning Model.
Phase 3 - Simulates a computer network including the software application (GUI) like the real- world system.
Phase 4 - An in-depth test to assess and analyze the understanding of the learning Model including Privacy and security issues and further adjust the system using consumer input and perform all learning processes for students. These processes are repeatable for more students as well.

The general goal for this project is to perform a usability test by observing the users' behaviors during performing four specific tasks on the teaching and learning model including analysis of materials available, setup the policies, configure security issues, successfully log-in to the system, and perform a full learning process for students. All that was done with limited information the user got from the developer. The developer only shows the users what to do, not how to do it. Five participants from SXU – Computer Science department were volunteers to visit the environment and do the testing. Testing occurred in person with the user directly manipulating the prototype. The user interface prototype included all the web-based GUI features of the real system in real life. The test observer verbally described the test task and allowed the user to complete the task unaided. The test observer provided verbal feedback to the user strictly adhering to this test procedure. The user was presented with only one test case at a time.

Figure4. VWEG with participants.

Figure5. GUI of the system used within the VWEG.

Metrics and Measurement - For each test, the following variables will be recorded: (1) The runs time out for performing a task, (2) The error handling for a task.
Successful criteria - The success criteria for each task would be achieved by:

1. Initial setup of the system including policy, privacy, and security.
2. Successfully logged in to the system.
3. Successfully perform one actual learning model sequence of designing a full classroom, uploading the instructor presentation slides, or updating the existing information.

All these sequences must be performed successfully to perform one successful test to pass, otherwise the learning and teaching model needs to be revised.

7 Results and Reflections

The test plan and test procedures worked very well according to the plan to achieve it. Each user successfully completed all four tasks and then responded to the survey. The observer recorded the run time for performing each task (Table 1), and the error handling for a task. The time to perform the task and survey came within the developer's expectations. The survey results were high. That could be due to the simplicity of the test tasks. This is the stage of the low-fidelity prototype used for development, evaluation, and discovering the errors that could be addressed in the design process. There is a need for future interaction design with other professors who could likely benefit from techniques for adding more complexity to the software design. Interaction is necessary for enhancing the design process. The interaction processes would be very helpful to support more convergent interaction including collaborative environments, embodied virtual systems, and immersive virtual reality.

Table 1: The observer's recorded run-time for each task

Task	Challenge	Action	Discovery	Current Level	Minimum Acceptable	Planned Target	Best Possible	Test Results
1	Introduction	Find the introduction manual and familiarize yourself with the test procedure.	5	3 min	2 min	2 min	3 min	5
2	Initial setup	Run the software and do all the necessary preparation to access to the system	5	2 min	2 min	3 min	2 min	3
3	Establish the policy, privacy, and security.	Establish the system policy, privacy, and security.	5	5 min	5 min	4 min	3 min	4
4	Process the information and send it.	Perform all the task in sequence.	5	15 min	13 min	12 min	11 min	15
Survey	Overall Reactions - wonderful to terrible.	QUIS Survey - scale: 1 to 9		5	6	7	8	9
Survey	Easy to use - difficult to easy.	QUIS Survey - scale: 1 to 9		3	6	7	8	8
Survey	The Capabilities of the System.	QUIS Survey - scale: 1 to 9		5	6	8	9	7

8 Participant Feedback and Suggestions

Table 2: The participants Feedback and suggestions

Participant	Feedbacks and Suggestions
Participant #1	Add a signal communication indicator to make an indication that there is communication between the user and the software.
Participant #2	Add a registration and visitor counter to count the number of students who already visit the site and run the test.
Participant #3	Make a PowerPoint presentation that includes all the testing sequence steps. In this way, any user can read these instructions and easily perform the test.
Participant #4	For security purposes, the user must ask the administrator permission to start the test and the administrator has the right to allow or deny the user.
Participant #5	For security purposes, it is a good idea to add a fingerprint prime to recognize and allow only the authorized person to perform the brake test for more security purposes.

9 Conclusion

In conclusion, creating Virtual Educational Grid as a next generation learning model for both the students and educators will provide new paradigms to be used to personalize the learning process for each student. The Virtual Educational Grid would be a standalone educational environment, promoting new ways to learn within virtual worlds. Teamwork will be encouraged, and students will play the collaboration role. Having strong online collaboration skills and communication will cause professional benefits. The requirements for a computer-supported cooperative project are an understanding of objectives, guidance, good communication, and feedback. This study presents evidence that the new VWEG provides the segments to the design framework that described the proposed design activities for an effective and course-efficient environment.

References

[1] Association of American Railroads. Retrieved on May 05, 2022. URL: http://www.aar.org.

[2] Yiyu Cai, Sui Lin Goei, and Wim Trooster. *Simulation and serious games for education*. Springer, 2016.

[3] Maurice Dawson et al. "Technology enhanced learning with open source software for scientists and engineers". In: *INTED2013 Proceedings*. IATED. 2013, pp. 5583–5589.

[4] Christian Ebner and Andreas Gegenfurtner. "Learning and satisfaction in webinar, online, and face-to-face instruction: a meta-analysis". In: *Frontiers in Education*. Vol. 4. Frontiers Media SA. 2019, p. 92.

[5] Maurice Eugene Dawson and Imad Al Saeed. "Use of open source software and virtualization in academia to enhance higher education everywhere". In: *Increasing Student Engagement and Retention Using Immersive Interfaces: Virtual Worlds, Gaming, and Simulation*. Emerald Group Publishing Limited, 2012, pp. 283–313.

[6] Horácio Gaspar et al. "Research priorities in immersive learning technology: the perspectives of the iLRN community". In: *Virtual Reality* 24 (2020), pp. 319–341.

[7] Leonel Morgado et al. "Integration scenarios of virtual worlds in learning management systems using the MULTIS approach". In: *Personal and Ubiquitous Computing* 21 (2017), pp. 965–975.

[8] Imad Al Saeed. "Examine the adoption of a virtual world to improve the hybrid courses". In: *Journal of Computing Sciences in Colleges* 35.5 (2019), pp. 17–27.

[9] Imad Al Saeed. "Examine the Use of Virtual Worlds for Modeling, Prototyping, and Testing Purposes in a Classroom Setting". In: *Journal of Computing Sciences in Colleges* 38.4 (2022), pp. 14–21.

[10] Imad Al Saeed. "Examining the use of the virtual world to enhance group learning in hybrid courses". In: *Journal of Computing Sciences in Colleges* 36.4 (2020), pp. 32–40.

Clicking Your Way to Security: A Review of Continuous Authentication with Mouse Dynamics*

Ali Rashid, Cole Schoenbauer, James O'Reilly
Marcho Handoko, Rushit Dave, Mounika Vanamala
Computer Science
Minnesota State University, Mankato
Mankato, Minnesota 56001
{ali.rashid,cole.schoenbauer}@mnsu.edu
{james.oreilly,marchosetiawan.handoko}@mnsu.edu

Abstract

The increasing reliance on information management technology and the prevalence of cyber threats have made user authentication an important issue in computer security. The traditional approach of static or one-time authentication has its limitations, and continuous authentication has emerged as a promising approach to keep systems secure. Continuous authentication utilizes behavior-based metrics to validate users' identity, and it is typically achieved using machine learning techniques to model the user's behavior throughout their session. This paper reviews the literature on continuous authentication, focusing on movement-based authentication, particularly on mouse dynamics. We discuss the advantages and limitations of continuous authentication and highlight the need for further research in this area. This paper aims to provide a comprehensive review of the current state of continuous authentication research and to identify future research directions.

1 Introduction

In the modern era, the security of systems which govern nearly all sectors of civilized life is of utmost importance. As a global community, we have become increasingly reliant on the use of information management technology, which stores and processes sensitive information, in our everyday lives. For example, our smartphones, cars, laptops, and a plethora of other systems contain personal information which should be kept safe. Thus, it is paramount that the users who access these systems are authenticated with diligence to ensure the security of our information.

Simply put, authentication is the verification of a user's identity before allowing the user access to protected information. There are many forms of authentication which utilize a combination of three authentication approaches: knowledge- based, possession-based, and biometric-based[17]. Knowledge- based authentication schemes validate user identities by verifying something that a user knows (passwords, patterns, pins, etc.), possession-based authentication schemes rely on the user having access to something which validates their identity (cards, email accounts, etc.), and biometric-based authentication uses the biometrics of the user themselves to verify their identity (fingerprints, faces, iris, etc.).

These forms of user authentication are often known as static or one-time authentication as it verifies the user's identity once at the beginning of their session. However, this mode of authentication has a fundamental flaw which makes it susceptible to cybersecurity attacks. Typically, static authentication only provides an initial guarantee of the user's identity when they begin their session; however, it fails to protect a system over the entire course of the user's session. In such a case, it is possible for an unverified user to gain access to protected information by either hijacking a user's current session or bypassing the initial authentication query. As such, static authentication alone is insufficient to guarantee the security of a system.

Continuous authentication schemes aim to keep a system secure across the entire course of a user's session. In this approach, a user is validated using behavior-based metrics which are thought to be uniquely identifiable[8]. This is typically done using machine learning approaches which construct a model of the user's behavior that is continually evaluated throughout their session. Thus, continuous authentication aims to keep systems secure using a more dynamic approach.

There are several different types of metrics regarding a user's behavior used in continuous authentication, each with their own advantages and disadvantages. A user's speech pattern may be used for continuous authentication as it identifies users based on the pitch, rhythm, and tone of their voice. Facial features may also be used to identify users along with eye tracking which uses patterns of an individual's gaze[1].

Another facet of continuous authentication which is under active research is movement-based authentication. This approach most generally identifies users based on the dynamics of their behavior when interacting with a device. Within movement-based authentication, keystroke dynamics (how the user types), youch dynamics (how a user interacts with a touchscreen), and mouse dynamics (how the user interacts with a mouse) are most commonly discussed.

However, continuous authentication still has many limitations which must be addressed. First, it is difficult to implement as it relies on the appropriate hardware to be present in the device to measure data about user behavior. Secondly, user behavior is largely dynamic and is dependent on the user's environment which might make authentication unreliable in some cases. Furthermore, it is susceptible to attacks which mimic or replicate the user's behavior[1]. The goal for this paper is to sift through the literature on continuous authentication and to identify further avenues for research.

2 Categories of Continuous Authentication

2.1 Mouse Dynamics

Mouse dynamics involves polling the user's mouse movements and using that gathered data to determine whether the given user is an intended user or an imposter. When polling for mouse data there are many different datapoints that can be gathered and used to authenticate a user. There were some methods that were consistent among most papers we read. Mouse events such as mouse clicks, wheel usage, drag-and-drop actions, and mouse position monitoring are commonly gathered events. Each recorded event is also marked with a timestamp. Having a timestamp for each event allows for feature extraction for data like time between mouse clicks, a user's movement patterns for a mouse, typical velocity, and acceleration for how a user moves a mouse, and so on. Once these features are extracted from the data, they are commonly put through a classifier that continuously determines whether a user appears genuine or not based on their actions at that moment. In many testing scenarios researchers would build a model using machine learning or deep learning that continuously gauges a user's trust level by routinely testing their inputs against a profile that was previously built for the intended user. Therefore, if a user's inputs are similar to those of the intended user, then the user is likely the intended user and won't be locked out of the system. Otherwise, if the user's inputs are different enough to not be deemed trustworthy, that user will be flagged as an imposter and will consequently be locked out of that system. The process of continuous authentication using mouse dynamics can be seen in Figure 1.

In most cases each team of researchers gathered their own dataset. In these cases, the researchers would get volunteers to sit down and interact

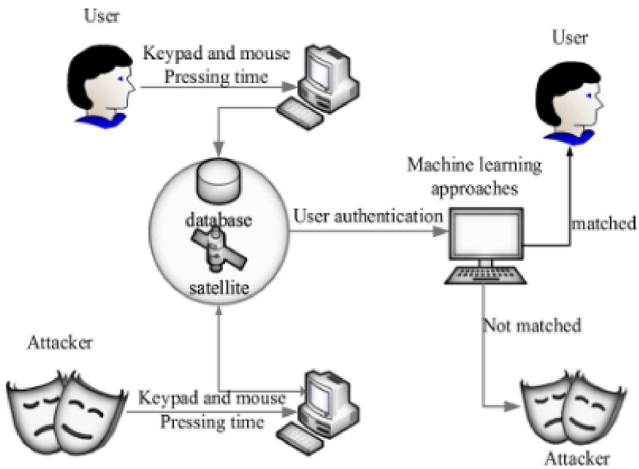


Figure 1: Mouse dynamics architecture[20]

with a program for a set amount of time. The users' mouse actions would then be recorded, and the data would be used to evaluate the classifiers built for the research. There were some publicly available datasets used as well. These include the Keystroke and Mouse Dynamics for UEBA dataset and the Buffalo dataset[25].

Results between research attempts differed greatly in many cases. The way results for papers also varied many of the times. Moundal and Bours[14] tested the efficacy of their model by performing leave-one-out testing where 1 user out of 52 was genuine and the remaining 51 users were imposters. The results for each user were then given 1 of 4 classifications based on the tradeoff between security and user-friendliness. In this study, 48 out of 52 users fell into the very good or good category[14]. This would be where either the genuine user is never locked out and the imposters are all locked out, or where the genuine user is never locked out, but some imposters were also not detected. Other research papers tested the success of their models by giving metrics. Common metrics were accuracy, equal error rate (EER), false acceptance rate (FAR) and false rejection rate (FRR). When it comes to accuracy, some studies had accuracy rates as high as around 90%[22], while others had accuracy rates as low as roughly 60%[13]. EER was usually below 5% in studies that listed it. When it comes to FAR and FRR, FAR was typically close to 0% while FRR was anywhere between 0- 5%[2, 24, 25]. This is ideal as institutions that prioritize security would rather have a genuine user locked out rather than have an imposter be wrongly accepted into a system.

2.2 Keystroke Dynamics

Keystroke dynamics is a mode of continuous authentication which uses a user's keystroke data to ensure the identity of a user. This form of continuous authentication constructs a model of the user's typing style by considering features such as duration of key held, time between keys pressed, and sequential keypress events[11]. Once extracted from the data, these features are used to create keystroke-based authentication models, an example of which is shown in Figure 2.

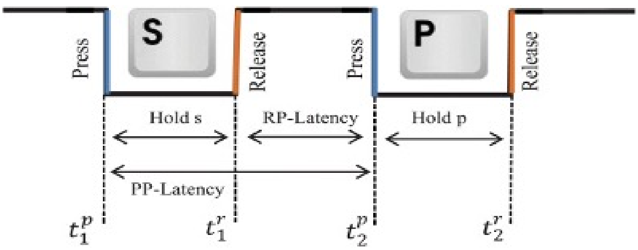


Figure 2: Keystroke dynamics features[15]

There are two types of datasets used in keystroke dynamics: fixed text and free text[15]. Fixed-text datasets model a user's typing style with the same text that is used to authenticate their identity within a system. As such, these types of datasets typically consider small text sequences and are suited for static password authentication. Free text datasets differ from fixed text datasets as the data used to model a user's typing style can be different that the data used to authenticate the user. Thus, models trained on free-text datasets are more likely to be an accurate representation of a user's typing style which can be used for continuous authentication. From the literature, there is a wide variety of datasets that were evaluated. While some were created by the researchers for testing their hypothesis, the largest datasets that were used are the Clarkson II[11] and Buffalo[20] datasets, the former being a free text dataset and the latter being mixed.

There are a wide variety of models that are being considered to implement Keystroke authentication. The models with the most promising results include deep learning approaches using a variety of neural networks and binary classification models using SVM and kNN. Studies have shown that Keystroke dynamics has the potential to achieve high accuracy ($< 90\%$) and low FAR and FRR ($> 5\%$) [9, 10, 11, 20]. Furthermore, due to the nature of Keystroke dynamics, it is unobtrusive and easy to implement on various systems which natively support keypad functionality.

However, there are some limitations. The accuracy of keystroke-based authentication systems can be affected by various factors, such as the typing environment, typing speed, and the typing hardware[9]. Additionally, there are privacy concerns about the collection of keystroke data, as it could potentially be used for malicious activities and leaves itself susceptible to data leaks which would be more troublesome compared to other approaches[19]. If these limitations are overcome, keystroke dynamics remains a promising way for secondary or even tertiary modes of authentication to be implemented.

2.3 Touch Dynamics

Continuous user authentication using touch dynamics refers to the practice of using user inputs from a touch device to identify a user uniquely. A model of the user is created over time and repeatedly checked against recent inputs in order to classify the user as a match or not to the user model.

Many mouse-based models may be applicable to touch dynamics-based models as the models receive the cursor position and click actions, but no study could be located to evaluate their accuracy. The data and features differ from that of mouse dynamic models in that each touch of the device can be thought of as a swipe with a start position, sequence of points, and end position[26] rather than a continuous location as in mouse dynamics. Touch-based models depending on the underlying hardware, may also have an absolute position, touch pressure, or touch area information associated with each swipe or each point[26]. Touch based has pretty similar architecture with mouse which ca be seen in figure 3[16]. In addition, many touch devices support multitouch or gestures. A study recently examined multi-finger touch dynamics identification and achieved a 93% accuracy but also highlighted the significant complexities of including such data[5].

Touch-based models can archive relatively high accuracy, with one study finding a 4% error rate when retesting after one week[6]. Models previously examined for touch-based authentication have been Support Vector Machine, Random Forest, Decision Tree, K Nearest Neighbor, Naive Bayes, and Logistic Regression[23].

2.4 Other Methods

In recent years, there has been an increased interest in the development of continuous authentication methods to improve the security of digital systems. Among these methods, speech recognition, face recognition, eye tracking, and heart rate analysis have emerged as promising approaches. Speech recognition uses a user's voice as a biometric identifier to continuously authenticate their identity. Face recognition, on the other hand,

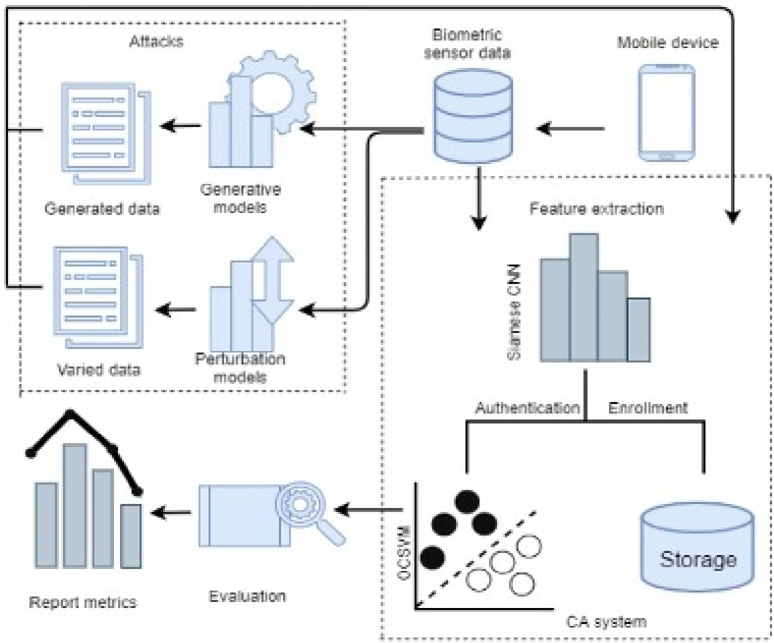


Figure 3: Touch dynamics architecture[16]

uses specific facial characteristics to verify a user's identity over time. Eye tracking is another method that continuously monitors a user's eye movements and iris shape for authentication purposes. Finally, heart rate analysis is a continuous authentication method that identifies individuals based on their unique heart rate patterns. These methods offer several advantages over traditional authentication methods, including increased security and improved user experience. In this literature review paper, we will examine the current state of research on these continuous authentication methods, their strengths and weaknesses, and their potential applications in various domains.

The biometric technique of speech recognition verifies a user's identity by listening to their speech. This technique has been demonstrated to be effective at identifying people, and it can be combined with other authentication techniques to increase security[21]. Thomas P. in his study compares different methods of continuous authentication using keyboard, mouse, and voice recognition shown in Figure 4. Early outcomes show promise in the use of machine learning techniques to enhance voice recognition accuracy[4]. However, it is restricted when someone is speaking, which hardly ever happens while working. It is also only appropriate for tools that frequently use voice, such voice assistants.

Figure 4: Mouse, Keystroke, and voice comparison architecture[21]

Facial recognition is another biometric method that uses a user's facial features to authenticate their identity. This method has been widely used in mobile devices and security systems[27]. However, facial recognition can be affected by factors such as lighting and facial expressions, leading to false positives and false negatives[18].

A biometric technique called eye tracking uses a user's eye movements to verify their identity. This technique can be applied to determine whether a user has left their device unattended or to check that they are paying attention throughout a session[12]. But this approach may be impacted by things like poor lighting and eyesight problems[7].

Heart rate monitoring is a biometric technique that measures a user's heart rate with the use of sensors and uses the information to confirm their identification. If a user has left their device unattended, this technique can be used to determine it[3]. However, elements like exercise and stress may have an impact on this technique.

3 Limitations

A major limitation of continuous authentication is the potential of false positives and false negatives. Either case can lead to significant consequences which at best might impede productivity and at worst leave protected systems vulnerable to unauthorized access. Additionally, continuous authentication also raises implementational issues, particularly when employing low-power devices with limited computational capabilities, like an office computer. If a device does not have the computational power necessary for authentication, sending data to a remote server might be a solution; however, doing so creates other issues like latency and privacy concerns. Additionally, unanticipated changes in the user data, such as an injury or environmental factors, can cause the models used for authentication to be unreliable, which could lead to inaccurate classification.

However, to address these limitations, mouse dynamics can utilize a machine learning approach to analyze and model user behavior over time. By doing so, models trained on user data can readily identify deviations from their baseline behavior and report suspicious activity easier than other approaches to continuous authentication. Moreover, a user's mouse dynamics can be captured in real-time without any additional hardware or software dependencies. Furthermore, this approach also comes with the added benefit of not capturing any sensitive information which makes it less obtrusive to user privacy.

4 Research Question

Further research needs to be done to explore the viability of mouse dynamics as a continuous authentication method and to assess its effectiveness in user identification and verification. It is still unclear if mouse dynamics models are flexible enough to accurately identify users based on their general mouse usage or if these models identify users in specific applications. Furthermore, additional research could be done to identify and mitigate potential environmental factors that could influence model accuracy.

5 Conclusion

In conclusion, this paper examined the current state of research on continuous user authentication and purposes that mouse dynamics solves many of the limitations of other authentication techniques. As the literature suggests, using mouse dynamics for continuous authentication has shown promising results in accurately distinguishing between legitimate users and imposters. Furthermore, it remains a secure and unobtrusive form of authentication which can be readily implemented in a plethora of devices which natively support mouse functionality.

References

[1] Ahmed Fraz Baig and Sigurd Eskeland. "Security, privacy, and usability in continuous authentication: A survey". In: *Sensors* 21.17 (2021), p. 5967.

[2] Long Chen et al. "Continuous authentication based on user interaction behavior". In: *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE. 2019, pp. 1–6.

[3] William Cheung and Sudip Vhaduri. "Continuous authentication of wearable device users from heart rate, gait, and breathing data". In: *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE. 2020, pp. 587–592.

[4] Gabriel Dahia, Leone Jesus, and Mauricio Pamplona Segundo. "Continuous authentication using biometrics: An advanced review". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10.4 (2020), e1365.

[5] Zach DeRidder et al. "Continuous User Authentication Using Machine Learning and Multi-Finger Mobile Touch Dynamics with a Novel Dataset". In: *2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE. 2022, pp. 42–46.

[6] Mario Frank et al. "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication". In: *IEEE transactions on information forensics and security* 8.1 (2012), pp. 136–148.

[7] Robert JK Jacob and Keith S Karn. "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises". In: *The mind's eye*. Elsevier, 2003, pp. 573–605.

[8] Zach Jorgensen and Ting Yu. "On mouse dynamics as a behavioral biometric for authentication". In: *Proceedings of the 6th ACM symposium on information, computer and communications security*. 2011, pp. 476–482.

[9] Pilsung Kang and Sungzoon Cho. "Keystroke dynamics-based user authentication using long and free text strings from various input devices". In: *Information Sciences* 308 (2015), pp. 72–93.

[10] Jianwei Li, Han-Chih Chang, and Mark Stamp. "Free-text keystroke dynamics for user authentication". In: *Artificial Intelligence for Cybersecurity*. Springer, 2022, pp. 357–380.

[11] Xiaofeng Lu et al. "Continuous authentication by free-text keystroke based on CNN and RNN". In: *Computers & Security* 96 (2020), p. 101861.

[12] Päivi Majaranta and Andreas Bulling. "Eye tracking and eye-based human–computer interaction". In: *Advances in physiological computing*. Springer, 2014, pp. 39–65.

[13] Soumik Mondal and Patrick Bours. "Combining keystroke and mouse dynamics for continuous user authentication and identification". In: *2016 IEEE international conference on identity, security and behavior analysis (ISBA)*. IEEE. 2016, pp. 1–8.

[14] Soumik Mondal and Patrick Bours. “Continuous authentication in a real world settings”. In: *2015 eighth international conference on advances in pattern recognition (ICAPR)*. IEEE. 2015, pp. 1–6.

[15] Aythami Morales et al. “Keystroke biometrics ongoing competition”. In: *IEEE access* 4 (2016), pp. 7736–7746.

[16] Narges Pourshahrokhi et al. “Generative adversarial attacks on motion-based continuous authentication schemes”. In: *2021 14th International Conference on Security of Information and Networks (SIN)*. Vol. 1. IEEE. 2021, pp. 1–6.

[17] Riseul Ryu et al. “Continuous multimodal biometric authentication schemes: a systematic review”. In: *IEEE Access* 9 (2021), pp. 34541–34557.

[18] Max Smith-Creasey, Fatema A Albalooshi, and Muttukrishnan Rajarajan. “Continuous face authentication scheme for mobile devices with tracking and liveness detection”. In: *Microprocessors and Microsystems* 63 (2018), pp. 147–157.

[19] Deian Stefan, Xiaokui Shu, and Danfeng Daphne Yao. “Robustness of keystroke-dynamics based biometrics against synthetic forgeries”. In: *computers & security* 31.1 (2012), pp. 109–121.

[20] Princy Ann Thomas and K Preetha Mathew. “An efficient optimized mouse and keystroke dynamics framework for continuous non-intrusive user authentication”. In: *Wireless Personal Communications* 124.1 (2022), pp. 401–422.

[21] Princy Ann Thomas and K Preetha Mathew. “A broad review on non-intrusive active user authentication in biometrics”. In: *Journal of Ambient Intelligence and Humanized Computing* 14.1 (2023), pp. 339–360.

[22] Hasan Can Volaka et al. “Towards continuous authentication on mobile phones using deep learning models”. In: *Procedia Computer Science* 155 (2019), pp. 177–184.

[23] Kanlun Wang and Lina Zhou. “What is more important for touch dynamics based mobile user authentication?” In: *PACIS 2020 Proceedings* (2020).

[24] Shuyu Wang et al. “Continuous Authentication Based on Keystroke and Mouse Dynamics in Video Private Network”. In: *2021 IEEE 9th International Conference on Information, Communication and Networks (ICICN)*. IEEE. 2021, pp. 377–383.

[25] Lu Xiaofeng, Zhang Shengfei, and Yi Shengwei. “Continuous authentication by free-text keystroke based on CNN plus RNN”. In: *Procedia computer science* 147 (2019), pp. 314–318.

[26] Heng Zhang et al. “Touch gesture-based active user authentication using dictionaries”. In: *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2015, pp. 207–214.

[27] Kaipeng Zhang et al. “Joint face detection and alignment using multitask cascaded convolutional networks”. In: *IEEE signal processing letters* 23.10 (2016), pp. 1499–1503.

An Alternative Programming Paradigm for Blocks-based Languages*

Huseyin Ergin[1], Jeff Gray[2] and Ryan Hunter[3]
[1]Department of Computer Science
Ball State University
Muncie, IN 47306
hergin@bsu.edu
[2]Department of Computer Science
University of Alabama
Tuscaloosa, AL 35487
gray@cs.ua.edu
[3]TechWise Academy
Muncie, IN 47303
ryan.hunter@bsu.edu

Abstract

Blocks-based languages are popular as an introductory programming language due to their ability to reduce the frustration of syntax errors within an ecosystem that supports recall over typing for core programming constructs. Although there are many variants of blocks-based languages, they often depend on the common basic building blocks of imperative programming: sequential/iterative/conditional programming concepts with some support for persistence of program state through variables or data structures. This paper describes our efforts to explore an alternative programming paradigm that merges the benefits of blocks-based languages with rule-based declarative transformations. As an alternative to the imperative style of programming, we aim to encourage the users of blocks-based languages to create a programming solution for a problem by using pre- and post-condition patterns that capture the transformation of program state in a visual manner. We experimentally evaluated a rule-based declarative prototype of the popular code.org coding puzzles called “Classic Maze” with middle school students. Although the students preferred the code.org approach from their background experiences, they also rated our new programming approach as interesting, fun, and different. The results suggest further investigation into this new rule-based paradigm for blocks-based programming languages.

1 Introduction

Blocks-based Programming Languages (BBPLs) are popular as an introduction to programming among the K-12 CS Education community [11]. Since the inception of Alice [3], many BBPLs have been created to aid young learners in their computer science journey (e.g., Scratch [9], SNAP! [6], and other variants built on top of these for other domains). In recent years, the CS advocacy group Code.org made BBPLs even more popular by organizing the Hour of Code activities worldwide. According to their webpage [2], there are more than 60 million students on their platform using many different BBPL environments created around different themes by the most prominent companies.

It is not surprising that BBPLs are gaining in popularity due to their ability to reduce the frustration of syntax errors. Learners are restricted from using the available blocks, and each block can only be connected with other compatible blocks. This assists in reducing syntactic errors in programming. There are many other benefits [1] to using BBPLs to teach basic programming principles. These benefits include reducing cognitive load by providing complex computational elements as blocks, enhancing understanding of the programs, and visualization of the program structure rather than entirely a textual representation.

Most BBPLs are built with the structures of imperative programming in mind. These are sequencing (step-by-step), selection (conditional), and iteration (loop). Part of the reason that most BBPLs use imperative programming is due to the popularity of the open-source blocks-based language engine called Blockly [5] by Google. Building a new blocks-based language from scratch with limited resources is challenging and costly. Therefore, the language creation framework provided by Blockly enforces the structure of the new BBPLs built on top of it.

In our paper, we dissect one of the most popular BBPL puzzles in code.org called “Classic Maze”[1] and replace the underlying conventional imperative programming approach with a declarative programming paradigm. Figure 1 is a

[1] https://studio.code.org/hoc/1

screenshot from the Classic Maze that represents the fundamental structures of a BBPL environment.

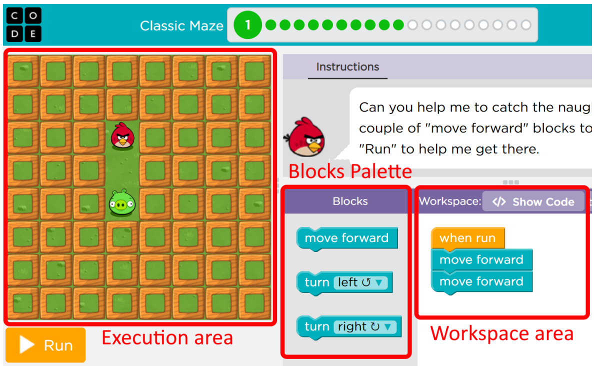


Figure 1: A Screenshot of an Hour of Code Puzzle in an imperative Blockly-based BBPL (screenshot from https://studio.code.org/hoc/1 on May 15th, 2023) ©code.org

In the Classic Maze, the goal is to move the bird to the same cell as the pig and complete the puzzle using the available blocks. A list of available blocks are provided in a blocks palette (Figure 1, middle) and they can be dragged/dropped into the workspace area (Figure 1, right) to create a solution to complete the task in the execution area (Figure 1, left). A learner can use iteration (loop) and selection (condition) in more advanced puzzle levels.

We have built a declarative version of the same code.org puzzle. In declarative programming, the program's state is updated with rules that consist of pre- and post-condition patterns of the state. In other words, the learners should think about the state of the program before and after executing a single rule in declarative programming rather than simply dragging/dropping an available block that changes the program state without knowing the details. Figure 2 displays the new environment we have built that is called Bird Command. In this new environment, the blocks palette only consist of essential elements in the execution area (i.e., the bird, the pig, the cell). The central construct to manipulate the execution area is a rule (turquoise rectangle in the blocks palette), which encodes the state of the execution area before and after this rule is executed. The learners are supposed to create as many rules as possible to achieve the goal in these puzzles (i.e., catching the pig by moving the bird). In both environments, the learner starts with a blank workspace and is only allowed to use the available blocks.

The rest of the paper is organized as follows. Section 2 provides more details

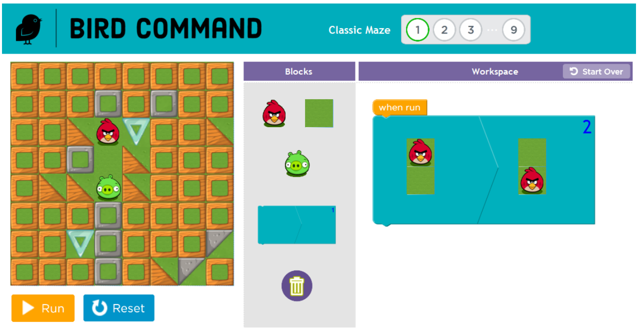


Figure 2: A Screenshot of the new Declarative Environment

about the new declarative rule-based alternative. Section 3 depicts the structure of an initial empirical study we have conducted on middle school students by letting them use both Classic Maze and Bird Command. In Section 4, we share the lessons learned from our implementation and the result of the study and create a vision for the future. Finally, we conclude in Section 5.

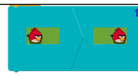
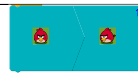
2 A Declarative Rule-based Alternative

This section discusses the details of the alternative we have created: Bird Command[2]. We have adopted a declarative approach for learners to specify solutions to the puzzles. In this approach, the learner specifies the result of a program by manipulating the program state directly rather than writing imperative commands. One way to write declarative code is through rule-based transformations. There are a few other efforts that use rewriting rules in visual programming environments [8, 10]. However, these efforts were investigated before BBPLs become mainstream. To the best of our knowledge, no studies combine declarative rule-based programming with BBPLs.

As in Classic Maze, a student begins their interaction with an empty workspace in our prototype. Figure 3 (left) depicts this workspace. The learner solves each puzzle by creating the rules and the patterns by dragging and dropping rules and assigning each rule a number to specify how many times it should execute (i.e., the count). In our prototype, the list of available blocks (Figure 3, "Blocks") is substantially different from the Classic Maze environment. The available blocks consist of the actors in the execution area (the bird, the pig, the cell) and a *rule* structure. Rule is the main container for any

[2]You can download the source code and prototype at `https://github.com/hergin/BirdCommand`

Table 1: Imperative (Classic Maze) vs Declarative (Bird Command)

Classic Maze	move forward	turn right ↻	turn left ↺
Bird Command			

pattern that can be modeled [4, 7]. A student's program in Bird Command is a list of rules, where each rule encodes a program state modification.

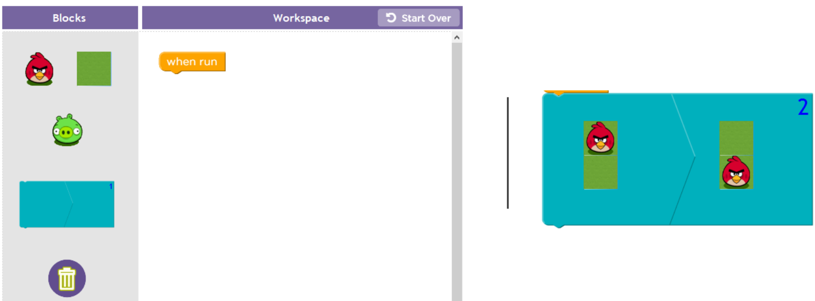


Figure 3: An empty workspace welcomes the learner (left), a sample rule (right)

Figure 3 (right) represents a rule in our prototype. A rule consists of pre- and post-condition (left and right of the arrow in the middle of the rectangle). Pre-condition is the state of the execution area before the rule is applied. For a rule to be applied, the pattern in the pre-condition should match a pattern in the execution area. Post-condition is the state of the execution area after the rule is applied. Our prototype simply tries to find a pattern that matches the pre-condition and replaces the match with the pattern in the post-condition. For example, the rule in Figure 3 moves the bird one cell down. By executing the rule as many times as specified (the number on the top-right), the maze in the execution area is updated.

The patterns that the learners can create with the available blocks are endless. We have also added the capability to turn the bird element in any direction. A learner can easily replicate all of the given blocks in Classic Maze with many more additional possibilities. Table 1 provides a mapping of the basic blocks provided in code.org's Classic Maze and the corresponding rule-based declarative versions that can be created in Bird Command.

In the remainder of this section, we provide solutions to the two of the puzzles we have shown to the learners in our empirical studies. These solutions provide an example of the differences between the traditional imperative paradigm and our new declarative paradigm. Then, we discuss the empirical study's details in the next section.

2.1 Maze 1

Maze 1 (https://studio.code.org/hoc/1, Figure 4, left) introduces the concepts to the learners and can be solved by just using two moveForward blocks (Figure 4, middle).

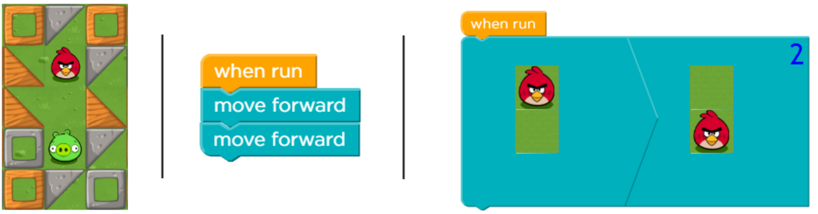


Figure 4: Maze 1 (left), Solution in Classic Maze (middle) and in Bird Command (right)

In Bird Command, learners have to create a pattern to move the bird forward and mark it to execute two times, i.e., changing the number on the top-right (Figure 4, right).

2.2 Maze 9

Maze 9 (https://studio.code.org/hoc/9, Figure 5, left) introduces the syntax for iteration (or repeat block in Classic Maze). The solution in Classic Maze (Figure 5, middle) considers putting smaller steps inside a repeat block.

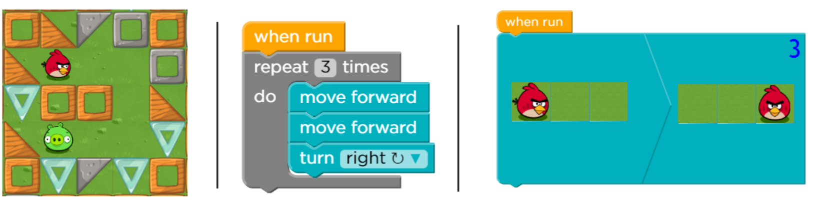


Figure 5: Maze 9 (left), Solution in Classic Maze (middle) and in Bird Command (right)

In Bird Command though, we do not have a repeat block. However, if the student comprehends the idea of pre- and post-condition patterns in a rule, they can easily create a pattern that can be repeated with the rule count (top-right). The patterns in Figure 5 (right) encode the same solution of classic maze (middle) inside the repeat block.

The rest of the mazes and their solutions are presented in Appendix A.

3 Empirical Study of Our Rule-Based Bird Command

In our initial empirical study, we did not try to answer large research questions or challenge the mature Classic Maze puzzles with our new programming paradigm prototype. Instead, we opted for a simpler study where we aim to gain helpful feedback from the students who have used the two approaches. Our one research question for this study is:

- RQ: What effect does a new programming paradigm have on blocks-based programming languages users who attend middle school?

Method We built four of the mazes of Classic Maze as a translation to Bird Command. We collaborated with a local programming/robotic event organization (TechWise Academy `https://techwiseacademy.com/`) that provided us with a list of students willing to participate in our study. The sequence of the empirical study is listed as: 1) The instructor reads the welcome letter that explains the purpose of the study. 2) Participants respond to the pre-survey[3]. 3) Participants complete the Classic Maze puzzles (numbers 1,2,3 and 9). 4) Participants complete the same puzzles in Bird Command. 5) Participants respond to the post-survey.

We conducted the study with two pilot groups first to check and adjust the study and completed two actual groups after the modifications. In the actual study, we switched steps 3 and 4 for the second group to balance the sequence bias. We also recorded the screens of the participants to understand how they solve the puzzles.

Demographics Two actual groups had a total of 16 students (7 and 9 students, respectively). The students are in sixth (9 students), seventh (4 students), and eighth (3 students) grades. 15 out of 16 have used a blocks-based language before, including Scratch, code.org, Blockly, and Snap. In addition, 6 out of 16 have used a textual programming language before, including Python, C#, and Java. Their career interests are heavily architecture & engineering, and science & technology.

[3] Survey questions are in the appendix B

Results 15 out of 16 students preferred to use the Classic Maze environment again. The reasons for selection include the interface of Classic Maze being easier, more fun, and familiar. The familiarity with Bird Command took more time and the prototype crashed for some students. Bird Command is rated 2.9 out of 5 in hardness (5 being the hardest), and 2.5 out of 5 in intuitiveness (5 being very intuitive). Participants also commented on designing the internal details of a block in Bird Command. The reactions include being "different but taking a long time," "being out of their comfort zone," "great but confusing," "interesting," and "fun."

Discussion To support answering our RQ, we also asked for free-form comments about the learners' feelings about designing the internals of a block in a rule-based system rather than just dragging/dropping a block in Classic Maze and not worrying about the internals. Initial reactions focused on the duration of solutions taking longer. This response was expected given the pattern design they were asked to make. Some students commented on the new environment being fun, different, and interesting. Some other students in our study suggested that the new rule-based alternative environment might be "good for the new people coming into coding so they can think about what the blocks are actually doing." Finally, many learners commented about their comfort zone when using Classic Maze because they were introduced to it in their CS experience before our study.

4 Lessons Learned

In this study, we investigated an alternative to the mainstream imperative programming that is taught as a de facto standard in CS education. We had to show both approaches to make a comparison and show the difference. This has produced its obstacles. In this section, we discuss some crucial lessons from the survey by combining them with our experiences, analyzing the learners' screen recordings, and putting a set of observations and suggestions together for potential future work in the area.

4.1 Changing the Mindset

In our first pilot study, we discovered that regardless of how intuitive the rule-based declarative approach is, it is hard to change the mindset of the learners. The students in the pilot could not solve the puzzles by just learning the new paradigm on the board when asked to solve the puzzle without any initial help. Therefore, in our second pilot study, we introduced a board game as an unplugged activity to better teach the rule-based declarative paradigm. Even though this caught the learners' attention, the unplugged activity increased the time needed to run the study. Finally, for this first exploration of an alternative paradigm, we decided to go through the first puzzle with students and leave the rest of the puzzles to them. This made the transition to a new paradigm easier and helped collect their thoughts on the practical use of the new paradigm rather than their struggle with a new topic.

4.2 Exposure to Code.org (or Other BBPLs)

One of the repetitive comments about the rule-based paradigm is the students' exposure to code.org in their CS education before our study. According to the pre-survey data, only one of 16 students did not use code.org or Scratch before. Although this one student did not use BBPLs before, in the post-survey data, he commented on finding the Classic Maze easier because of his "familiarity with that type of code." Empirical studies similar to ours may not be fair under the popularity of code.org environments and imperative programming. This popularity makes it hard to find an audience not exposed already to code.org (or other imperative BBPLs) before. In the future, an empirical study should be repeated with an audience that is not exposed to imperative BBPLs to have a fair and more accurate comparison. Another option is to repeat the study on a relatively less popular topic than Classic Maze, such as chemical reactions in a science class. This may level the playing field for the paradigms in these studies as code.org (or imperative paradigms) has not become popular while teaching these kinds of topics. Additionally, we can do the study with elementary school students where BBPLs and CS education has barely started.

4.3 Participants Who Got the Idea

After analyzing the videos and reading comments from a few students, we found that students who understood the core idea of the rule-based paradigm quickly completed all of the puzzles without difficulty.

Furthermore, some students created the exact maze pattern in the rule and completed the puzzle in just one rule by skipping the micro-steps needed. For example, Figure 6 shows the maze of the puzzle on the left and the solution that a student created on the right. Most students solved the maze by using micro-steps as in Figure 5 (right). These students got the idea very well and knew what is needed in the bare minimum to solve the puzzle in Bird Command without dealing with micro-steps.

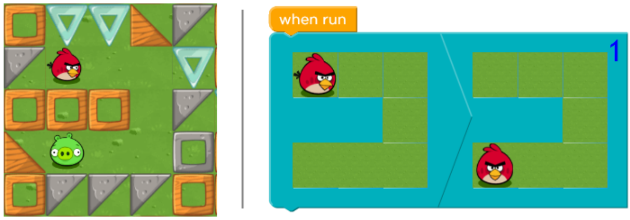


Figure 6: The maze in the puzzle and alternative solution of Maze 9 in Bird Command

4.4 Limitations of the Prototype

Our rule-based paradigm was introduced to the students as a prototype. Therefore, there were some limitations and usability issues. These issues did not prevent most students from completing tasks; however, the students noted these issues and other errors in the prototype in the post-survey.

In the future, the rule-based paradigm may be used as a standalone and not compared with anything to avoid competing with a professionally developed tool such as Classic Maze. However, since the study made the students complete the puzzles in both paradigms, students expect the prototype to be of the same quality as the Classic Maze regardless of being specifically told about the tool's being a prototype.

4.5 Alternative Rules

Alternative rules can do the same task in the rule-based paradigm. We discussed the possible move forward rules in the solution of Maze 3 in Appendix A.2. In our video analysis, we could not find any evidence that shows the students' using the alternative rules to complete a task. Instead, they created the patterns from whatever they saw in the maze. For example, in Figure 8 (right), the solutions of the students had two variations of move forward rules and these rules were created based on the direction of the bird. "Can the students think independent from the direction the bird is facing and create a generic direction-independent move forward rules?" remains a point to investigate in a future study. This can be analyzed more in-depth as a metric to measure students' comprehension of the new paradigm.

4.6 Reaching More Learners

We believe this alternative paradigm may expand the target population for computational thinking and open the door for a set of new under-served populations. For example, non-native English speakers might benefit from creating a pattern for a solution instead of reading the block names in Classic Maze. They can recognize the patterns and what they do to create new solutions to new puzzles. In the future, we will include more students of diverse backgrounds to measure this effect.

5 Conclusion

In this paper, we introduced a rule-based declarative programming paradigm as an alternative to the mainstream and commonly used imperative programming in BBPLs. We created a similar replica of one of the most popular imperative environments and its puzzle (Classic Maze). We conducted an empirical study with middle school students to experimentally evaluate the new programming paradigm. Although the students preferred Classic Maze from their background experiences, they also rated our new paradigm as “interesting,” “fun,” and “different.” We believe the lessons we have learned and shared will guide further studies that introduce other alternative programming paradigms in the area. The results look promising for further investigation into this new rule-based alternative paradigm for BBPLs.

References

[1] David Bau et al. “Learnable programming: Blocks and beyond”. In: *Communications of the ACM* 60.6 (June 2017), pp. 72–80. ISSN: 15577317. DOI: `10.1145/3015455`. arXiv: `1705.09413`. URL: `https://dl.acm.org/doi/10.1145/3015455`.

[2] Code.org. *Hour of Code*. 2020. URL: `https://hourofcode.com` (visited on 05/15/2020).

[3] Stephen Cooper, Wanda Dann, and Randy Pausch. “Teaching Objects-First in Introductory Computer Science”. In: *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*. SIGCSE ’03. Reno, Navada, USA: Association for Computing Machinery, 2003, pp. 191–195. ISBN: 158113648X. DOI: `10.1145/611892.611966`. URL: `https://doi.org/10.1145/611892.611966`.

[4] Krzysztof Czarnecki and Simon Helsen. “Feature-based survey of model transformation approaches”. In: *IBM Systems Journal* 45.3 (2006), pp. 621–645. ISSN: 00188670. DOI: `10.1147/sj.453.0621`.

[5] Google. *Blockly*. 2020. URL: `https://developers.google.com/blockly` (visited on 05/15/2020).

[6] Brian Harvey et al. "SNAP! (Build Your Own Blocks) (Abstract Only)". In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. SIGCSE '13. Denver, Colorado, USA: Association for Computing Machinery, 2013, p. 759. ISBN: 9781450318686. DOI: 10.1145/2445196.2445507. URL: https://doi.org/10.1145/2445196.2445507.

[7] Nafiseh Kahani et al. "Survey and classification of model transformation tools". In: *Journal of Software and Systems Modeling* 18.4 (Aug. 2019), pp. 2361–2397. ISSN: 16191374. DOI: 10.1007/s10270-018-0665-6. URL: http://link.springer.com/10.1007/s10270-018-0665-6.

[8] Alexander Repenning and Tamara Sumner. "Agentsheets: A medium for creating domain-oriented visual languages". In: *Computer* 28.3 (Mar. 1995), pp. 17–25. ISSN: 00189162. DOI: 10.1109/2.366152.

[9] Mitchel Resnick et al. "Scratch: Programming for all". In: *Communications of the ACM* 52.11 (Nov. 2009), pp. 60–67. ISSN: 00010782. DOI: 10.1145/1592761.1592779. URL: https://dl.acm.org/doi/10.1145/1592761.1592779.

[10] David Canfield Smith, Allen Cypher, and Kurt Schmucker. "Making programming easier for children". In: *Interactions* 3.5 (Sept. 1996), pp. 58–67. ISSN: 10725520. DOI: 10.1145/234757.234764. URL: http://portal.acm.org/citation.cfm?doid=234757.234764.

[11] David Weintrop. "Block-Based Programming in Computer Science Education". In: *Commun. ACM* 62.8 (July 2019), pp. 22–25. ISSN: 0001-0782. DOI: 10.1145/3341221. URL: https://doi.org/10.1145/3341221.

Appendix A Rest of the Mazes

A.1 Maze 2

Maze 2 (https://studio.code.org/hoc/2, Figure 7, left) is another practice run before the introduction of other blocks. The student must use the moveForward block three times (Figure 7, middle).

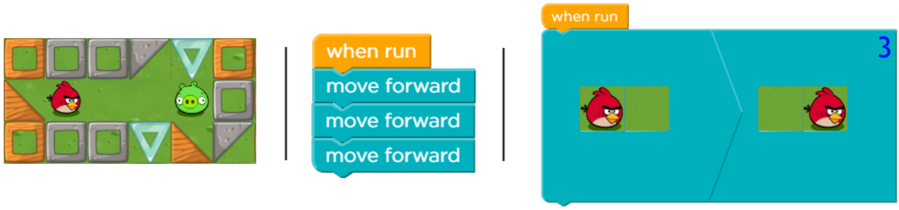


Figure 7: Maze 2 (left), Solution in Classic Maze (middle) and in Bird Command (right)

The same case happens in the Bird Command. One difference in Bird command solutions between Maze 1 and Maze 2 is the direction of the rule patterns moving the bird forward. The rule in Figure 7 (right) is semantically the same as the rule in Figure 4 (right) (except the count). Specifically, the bird moves one block in the direction it is facing.

A.2 Maze 3

Maze 3 (https://studio.code.org/hoc/3, Figure 8, left) enforces the use of new blocks to solve the puzzle, such as turning left or right (Figure 8, middle).

In Bird Command, any kind of turn rules can be created easily. For example, for turn right, the pre-condition has the bird facing in one direction, and the post-condition changes the bird's direction to the next clockwise (Second rule in Figure 8, right) direction.

Another interesting thing to mention in this solution is the use of different move forward rules in the solution's first and third parts. One can naturally model the forward-moving rules like in Figure 8 by looking at the puzzle at the execution area (Figure 8, left). The bird starts the puzzle facing right; hence, the first move forward rule in Figure 8 moves the bird right. However, after the bird turns right in the corner, it then faces south (or down). Therefore, the second move forward rule (the last rule in Figure 8, right) moves the bird down. However, these two move forward rules are semantically same.

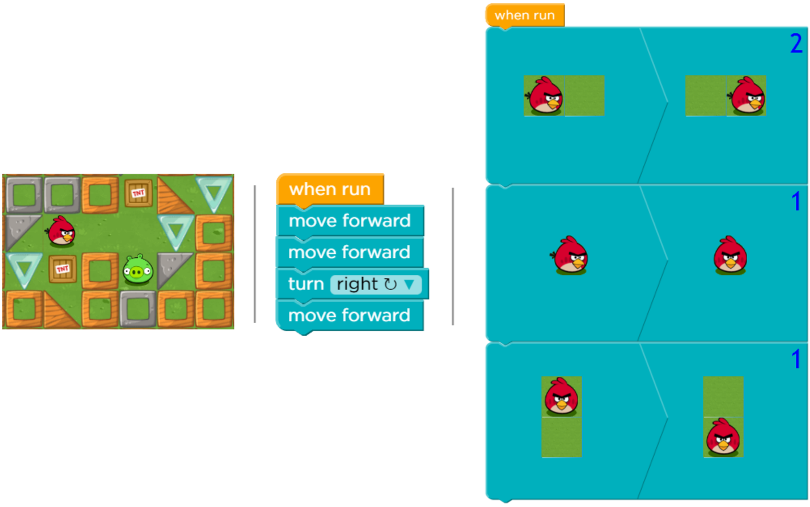


Figure 8: Maze 3 (left), Solution in Classic Maze (middle) and in Bird Command (right)

Appendix B Survey Questions

In the pre-survey, we asked the following questions to understand the participant demographics.

- What is your school grade?
- Have you ever used a blocks-based language before, such as code.org platforms or Scratch?
- Have you ever used a textual programming language before, such as Java, JavaScript, Python.
- What career would you like to pursue as an adult?

In the post-survey, we asked the following questions to gather feedback from the study.

- Which tool do you prefer to use again? Why?
- How hard was Bird Command to program?
- How intuitive were the icons and toolbars?
- In Classic Maze, you did not focus on the internals of the moveForward block. However, in Bird Command, you had to think about the internals and how to move the bird forward by modeling the pattern. How do you feel about designing the details of the blocks in Bird Command?
- Do you have any other comments?

Generating Functions Without Toil*

Christopher A. Healy
Department of Computer Science
Furman University
Greenville, SC 29613
chris.healy@furman.edu

Abstract

This paper advocates for including the topic of generating functions in discrete mathematics classes. Generating functions provide an additional way to solve many combinatorical problems. The approach is robust, and can easily handle complex problems that would be infeasible to solve using traditional formulas. Available online software takes the tedium out of obtaining numerical answers from a generating function. The author has taught discrete mathematics seven times with a short unit on generating functions, and exam results show that students learn and retain this new topic at least as well as the rest of the course.

1 Introduction

A course in discrete mathematics or "discrete structures" is an important part of the computer science (CS) curriculum. It is here that students learn many of the quantitative and rigorous techniques that upper-level CS courses rely upon. Discrete mathematics has wide-ranging applications for computing, such as logic, proof, combinatorics, graph theory, functions, and relations. This paper assumes the usual place of the discrete mathematics course in the lower division undergraduate level, intended to be taken alongside or shortly before the second computer programming course.

Within discrete mathematics, one of the central topics is combinatorics, or "counting." Students learn techniques for solving various types of counting problems, such as permutations and combinations. However, one technique that is rarely seen in discrete mathematics textbooks today is the subject of using generating functions to solve combinatorical problems. This is unfortunate, because generating functions are an elegant and robust method to encode many kinds of combination problems. For the purpose of solving these problems, a generating function is simply a polynomial, in which the parameters of the problem are encoded as the exponenets, while the numerical answer(s) are found in the coefficients.

A generating function may be defined [6] as "a formal power series

$$f(x) = \sum_{n=0}^{\infty} a_n x^n$$

whose coefficients give the sequence $a_0, a_1, \ldots$." In other words, a generating function is a polynomial used to encode a sequence of numbers. Note that the word "generating" here is an adjective, not a verb or participle. Although they have several applications to discrete mathematics, generating functions are not mentioned in any of the ACM recommended CS curricula from 1965, 1968, 1978, 1991, or 2001. However, the subject is mentioned in passing in the 2013 curriculum in one of the model syllabi for an algorithms class, but only as a means of solving recurrence relations [2].

The reason why generating functions have not been a popular topic in discrete mathematics is likely because of the considerable effort needed to compute the necessary coefficients. To calculate the coefficents manually requires considerable facility with infinite series, as well as the use of the binomial theorem with negative exponents. In practice, manual computation of the coefficients of a generating function is quite tedious. For example, the answer to a combinatorical problem may require us to obtain the coefficient of x^4 in the expansion of $(1+x+x^2+x^3+x^4+x^5)^3$. Multiplying out the whole polynomial is not a sustainable strategy, because another similar problem may involve larger exponents. It is also unnecessary, because we only need one of the coefficients, not all of them. Traditionally, students would learn *ad hoc* algebraic techniques to determine coefficients. In this example, the generating function could be rewritten as $(1-x^6)^3(1-x)^{-3}$. The binomial theorem would be applied to the factors $(1-x^6)^3$ and $(1-x)^{-3}$ separately. Finally, we would examine these two polynomial expansions to logically deduce how the coefficient of x^4 is formed. Anecdotally, tedious procedures like this turn students off generating functions, so it is no wonder the topic has been omitted.

One popular discrete mathematics textbook today is by Rosen [8]. This thorough text does contain a 15-page section on generating functions, but it is relegated to a chapter entitled "advanced counting techniques," and only five pages are devoted to showing how generating functions can be used to solve counting problems.

Fortunately, it is not necessary to carry out a laborious manual computation. We have the benefit of computer algebra systems such as Mathematica and MAPLE to simplify the computation. Nevertheless, to the author's knowledge, textbooks have not made use of this technological advance. This author has taught generating functions successfully for the last seven offerings of the course. And thanks to free websites such as Wolfram Alpha, students do not even need specialized software such as Mathematica. As a result, what used to be one of the most intimidating topics in discrete mathematics is now one of the most straightforward.

2 Related Work

Other educators have reported on the usefulness of computer algebra systems in CS courses. Two types of courses mentioned in the literature are in the areas of algorithms and discrete mathematics.

The use of computer algebra systems in algorithms classes has been noted by [1, 3, 4]. Harris [3] used MAPLE to streamline the teaching of generating functions for solving recurrence relations in a class on the analysis of algorithms. Koeller and Hubey [4] also used MAPLE, this time in a course on data structures and algorithms, to observe the growth of functions, including those having two input variables. Finally, Byrd et al. [1] created their own computer algebra system to solve recurrence relations in an algorithms course.

Computer algebra systems in discrete mathematics have been reported by [5, 9]. Marion [5] discussed the use of MAPLE to observe the growth of functions in order to visualize a Big-O relationship, to solve recurrence relations, to assist in working out induction proofs, and to implement the RSA algorithm. Schoenefeld and Wainwright [9] discussed the integration of Mathematica into a discrete mathematics course. Combinatorics was listed in the outline of course topics, but there was no mention of generating functions.

Of these five studies, only Harris [3] combined generating functions with a computer algebra system. In particular, none of these works mentioned the use of generating functions to solve combinatorical problems. The novelty of this paper is using a system such as Mathematica to solve combinatorical problems that require generating functions, in a discrete mathematics course accessible to CS students early in their curriculum.

3 Usefulness of Generating Functions

Two of the most common types of counting problems covered in discrete mathematics are combination problems and so-called "ball-in-urn" problems. A combination problem asks for the number of ways to select r elements out of a set of n distinct objects. Meanwhile, with a ball-in-urn problem, we have a collection of n identical objects, and we wish to find the number of ways to divide it into r partitions. A key difference between these two types of problems is that in combinations, the n objects are all distinct, but in a ball-in-urn problem, the n objects are identical.

This immediately raises a question: What if some of the n objects are distinct, but others are identical? In other words, how do we solve a combination problem if the objects are selected, not from a set, but from a multiset [7]? Consider this problem: We have a list of 24 numbers: (1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 25, 50, 75, 100). How many ways are there to select six numbers from this list? Because the list is not a set, we cannot use combinations. Had the 24 numbers all been distinct, the answer would have been $C(24, 6)$ = 134,596. However, the correct answer is much less: 13,243. This seems like the type of problem we want our students to be able to solve with confidence. It seems silly to tell the class that this question is beyond the scope of the course, or that it needs to be divided into many cases.

4 Problem Solving for Combinations

The motivation behind using generating functions to solve combination-type problems can begin by considering the following expansion to model the subsets of the set {a, b, c}:

$$(1 + ax)(1 + bx)(1 + cx) = 1 + (a + b + c)x + (ab + bc + ac)x^2 + (abc)x^3$$

The factor $(1 + ax)$ signifies that we may select the letter a or not. The other factors $(1+bx)$ and $(1+cx)$ similarly deal with possibly selecting the elements b and/or c. The right side of the equation itemizes all of the possible outcomes. There are four terms, corresponding to the size of our selection. The first term 1 (*i.e.* x^0) contains no letter coefficients, so this corresponds to the empty set: There is one way to choose nothing. The second term shows all of the possible ways of selecting one element from {a, b, c}. The third term itemizes the possible outcomes of selecting two elements from {a, b, c}. And the final term shows the only option of selecting all three elements.

If we remove the letter coefficients of this example, we are left with

$$(1 + x)(1 + x)(1 + x) = (1 + x)^3 = 1 + 3x + 3x^2 + x^3$$

This generating function does not itemize the individual subsets, but rather counts them. The coefficients are now 1, 3, 3, and 1, indicating how many subsets of {a, b, c} of each size exist. Most combinatorical questions ask for the number of selections that can be made, rather than an itemization, so this type of generating function, *i.e.* without the letter coefficients, is more commonly useful. Therefore, the number $C(n, r)$ can be found by asking for the coefficient of x^r in the generating function $(1+x)^n$, which is the same result as using the binomial theorem. But in the language of generating functions, we would say the following: Each of the elements in {a, b, c} corresponds to a factor of $(1 + x)$ in the generating function, signifying whether we select it or not.

The next issue is how to deal with a multiset. For example, how would one distinguish between these two situations:

- Selecting two objects from the set { a, b, c, d }
- Selecting two objects from the list (a, a, b, b)

In the first case, we have four distinct elements a, b, c, and d. Each element corresponds to a factor of $(1+x)$. Therefore, the generating function is $(1+x)^4$. But the second case is a multiset. We only have two distinct elements, a and b, but we are able to choose each letter up to twice. Its generating function is $(1 + x + x^2)^2$. And to find out the number of ways that two objects can be taken, we seek the coefficient of x^2.

To return to the earlier example of a multiset with 24 numbers from which to select six, we would seek the coefficient of x^6 in this generating function:

$$(1 + x + x^2)^{10}(1 + x)^4$$

The generating function was determined as follows. In the multiset we have two copies of each of the numbers 1 through 10. Hence, we need ten factors of $(1 + x + x^2)$. Then, we have one each of the numbers 25, 50, 75, and 100, which means we want four factors of $(1 + x)$.

With this basic knowledge, students can begin to formulate solutions to combinatorical problems using generating functions. This entails writing down the generatng function and specifying which terms coefficient contains the numerical answer to the problem. The student then consults Mathematica or Wolfram Alpha with this query. Wolfram Alpha is a freely available website, and the author frequently uses it in class. It accepts a natural language query such as: `coefficient of x^3 in (1 + x)^10`, or by using the built-in `Coefficient` function as in: `Coefficient[(1 + x)^10, x^3]`. A computer algebra system can also expand the generating function in order to obtain all of the coefficients, if desired.

5 Generating Functions of Two Variables

It is possible for a generating function to use two variables rather than one. This is useful when we need to keep track of two quantities when making a selection of numbers. Fortunately, the formulation of two-variable generating functions is also straightforward. As a simple example, consider the multiset (0, 1, 2, 2, 3, 4). We may wish to make a selection of r numbers from this multiset having a sum of s. We can let the exponent on x count how many numbers are being selected, while the exponent on y will tally the sum of the chosen numbers. As before, each distinct value in the multiset corresponds to one factor in the generating function. The generating function corresponding to this multiset is

$$(1+x)(1+xy)(1+xy^2+x^2y^4)(1+xy^3)(1+xy^4)$$

When expanded, one of the terms in this generating function is $3x^2y^4$. This tells us that there are three ways to select two numbers having a sum of four.

Generating functions of more than two variables are possible, but in the authors experience, it is not necessary to delve deeply in this direction in this course. Problems of several variables can be dealt with using techniques of linear algebra.

6 Ball-in-urn Problems

Another type of combinatorical problem that generating functions can model equally well is the so-called ball-in-urn problem. We have n identical balls to distribute into r distinct urns. The urns are distinct, so each urn corresponds to a factor of the generating function. We may choose any number of balls (up to n) to place in each urn. Therefore, the generating function becomes $(1+x+x^2+...+x^n)^r$. Since we have a total of n balls to place, we desire the coefficient of x^n. For example, the number of ways to place 12 balls in four urns is the coefficient of x^{12} in $(1+x+x^2+...+x^{12})^4$. Although this approach gives the correct answer, it is more straightforward to use the standard ball-in-urn formula, $C(n+r-1, r-1) = C(15, 3)$.

The elegance of the generating function approach becomes apparent when restrictions are added to the problem. We can modify the above example as follows. Among the four urns, let us suppose that the first urn can take no more than two balls, the second urn must take at least one ball, and the last two urns must each take an odd number of balls. It is straightforward to modify the original generating function to become

$$(1+x+x^2)(x+x^2+x^3+...+x^{12})(x+x^3+x^5+...+x^{11})^2$$

from which we desire the coefficient of x^{12}. Once again, we have a separate factor for each urn. The first factor allows for up to two balls. The second factor (for the second urn) allows for 1-12 balls. The third and fourth factors allow only for an odd number of balls, exactly as the problem requires. In fact, an astute observer will notice that this generating function can be simplified. Some unnecessary terms can be removed because certain combinations of selections are impossible. The last three urns must each have one ball. Therefore, the second urn cannot contain more than ten balls. By the same token, the last two urns cannot have more than nine. The generating function becomes:

$$(1 + x + x^2)(x + x^2 + x^3 + ... + x^{10})(x + x^3 + x^5 + ... + x^9)^2$$

However, in practice it is not necessary to perform this optimization, since it only affects terms having exponents higher than x^{12}, the term that we desire.

What is especially convenient about using generating functions is that impossible cases do not need to be subtracted away. As another example, let us suppose we wanted to find how many six-digit numbers have a sum of 12. This can be modeled as a ball-in-urn problem using the combination formula $C(n+r-1, r-1)$. But we have to subtract cases in which any urn takes more than nine balls. If we were seeking a digit sum of 20 or higher, the overlapping restrictions make it necessary to apply the inclusion-exclusion principle in order to arrive at the correct answer. Assuming that all six digits can be 0-9, the generating function to solve this problem is $(1 + x + x^2 + ... + x^9)^6$. Here, the impossible cases never arise because the design of the generating function avoids having any digit exceed nine.

If we had to rely on the standard combination formula to solve a ball-in-urn problem with various restrictions, we would be forced to break the problem up into many separate cases. Using generating functions is a more expeditious and robust approach. And once again, a computer algebra system can swiftly report the numerical answer.

7 Pedagogy

In the authors class, students are taught the traditional approaches to solving permutation and combination problems immediately before the unit on generating functions. To start with, students need to realize that generating functions are simply polynomials. However, polynomials used for generating functions are slightly different from those seen in ordinary algebra. First, it is customary to write the terms with ascending powers of x. For example, we write $1 + x^2$ rather than $x^2 + 1$. One reason why is that some generating functions have an infinite number of terms, and it is more intuitive to write the ellipsis on the right end of the polynomial rather than the left. The second novelty of generating functions is that the value of x is immaterial. The polynomial is being used merely as an algebraic structure to store information about a conbinatorial problem. The idea of storing information inside some structure is a familiar concept that CS students can appreciate. They can perceive generating functions as a class possessing certain attributes and operations.

The unit on generating functions can be completed within one week. The units topics include:

- Selections from a set and multiset. Ball-in-urn problems.
- Two-variable problems.
- Using letter coefficients to enumerate selections.
- Multiset problems containing a restriction.
- Turning the problem around: Given a generating function, pose a question to which the answer is found by one of its coefficients.
- Coin problems such as: Given 15 dimes, 20 nickels, and 25 pennies, what is the number of ways to make a dollar of change?
- Dice problems such as: How many ways can we roll a sum of 15 from four dice? What is the probability that a sum of 15 will appear? Also, rolling dice having different numbers of sides.
- Digit problems such as: How many six-digit numbers have a sum of digits of 29? We also write a short program to loop through all six-digit numbers to enumerate every possible sum and observe that the answers match the coefficients of the generating function.

Within the discrete mathematics course, the other major application that makes use of generating functions is in solving recurrence relations [6]. The author decided not to use generating functions in the recurrence relations unit mainly due to lack of time, and also because it would be more difficult than solving combinatorical problems.

8 Results

The author has taught a discrete mathematics course seven times incorporating a short unit on generating functions. Two or three tests are given during the term, plus a final examination. Questions on generating functions appear on one of the tests as well as on the final examination. The students' performance on these questions can be compared to their overall exam performance to gauge how well the students have mastered the generating function concept.

Table 1 shows the results of tests given during the seven offerings of the course. The third column of the table measures the average score earned by students on the generating function questions on the test. The fourth column shows the average score earned by students on the test that contained the

Table 1: Student Performance on Generating Function Questions

Section	# students	During-term averages GF questions	During-term averages Overall	Final exam averages GF questions	Final exam averages Overall
1	12	73	77	74	66
2	8	82	80	77	71
3	15	72	78	84	72
4	14	80	79	85	79
5	12	86	82	90	65
6	14	81	78	71	76
7	19	82	81	88	75
Total	94	79	79	82	74

generating function questions. Similarly, the fifth column shows the avearge score earned on the generating function questions on the final exam, while the last column shows the average score on the final exam. The results show that the students perform at least as well on the generating function questions as the rest of the test material. The results on the final exam are particularly striking. They show that the students had better long-term retention on the generating function concepts than they did for the other topics asked on the final exam. The idea that generating functions are too difficult for a first course in discrete mathematics is a myth.

9 Conclusion

This paper has described how it is possible to offer an accessible introduction to generating functions in a discrete mathematics course. In one week, students learn how to solve various combinatorical problems with this technique. Familiar combination problems can be encoded in this fashion. Generating functions are easily scalable to allow students to solve complex problems that the standard *ad hoc* formulas cannot.

The intent is to teach generating functions in a way that students appreciate their utility. It is far more important to know conceptually what a generating function is, how to formulate one to solve a combinatorical problem, and then use off-the-shelf software to handle the rest. By de-emphasizing the manual computation needed to obtain numerical answers, considerable class time is saved.

This approach has been used in the authors class for the last seven offerings, and students learn this unit at least as well as the other topics of the course. For students who wish to learn more, an upper-division course in combinatorics is available that can be taken as a CS elective. There, students receive a more detailed treatment of generating functions, just as with other topics that were introduced in the first discrete mathematics course.

References

[1] Jonathan Byrd et al. "Solving Recurring Student Problems with Recurrences". In: *ACM SIGCSE Bulletin* 39.4 (2007), p. 343.

[2] Joint Task Force on Computing Curricula. *Computing Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. `https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf`.

[3] James Harris. "Using Computer Algebra Systems in the Teaching of Analysis of Recursive Functions". In: *Journal of Computing Sciences in Colleges* 21.5 (2006), pp. 116–122.

[4] Andreas Koeller and H. Mark Hubey. "The Use of Computer Algebra Systems in Computer Science Education". In: *Journal of Computing Sciences in Colleges* 19.2 (2003), pp. 223–231.

[5] Bill Marion. "Computer Algebra Systems in Discrete Mathematics". In: *Journal of Computing Sciences in Colleges* 26.2 (2010), pp. 139–148.

[6] Wolfram MathWorld. *Generating Function*. `https://mathworld.wolfram.com/GeneratingFunction.html`.

[7] Wolfram MathWorld. *Multiset*. `https://mathworld.wolfram.com/Multiset.html`.

[8] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. New York: McGraw Hill, 2012.

[9] Dale Schoenefeld and Roger Wainwright. "Integration of Discrete Mathematics Topics into the Secondary Mathematics Curriculum using Mathematica – A Summer Institute for High School Teachers". In: *ACM SIGCSE Bulletin* 25.1 (1993), pp. 78–82.

Pilot Study of Deploying IoT Micro Air Quality Sensors in an Urban Environment: Lessons Learned*

Julia Gersey, Brian Krupp, Jonathon Fagert
Computer Science and Engineering Department
Baldwin Wallace University
Berea, OH 44145
{jgersey20,bkrupp,jfagert}@bw.edu

Abstract

In the United States, overall air quality has steadily improved since the passage of the Clean Air Act of 1970. However, not all communities have realized these gains and reported air quality can differ significantly between communities. One approach to improve awareness is to add additional regulatory sensors to provide more fine-grained sensing. However, regulatory sensors are expensive and may not be an affordable option. To provide fine-grained sensing and localized air quality, low-cost sensors that detect particulate matter have become an affordable option. Combined with an Internet of Things (IoT) platform, these sensors can provide real-time data to communities. This paper presents findings from a pilot deployment of an IoT air quality sensor network in Slavic Village within Cleveland, Ohio. Through this research, we share our findings and lessons learned from deploying these sensors in partnership with a community partner. From a pilot deployment, we are able to determine a significant difference between two locations that are less than 4 miles apart, supporting the need for fine-grained air quality monitoring. We discuss challenges in deploying in urban environments where both power and connectivity are constraints and how future work will reduce both of these constraints.

1 Introduction

Air quality is an important factor for human health. The World Health Organization (WHO) estimates that each year, 7 million people die prematurely due to ambient and indoor air pollution [10]. Whether or not an individual has any previous health conditions, the quality of the air they breathe can have a dramatic effect on their health. Particulate Matter 2.5 (PM 2.5) poses the greatest risk to humans, as these particles have a 2.5 micrometer diameter, allowing them to not only penetrate into human lungs, but into the bloodstream as well [2]. PM 2.5 contributes to premature death, heart attack, decreased lung function, and other health effects [1]. Long term exposure can lead to asthma and lung cancer, among other health conditions. Confirmed by epidemiological studies, humans can contract respiratory diseases more easily with PM 2.5 exposure[11].

The EPA provides the *AirNow* portal for communities to check current PM 2.5 levels [5]. However, in the area of this study which includes a major metropolitan area, one PM 2.5 sensor is shown to cover an area of more than 400 square miles. Other studies have shown that air quality can differ between different streets and based on the orientation of bus shelters [7, 3]. Deploying more regulatory sensors may not be an affordable option. However, with the advent of low-cost particulate matter sensors combined with an Internet of Things (IoT) platform, fine-grained air quality monitoring can become a reality. While these low-cost sensors may not be as accurate as a regulatory sensors [9], they can used to provide general trends to the community as well as identify pollution sources.

In this paper, we describe a pilot study of deploying low-cost sensors on an IoT platform. We utilize generally available single-board computers (Raspberry Pis) and common low-cost particulate matter sensors (Plantower PMS5003). We deployed these sensors in partnership with a community partner that provides low-cost internet access to low-income communities. Through this pilot deployment, we found a significant difference in air quality in a distance of less than 4 miles and differences between reported data from regulatory sensors and the sensors we deployed. The rest of this paper is organized as follows: Section 2 discusses the related work in this area, Section 3 describes the methodology for this study including sensor design and deployment, Section 4 includes findings from the evaluation of our study, and Section 5 summarizes the study and discusses future work.

2 Related Work

Building and deploying an IoT platform is the first step for being able to monitor air quality in urban environments. Frequent monitoring of urban environments in EU countries is now regulated, and one study found that using high-quality sensors may not provide data with an appropriate spatial and temporal resolution, and that instead a wireless sensor network with a large number of low-cost sensors is best for monitoring urban environments [6]. Besides keeping sensor costs down, factors such as low maintenance of both the sensors and infrastructure are important when aimed to be used in amateur or community projects [4]. It is important to consider all factors from both a software and hardware perspective.

When deploying a large network of sensors in an urban environment, it is critical to think of their placement when compared to environmental events (highways, steel mills, parks) so the data can accurately represent the urban environment. Regulation and previous research states that these sensors should be 1.6 meters, or 5.25 feet, above the ground to determine pollution exposures for adults [6]. Lamp-posts seem to provide an ideal infrastructure, as most contain the power and height requirements specific to air quality sensors[4]. By deploying a large network of IoT devices, it reduces human effort to collect, transfer, and log a large amount of data [8].

With the major causes of air pollution being industrialization, urbanization, and motorization, it is important to monitor outdoor air quality [8]. To accurately capture outdoor air quality, it requires housing that protects it from the elements, while still making sure enough air is getting passed to the sensor and that the sensor is still able to communicate its data to the network. For example, in a deployment of sensors in metal cages, the sensors are protected from the elements and possible human interference, but the metal enclosure can interfere with wireless connectivity [6].

3 Methodology

3.1 Sensor Design

The sensor modules designed for this study utilize generally available single-board computers (Raspberry Pi Model 3B+) and a Plantower PMS5003 particulate matter sensor, which uses light scattering to measure the quantity and diameter of particles. For this study, we gather PM 1.0, PM 2.5, and PM 10.0. The sensor software was written in Python where sensor readings are collected every 60 seconds and reported to a web service. Other models were developed using microcontrollers where the software was written in the C programming language. However, for this study, only the Raspberry Pi Model 3B+ units were used. A web service on a Linux, Apache, MySQL, PHP (LAMP) stack receives the sensor ID and PM readings and stores the data in a MySQL database. The sensors that were deployed in the field utilize Ethernet to connect to the internet. However in testing differences in enclosures in our lab, we utilized a Raspberry Pi Zero W, where we used WiFi connectivity. An image of the sensor module outside the enclosure is shown in Figure 1.

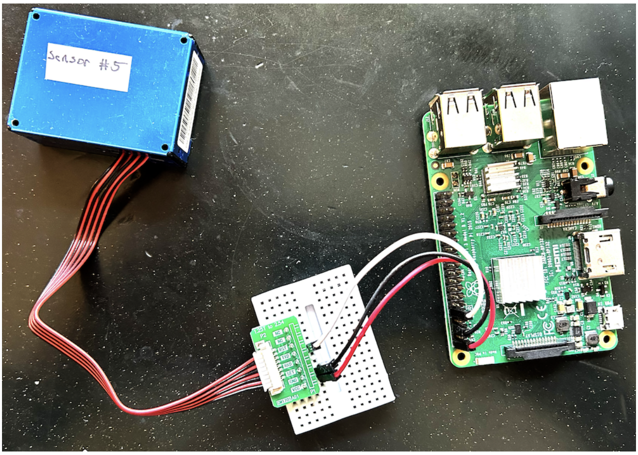


Figure 1: Sensor Module with PM 2.5 Sensor Connected to Raspberry Pi 3B+

The primary motivation in using generally available boards is to support a future goal in creating curriculum for middle and high school students where they can learn to build their own sensor module, create the software, and install it at their school. Within this curriculum, we aim for students to learn computer science and engineering topics that allows them to apply these concepts to an area that affects their community.

3.2 Local Partnership

Two common constraints in deploying IoT-enabled sensors in the field is supplying sufficient power and connectivity to the internet. Various approaches exist such as utilizing battery powered sensors, charging batteries via solar, or having fixed units that have a consistent source of power. With connectivity, popular options have included WiFi and cellular connectivity. To address these two constraints, we partnered with *PCs for People*, a non-profit organization that provides refurbished computers and WiFi hotspots to communities in need. Through this partnership, we were able to put our sensor modules in their towers, providing both power and connectivity. The location of *PCs* *for People's* towers was ideal for our pilot study due to their proximity to one of our neighborhoods of interest which is an area with close proximity to steel mills.

3.3 Web Dashboard

Beyond deploying the sensors and capturing the data, we wanted to provide an easy and intuitive way to view and interact with the data. To do this end, we created a web dashboard that shows the readings from each sensor in a line chart in real-time. The web dashboard also utilizes a background service to pull data from a nearby EPA regulatory sensor to compare the PM 2.5 data that was sensed with the regulatory sensor. To create the line chart, we use PlotlyJS, which allows us to pan and zoom into areas of interest and examine different trends. A screenshot of the dashboard is shown in Figure 2.

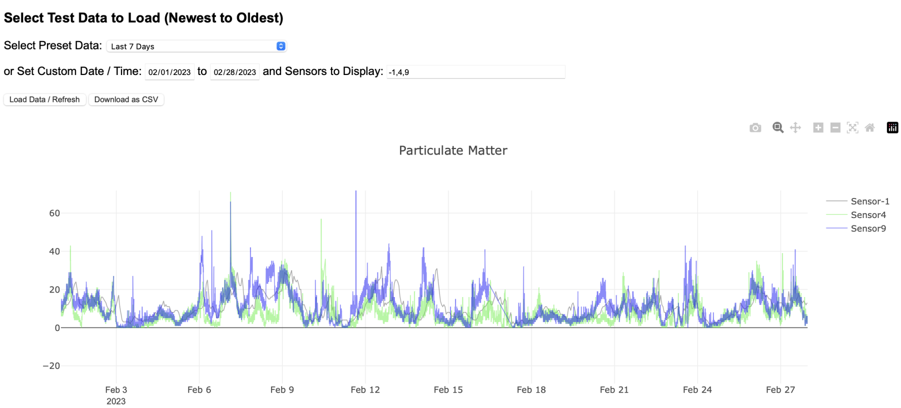


Figure 2: Screenshot of Web Dashboard Showing Sensor Data in Real-Time

4 Evaluation

4.1 Pilot Deployment

Our initial deployment involved having one sensor deployed in the office location of *PCs for People* from June 18th 2022 to August 13th 2022 and then at one of their cellular towers that provides internet access to low-income households from August 14th to September 12th 2022. There were two interesting observations from this deployment. One is that on the 4th of July at approximately the same time as a city fireworks show located west of the office location, the detected PM 2.5 level spiked to 125, as can be seen in Figure 3. The other interesting observation was that after moving the sensor to one of the cellular towers that was less than 4 miles south of the office location, the average PM 2.5 level doubled from 5.314 to 11.892. These averages include the entire readings from the deployment at both corresponding locations from June 18th to September 12th. Additionally, with the sensors placed near the major highways in Cleveland, the higher averages corresponded to rush hour times in the morning and evening. This pilot deployment supports the need for deploying more fine-grained air quality monitoring in this region as it showed a considerable difference in a distance of less than 4 miles.

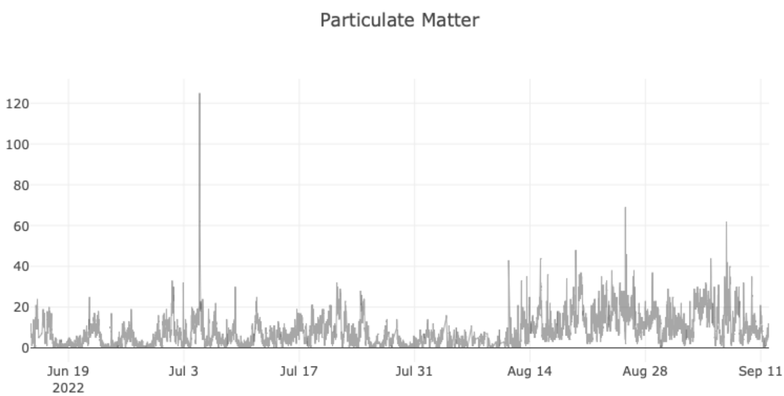


Figure 3: Initial Deployment of Air Quality Monitor in Two Separate Locations Less Than 4 Miles Apart

4.2 4 Month Deployment

An additional sensor was deployed in the region of the cellular tower 0.5 miles to the southeast. This location showed higher variability in the sensed particulate matter compared to the existing sensor where the new sensor had a standard deviation of 8.31 compared to the existing sensor of 6.14 and the regulatory sensor of 6.42. This difference is noticeable in observing the sensor data from the web dashboard where the sensors report higher peaks of PM 2.5 levels as shown in Figure 4.

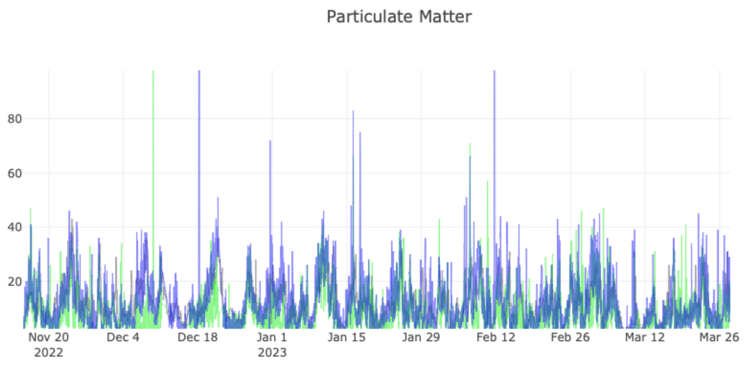


Figure 4: Deployment of Two Air Quality Monitors Over 4 Months

4.3 Enclosure

For situations where the *PCs for People* infrastructure isn't available (e.g., for future anticipated "edge nodes"), our sensing nodes will require their own enclosure to protect them from varying environmental conditions. The challenge with these enclosures is that they have competing requirements of preventing moisture (i.e., ran, snow, etc.) from entering, but allowing sufficient air flow to achieve high fidelity readings from the PM 2.5 sensors. As such, we developed a prototype enclosure from polyethylene terephthalate glycol (PETG), which is a polymer with high impact resistance and an adequate temperature range for our anticipated use [1].

To design the enclosure, we first developed a "naive" rectangular enclosure with a singular hole in the bottom for air flow. We then conducted a controlled experiment with one sensing node inside of the enclosure and one beside it, outside of the enclosure. We then observed the readings from each of the two sensors in the presence of a known source of particulate matter (an oil-based scent diffuser). In this experiment, we noted a significant difference in the two readings, which indicated that the first-generation enclosure was not allowing sufficient air flow to the PM 2.5 sensor.

To address the shortcomings of the initial sensor, we developed a computational fluid dynamics (CFD) model that allowed us to observe how air flow interacts with the enclosure. We then developed a second-generation enclosure with an external screen for blocking moisture, but offset from the main enclo-

[1] https://www.iemai3d.com/wp-content/uploads/2021/03/PETG_TDS_EN.pdf

sure by 1-2 cm to allow air flow. Through our CFD modeling, we observed that this screen allowed significantly more airflow to the sensor, and achieved near-parity with the external pressure (see Figure 5 for an example of the CFD modeling). Lastly, we conducted a second set of controlled experiments, and observed that the sensor in the second-generation enclosure recorded comparable PM 2.5 readings with the sensor located outside of the enclosure. An example of the experiment is shown in Figure 6. As a result, we concluded that this enclosure design should be adequate for deployments in locations where a *PCs for People* tower is not available.

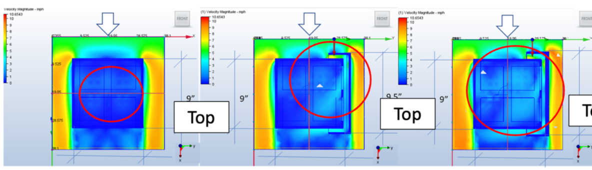


Figure 5: CFD Model of Airflow on Sensor Enclosure. Note that the newest-/final iteration shows higher airflow to the inside of the enclosure (where the sensor is located), and the airflow velocity is approximately equal to the applied velocity (5 mph).

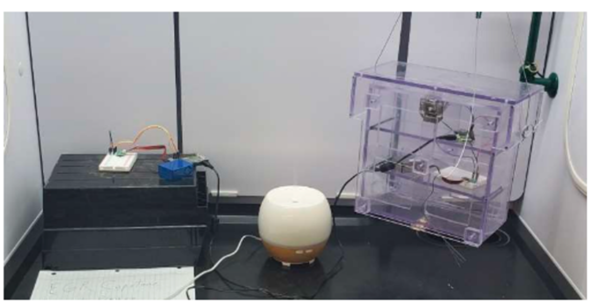


Figure 6: Controlled experiment with one sensing node located inside the prototype enclosure, and one located outside. Initial results show that the recorded PM2.5 levels were comparable between the two sensing nodes.

5 Conclusion and Future Work

This research has shown that by combining low-cost air quality sensors with an Internet of Things (IoT) platform, we can provide the community with fine-grained air quality monitoring that can provide valuable localized air quality data. From our pilot study, we found that in less than four miles there was a noticeable difference in sensed particulate matter where the average was more than double between the two locations. Additionally, in a longer-term deployment of over four months, there existed a higher variability in the sensed particulate matter than the regulatory sensor.

There are several areas of future work in this research. Limiting the deployment of sensors to cellular towers can restrict the ability to perform fine-grained sensing and detect differences among streets in close proximity to polluting sources. One approach to investigate is utilizing the modules deployed in the cellular towers as base stations that can communicate with edge sensor nodes by using LoRa. Another area is to perform more verbose error monitoring and reporting so that if a sensor is not reporting data that we can determine if it is an issue with the sensor module, the operating system itself, or a communication error. With the findings from testing the enclosure, more work is needed to design an enclosure that meets the goals of both allowing sufficient airflow while protecting the sensor module from the elements. Future work will include deploying additional sensors in the community as well as meeting with residents to better understand how air quality affects their daily lives.

Without a fine-grained deployment of air quality sensors, community members may never truly know an accurate measure of the quality of air they are breathing. Having only one PM 2.5 sensor for an area as large as Cuyahoga County is not feasible to provide this data, and it is not providing accurate or meaningful data to the community and its residents. We have found early success in our approach, and expanding the future work we have outlined will lead to more meaningful data for community members.

References

[1] Environmental Protection Agency. Health and environmental effects of particulate matter (pm). https://www.epa.gov/pm-pollution/health-and-environmental-effects-particulate-matter-pm, 2022.

[2] Environmental Protection Agency. Particulate matter (pm) basics. https://www.epa.gov/pm-pollution/particulate-matter-pm-basics, 2022.

[3] Joshua S. Apte, Kyle P. Messier, Shahzad Gani, Michael Brauer, Thomas W. Kirchstetter, Melissa M. Lunden, Julian D. Marshall, Christo-

pher J. Portier, Roel C.H. Vermeulen, and Steven P. Hamburg. High-resolution air pollution mapping with google street view cars: Exploiting big data. *Environmental Science & Technology*, 51(12):6999–7008, 2017.

[4] Jakub Dubec and Tibor Krajčovič. Building air quality monitoring iot platform. In *7th Conference on the Engineering of Computer Based Systems*, ECBS 2021, New York, NY, USA, 2021. Association for Computing Machinery.

[5] EPA. Airnow interactive map of air quality. https://gispub.epa.gov/airnow/.

[6] Xinwei Fang and Iain Bate. Issues of using wireless sensor network to monitor urban air quality. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*, FAILSAFE'17, pages 32–39, New York, NY, USA, 2017. Association for Computing Machinery.

[7] Adam Moore, Miguel Figliozzi, and Christopher M. Monsere. Air quality at bus stops: Empirical analysis of exposure to particulate matter at bus stop shelters. *Transportation Research Record*, 2270(1):76–86, 2012.

[8] Debanshee Saha, Manasi Shinde, and Shail Thadeshwar. Iot based air quality monitoring system using wireless sensors deployed in public bus services. In *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*, ICC '17, New York, NY, USA, 2017. Association for Computing Machinery.

[9] T. Sayahi, A. Butterfield, and K.E. Kelly. Long-term field evaluation of the plantower pms low-cost particulate matter sensors. *Environmental Pollution*, 245:932–940, 2019.

[10] World Health Organization (WHO). Air pollution data portal. https://www.who.int/data/gho/data/themes/air-pollution.

[11] Minqiu Zhou, Amir M. Abdulghani, Muhammad A. Imran, and Qammer H. Abbasi. Internet of things (iot) enabled smart indoor air quality monitoring system. In *Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things*, CNIOT2020, pages 89–93, New York, NY, USA, 2020. Association for Computing Machinery.

Engineering a Better Software Engineering Course*

Michael P. Rogers
Computer and Information Sciences
University of Wisconsin Oshkosh
Oshkosh, WI 54901
mprogers@mac.com

Abstract

One of the most difficult challenges in designing and teaching a software engineering course is to provide students with a realistic simulacrum of the workforce experience that awaits them after graduation. In industry, students will inevitably find themselves working on teams, interacting with clients who are much less technically adept than they are; and be expected to keep up with technologies that might not have even existed while they were studying computer science. In this paper, we describe an attempt to modify a traditional software engineering course to accomplish two goals: allow students to gain a deeper appreciation of the client perspective by serving as clients; and provide an opportunity for students to create a somewhat sophisticated software application using a newly emerging technology.

1 Course Design

Software Engineering, which we will refer to henceforth as CS 321, is a 3-credit course, offered once per year, open to students who have completed two introductory programming classes and a course in data structures. It is required for all computer science majors, and is an elective for those pursuing a minor. Students typically take it during their junior year.

CS 321 is a blend of theory and practice, based on and extending the ideas put forth in the ACM's Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [1]. While students are exposed to a substantial amount of theory in the course, the emphasis is on application: students work in teams of 3-4 for the entire semester, and iterate through the entire software development life cycle, using an agile framework. They spend a substantial amount of time becoming proficient with software development tools (IDEs, Git, Docker, etc.) and developing/honing their own strategies for managing a larger project. Teamwork and professionalism are emphasized, with students presenting their work at the end of each sprint.

Historically, students have been responsible for finding or devising suitable projects, with instructors using their veto power for those deemed too simplistic or ambitious. Instructors have aided somewhat in the search process, appealing to faculty at the university who might be in need of an app for either pedagogic or research purposes. The advantages of working for an external client have been well documented in the literature [10, 2, 5, 12, 11] and are well known to those who teach software engineering. Briefly these advantages distill down to a more realistic and personally satisfying experience. The apps fill a genuine need, the client having spent a considerable amount of time contemplating the shortcomings of their current, usually low-tech approach. Most clients are not technologically adroit, and so the act of eliciting user requirements, iterating through a series of prototypes, and gathering feedback, is of great benefit to the students.

Perhaps the biggest challenge in working with external clients is finding them. The ideal client a) has an appropriately scoped project, b) is willing to accept that success is by no means assured, c) is able to work around students' somewhat constrained schedules, and d) understands that long term support may be problematic. Beyond this, there are serious legal and reputational issues that need to be addressed before students and clients can start to collaborate.

To summarize, while the overhead involved in lining up a good external client is significant, there is a large payoff for students who are fortunate enough to find themselves working for such a client. Naturally, we would like all of our students to benefit from what is in some sense a quasi-internship, but it simply may be impractical to find an adequate number of clients. Thus, the instructor would appear to be confronted with two equally unpalatable choices. The first is to use no external clients, thus guaranteeing all students a comparable, but substandard, education. The second is to assign as many external clients as can be found, benefiting some of the teams, but violating the axiom that all students, in a single class, are entitled to the same education.

We decided instead to contemplate a third possibility, one that we were unable to find discussed in the literature (see section 4) and with which knowledgeable colleagues were also unfamiliar – modifying the project component of the course so that students would serve in both their traditional roles as developers and, novelly, as clients.

The new version of this course commenced in the same way as in prior years. At the beginning of the semester, students were placed in teams of 3-4 (chosen randomly by the instructor based on a first-week skills survey). After a discussion on the difficulties, and importance, of coming up with good app ideas, teams were required to complete the "Elevator Pitch" assignment (figure 1).

Topic Proposal: Propose **2** possible projects, substantial software products that you think would make somebody's life better. For each idea, describe:

1. The motivation (why it is a cool idea, why it is needed, who would use it, what makes it novel, etc.)
2. The high level functionality (what specific features it will provide, at least 4-5)

The project needs to be sufficiently complex to require a team of 4 people, working substantially each week, to develop it.

The end result will be a GUI front end developed in C# .NET using .NET MAUI technology, and persistent remote data storage (aka a database).

Each description should be around 100-150 words. Clearly indicate your **preferred** project (no ties allowed).

Include, in your proposal, a logistics section that will discuss

1. what technology you will use to communicate with (Zoom, Discord, FaceTime, Microsoft Teams), and
2. what time you plan to meet (either virtually or in person) at least 2x a week for a "daily" standup meeting.

Deliverables: Upload your project proposal (one submission per group) to Canvas and in the comments include a link to a GitHub repo containing your proposal (include your instructor on the team).

Figure 1: Elevator Pitch

Up until this point (week 3 of the semester), the course had been taught in exactly the same way as prior offerings. But after posting the elevator pitch assignment, the project component of the course began to deviate in three ways. First, a survey was administered (see section 3) to gauge student perceptions of the relationship between clients and developers, including the benefits of developers serving in the role of clients.

Second, once the elevator pitches were graded and posted anonymously, each team was asked to rank the projects that they would like to develop (excluding their own). This "Final Project Voting" assignment was a good team-building exercise in which all members, regardless of their technical abilities, could equally contribute (figure 2). Students were generally sanguine about the fact that the projects they had proposed were not the projects they would be building, in part due to several survey questions that encouraged them to reflect on the advantages of acting as clients.

Read over the final project proposals that have been submitted, and as a team discuss and then rank them in **decreasing** order of interest in developing (use the numbers). Do not include your own proposals in this ranking. Put your ranking in the submission comments - as a comma delimited list, please. For example:

26,42,21,25,33,30,31,35,23,35,28,41

means that you are most interested in developing app # 26; and least interested in app # 41. Your numbers will be between 1-14 (these were generated to avoid any appearance of favoritism).

Please do not confer with other teams about this, and do read over **every** single proposal -- all of them have something to commend them.

Your instructor will attempt to match your choices if possible, while balancing the imperative to provide a realistic software development experience.

Deliverables:

- The aforementioned ranking list, in the submission comments
- A photo or screenshot proving that all of you are at the meeting where this is discussed (to ensure that everybody is contributing).

Figure 2: Final Project Sorting

Finally, in a classroom activity dubbed the "big reveal", each team that had proposed a project, a "client team", was paired with the team that would develop it, a "developer team". It was possible to arrange matters so that multiple developer teams were able to develop their highest ranked proposal; no team was assigned a proposal that they had ranked lower than third; and each developer team was exclusively paired with a single client team, making scrum meeting scheduling much easier.

2 Emerging Technologies

In response to requests from area employers, for many years it has been a requirement that CS 321 projects be developed in C#, using Windows Presentation Foundation (WPF), a UI framework from Microsoft that is part of its .NET developer platform. The most significant drawback to WPF is hinted at in its name: in spite of the fact that .NET is cross platform, WPF only runs on Windows.

As we were preparing to introduce client-developer ideation into the course, however, Microsoft released a new UI framework, .NET MAUI (Multi-platform App UI), that allows developers to run apps on iOS, Android, macOS and Windows from a single codebase [9]. Since another implicit requirement of our area employers is that our students be proficient at learning new technology, we decided to take advantage of this fortuitous timing and change the platform to .NET MAUI.

In theory, the transition should not have been particularly difficult, as WPF and .NET MAUI are based on the same 3-tier Model-View-ViewModel architecture [8], and .NET MAUI is a direct descendent of Microsoft's well-established cross-platform product, Xamarin.Foms.

Some "teaching moments" did arise, however, as we began to work with .NET MAUI. Although it was a released product, and therefore ostensibly fully supported by Microsoft, it was apparent that it was a work in progress: new versions were being released weekly, and the documentation was not keeping pace. From a pedagogic perspective, however, these were invaluable opportunities to talk about real-life, practical issues that software engineers and architects face daily: how to evaluate upcoming technologies, when to switch from one technology to another, and how to develop strategies for getting assistance. One problem that we did not anticipate, and in hindsight should have dealt with more firmly when it first arose, was students' understandable desire to be able to develop on their own systems. The computer lab we reserved for CS 321 was well-equipped with modern, fast Windows machines: but perhaps 5-10% of our students had underpowered laptops, thoroughly unable to handle the combination of Visual Studio and the Android emulator, that made for a miserable development environment. In future offerings of CS 321, we will include and enforce a "minimum supported hardware" statement in the syllabus. We have also purchased Android hardware, to address this same issue.

As .NET MAUI matures and as the documentation stabilizes, it will be, overall, less useful as a vehicle for learning a new technology. Still, for the foreseeable future new packages and new features will be continually added, and those can approximately fulfill the same role.

3 A Survey of Client and Developer Roles

We conducted pre-and post-project surveys to assess student perceptions of the challenges facing client and developers, and whether there were steps that students could take to better understand these challenges. The pre-project survey focused on the hypothetical, whether students believed that they would benefit from serving as both clients and developers; the follow-up asked whether they had indeed benefited from so doing. We had a response rate of 88% (n=26) for the pre-project survey, administered during class at the beginning of the semester; the post-project survey, which due to time constraints was given during finals week, outside of class, had an unsurprisingly lower response rate of 27% (n=7): students were more concerned with completing their project presentations and preparing for exams.

Students clearly did have an appreciation for the difficulty, faced by clients, in expressing their needs to developers (figures 3 and 4).

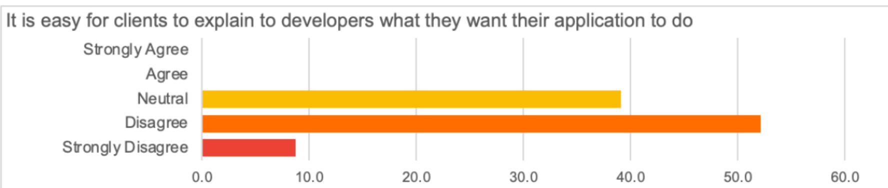


Figure 3: Client expository capabilities

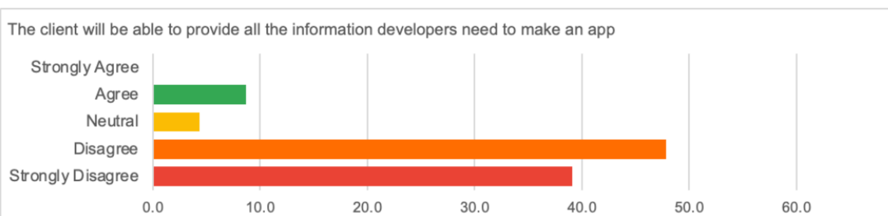


Figure 4: Client knowledge

However, while they expressed sympathy for clients, they were not as likely to believe that other developers understood the issues (figure 5).

Figure 5: Developer empathy

They did recognize that they had a moral imperative to protect their clients interests (figure 6):

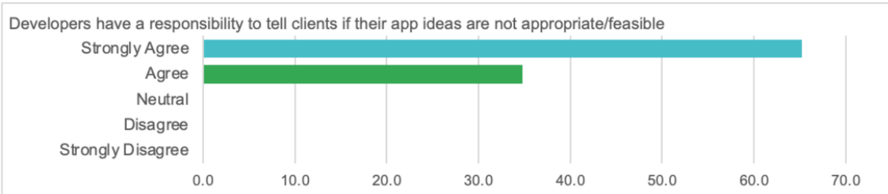


Figure 6: Developer responsibility to protect client interests

although when phrased in terms of meeting a client's expectations, most were willing to do what was required to meet those expectations (figures 7, 8):

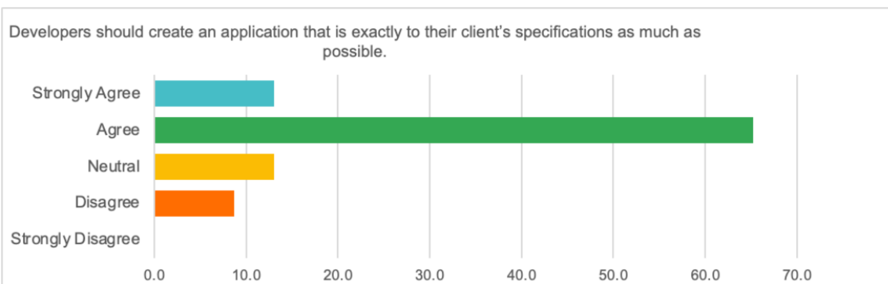


Figure 7: Developer responsibility to meet client expectations

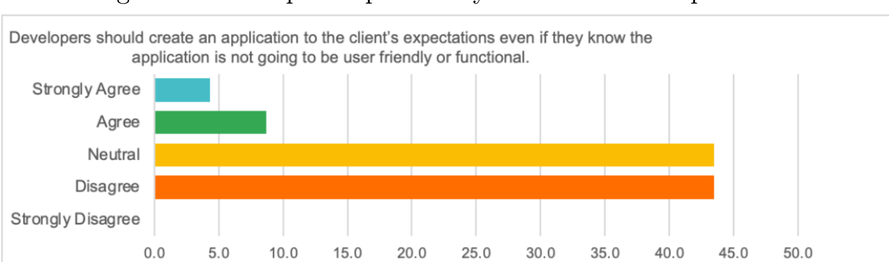


Figure 8: Developer willingness to create substandard apps

Importantly, students clearly and strongly valued working with clients. We asked the same 4 questions on both the pre-and post-surveys, and while there were some fluctuations due to response rates, they were not significant (figures 9-12):

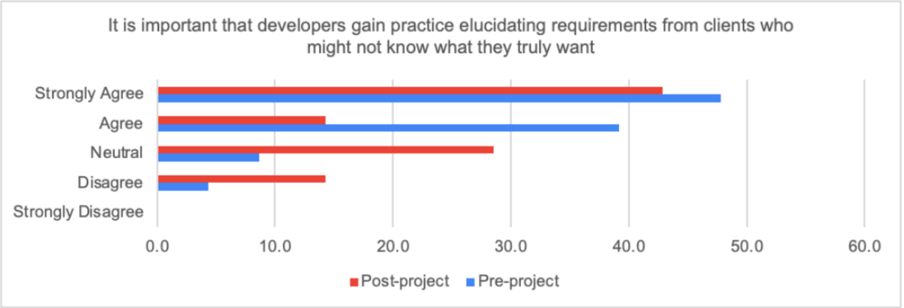


Figure 9: Need for practice gathering requirements

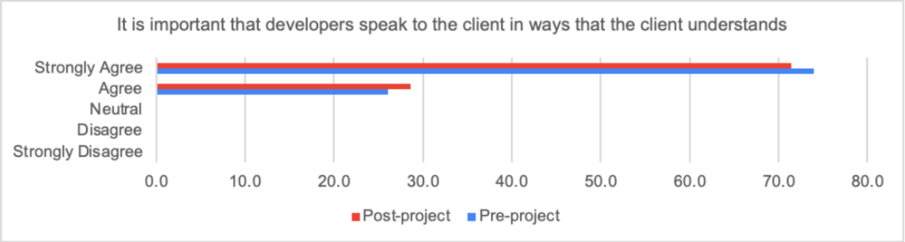


Figure 10: Communication skills

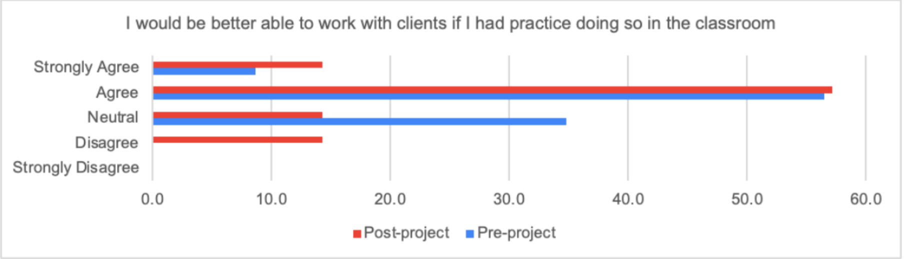


Figure 11: Need for students to have practice working with clients

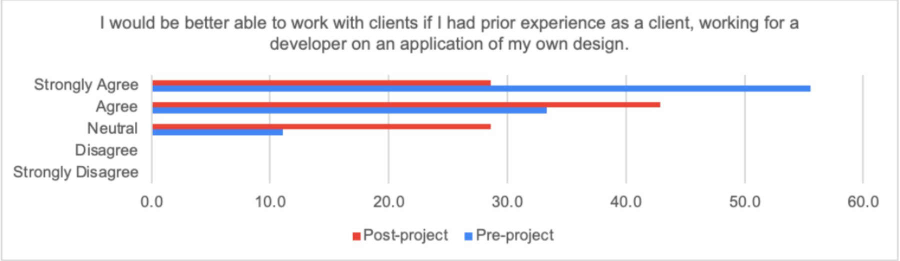


Figure 12: Need for students to have practice serving as clients

4 Prior Work

A review of the literature, and numerous conversations with colleagues, turned up no examples where students served as both clients and developers in the manner described in this paper. We did discover several references to the related concept of role playing. Costain and McKenna described using role-playing exercises to gather user requirements, and then record them in use case diagrams [3]. Decker and Simkins described a semester-long effort in which students acted as both owners and employees of an independent game studio, although there was no real separation between clients and developers [4].

Jeevamol and Renumol had an interesting exercise in which a graduate student role played being a client, using software projects they had created as an undergraduate: other students in the same team then generated various project artifacts (software requirement specification, design, test plan documents) [7]. Snidre had students role-play being both analysts and customers, to hone their oral communication skills [13]. Finally, Henry and LaFrance created some small role-play exercises that could be integrated into an existing course [6].

These works were generally smaller, shorter, focused exercises, not semester-long activities as described here, in which students genuinely developed a vested interest in seeing their visions realized. The only exception was [13], except in that case, students were owners/employees, there was no real distinction between clients and developers.

4.1 Conclusions and Future Work

By and large we were pleased with how this course went: the pairing of client and developer teams was successful, and student comments at the end of the semester were favorable. In their role as clients, students had to articulate their needs to their developers clearly, comprehensively, and early on, so it helped them focus on design. Bringing two teams together doubled the number of students who interacted on a regular basis, which had positive social benefits. Being able, during a presentation, to turn to the client team and get immediate feedback was highly valuable.

There were some lingering issues with the technology itself, and we have plans to address this – insisting that students use recent vintage computers, acquiring tablets so that students will not have to deal with balky emulators, and providing more comprehensive and accurate documentation, now that it is starting to appear in earnest.

The next time we teach this class, we will be more deliberate when it comes to conducting the surveys, administering the second survey slightly earlier (and, IRB-willing, use bonus point incentives to ensure higher post-project response rates).

There remains an open question as to whether or not this truly was as beneficial as if the students had worked with an external client, and one area of future research is to have half the class use internal clients, and pair half the class with external clients, and compare the results.

References

[1] ACM. "Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering". In: *ACM Computing Education* 15.4 (2014), pp. 1–31. DOI: 10.1145/2664239.

[2] Bernd Bruegge, Stephan Krusche, and Lukas Alperowitz. "Software engineering project courses with industrial clients". In: *ACM Transactions on Computing Education (TOCE)* 15.4 (2015), pp. 1–31. DOI: 10.1145/2739515.

[3] Gay Constain and Brad McKenna. "Experiencing the elicitation of user requirements and recording them in use case diagrams through role-play". In: *Journal of Information Systems Education* 22.4 (2011), pp. 367–380.

[4] Adrienne Decker and David Simkins. "Leveraging role play to explore software and game development process". In: *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2016, pp. 1–5.

[5] L. M. Grabowski, C. F. Reilly, and W. A. Lawrence-Fowler. "Emulating a corporate software development environment through collaboration between student projects in six courses". In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (2014), pp. 1–7. DOI: 10.1109/FIE.2014.7044137.

[6] Tyson R. Henry and Janine LaFrance. "Integrating role-play into software engineering courses". In: *Journal of Computing Sciences in Colleges* 22.2 (2006), pp. 32–38.

[7] J. Joy and R. V G. "Activity oriented teaching strategy for software engineering course: An experience report". In: *Journal of Information Technology Education: Innovations in Practice* 17 (2018), pp. 181–200. DOI: 10.28945/4116.

[8] Shaun Lawrence. *An Architecture to Suit You*. Berkeley, CA: Apress, 2023. DOI: 10.1007/978-1-4842-9234-1.

[9] Shaun Lawrence. *Introduction to .NET MAUI*. Berkeley, CA: Apress, 2023. DOI: 10.1007/978-1-4842-9234-1.

[10] Anne-Maarit Majanoja and Timo Vasankari. "Reflections on Teaching Software Engineering Capstone Course". In: *CSEDU* 2 (2018), pp. 68–77. DOI: 10.18287/2541-7817-2018-2-1.

[11] W. W. McMillan and S. Rajaprabhakaran. "What leading practitioners say should be emphasized in students' software engineering projects". In: *Proceedings 12th Conference on Software Engineering Education and Training (Cat. No.PR00131)* (1999), pp. 177–185. DOI: 10.1109/CSEE.1999.755199.

[12] Andres Neyem, Jose I. Benedetto, and Andres F. Chacon. "Improving software engineering education through an empirical approach: lessons learned from capstone teaching experiences". In: *Proceedings of the 45th ACM technical symposium on Computer science education* (2014), pp. 391–396. DOI: 10.1145/2534766.2534862.

[13] Guttorm Sindre. "Teaching oral communication techniques in RE by student-student role play: Initial experiences". In: *18th Conference on Software Engineering Education & Training (cseet'05)*. IEEE. 2005, pp. 85–92.

ChatGPT: The Good, The Bad, and The Ugly*

Conference Tutorial

James K. Huggins
Department of Computer Science
Kettering University
Flint, MI 48504
jhuggins@kettering.edu

ChatGPT [1] is a predictive AI chatbot that has recently gained significant attention for its ability to answer natural language questions on a wide range of topics with uncommon accuracy. It can even help users revise written texts, such as this abstract! However, teachers at all levels have expressed concerns about its impact on course assessment.

In this tutorial, we will present an informal and interactive introduction to ChatGPT. The tutorial will begin with a 45-minute presentation, showing ChatGPT's strengths and weaknesses (including a live demonstration), and presenting possible ways for academics to adapt their instructional practices in light of ChatGPT. Following a brief Q&A session, participants will be encouraged to spend the remaining time experimenting with ChatGPT, talking with other participants about their findings and developing plans for adapting their own teaching practices.

Biography

Dr. James K. Huggins is Associate Professor of Computer Science at Kettering University, where he has taught a variety of computer science courses over the last 26 years. His research interests include formal methods, computer science education, and computing ethics.

References

[1] OpenAI. ChatGPT. https://chat.openai.com/.

Bloom's for Computing: Crafting Learning Outcomes with Enhanced Verb Lists for Computing Competencies*

Conference Tutorial

Pam Schmelz
Ivy Tech Community College
Columbus, IN 47203
pschmelz@ivytech.edu

In this tutorial, participants will be introduced to Bloom's for Computing: Enhancing Bloom's Revised Taxonomy with Verbs for Computing Disciplines, a project of the ACM CCECC (Committee for Computing Education in Community Colleges). The Bloom's for Computing report offers a total 57 enhanced verbs across all six levels of Bloom's cognitive domain – Remembering, Understanding, Applying, Analyzing, Evaluating, Creating. The enhanced verb list is intended to support crafting more appropriate and less awkward learning outcomes and competencies that express the knowledge, skills, and dispositions required in computing disciplines. The Bloom's for Computing verb list and report is not just for use in future ACM curriculum guideline reports, but also for educators in computing disciplines who find themselves needing to craft learning outcomes or competencies – whether for programs, courses, or individual modules; whether two-year, four-year, graduate, or K-12 level; whether faculty, instructional designers, or program coordinators.

The presentation and activities in the proposed tutorial session are outlined below:

1. Introductions – tutorial facilitator and participants
2. Refresher on Bloom's Revised Taxonomy, its six cognitive levels, and common verbs lists
3. Interactive discussion on how faculty approach writing learning outcomes and some of the challenges encountered

4. Bloom's for Computing: Enhancing Bloom's Revised Taxonomy with Verbs for Computing Disciplines
5. Introduce the project and the verbs
6. Examples of learning outcomes using the Bloom's for Computing verbs Areas where the Bloom's for Computing verbs come in particularly handy
7. Activity where participants write or modify learning outcomes for courses they teach
8. Share out learning outcomes and thoughts on how the enhanced verbs might be used
9. Wrap up

Participants will be given a handout to take home with the complete list of verbs for each cognitive level.

This tutorial is relevant for anyone involved in writing, revising, or updating learning outcomes for programs, courses, or instructional units in computing disciplines such as Computer Science, Information Technology, and Cybersecurity.

Developing Identity-Focused Program-Level Learning Outcomes for Liberal Arts Computing Programs*

Conference Tutorial

Jakob Barnard[1], Grant Braught[2], Janet Davis[3], Amanda Holland-Minkley[4], David Reed[5], Karl Schmitt[6], Andrea Tartaro[7], James Teresco[8]

[1]University of Jamestown, Jamestown, ND 58405
Jakob.Barnard@uj.edu
[2]Dickinson College, Carlisle, PA 17013
braught@dickinson.edu
[3]Whitman College, Walla Walla, WA 99362
davisj@whitman.edu
[4]Washington & Jefferson College, Washington, PA 15317
ahollandminkley@washjeff.edu
[5]Creighton University, Omaha, NE 68178
DaveReed@creighton.edu
[6]Trinity Christian College, Palos Heights, IL 60463
Karl.Schmitt@trnty.edu
[7]Furman University, Greenville, SC 29690
andrea.tartaro@furman.edu
[8]Siena College, Loudonville, NY 12211
jteresco@siena.edu

The SIGCSE Committee on Computing Education in Liberal Arts Colleges (SIGCSE-LAC Committee) has found that liberal arts and small colleges approach design of their computing curricula in unique ways that are driven by institutional mission or departmental identity. This impacts how faculty at these colleges adopt curricular guidelines such as the current ACM/IEEE-CS CS2013[1]. The committee is developing guidance, informed by its sessions at recent CCSC and SIGCSE conferences, to help with the design and/or revision of CS curricula in liberal arts contexts [1]. This will ultimately be included in the committee's article in the Curricular Practices Volume that will be released as a companion to the new ACM/IEEE-CS/AAAI Computer Science Curricula guidelines (CS2023)[2]. Curricular guidelines like CS2013 or CS2023 inform curriculum design but are balanced with the vision for a program, departmental strengths, locale, student populations and unique academic experiences. The desire to craft distinctive curricula, combined with the size of prior curricular recommendations, requires an assessment of trade-offs between achieving full coverage of curricular recommendations and a school's other priorities. SIGCSE-LAC's guidance will encourage faculty to reflect on their programs and the role of CS2023, beginning with their institutional and departmental priorities, opportunities and constraints.

The specific goal of this session is to help participants develop program-level learning outcomes that align with the unique features of their programs. Following an overview and brief discussion of the newest CS2023 draft, participants will begin working through a preliminary version of the committee's reflective assessment process. This process is framed by a series of scaffolding questions that begin from institutional and departmental missions, identities, contexts, priorities, initiatives, opportunities, and constraints. From there, participants will be led to identify design principles for guiding their curricular choices including the CS2023 recommendations. Examples gathered from the committee's previous CCSC and SIGCSE sessions will be available to help to articulate identity and program design principles, which will then be used for the identification of identity-focused program-level learning outcomes. Participants will leave the session with a better understanding of how CS2023 can impact their programs and a jumpstart on the entire reflective assessment process. Feedback on the process and this session are welcome and will be used to refine the committee's guidance prior to its publication in the CS2023 Curricular Practices volume.

References

[1] Amanda Holland-Minkley et al. "Computer Science Curriculum Guidelines: A New Liberal Arts Perspective". In: *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2023. Toronto ON, Canada: ACM, 2023, pp. 617–623. ISBN: 9781450394314. DOI: 10.1145/3545945.3569793. URL: https://doi.org/10.1145/3545945.3569793.

[1] https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf
[2] https://csed.acm.org

The Influence of Generative AI on Pedagogy and Assessment in Computing Education*

Panel Discussion

Jean Mehta, Brett A. Becker, Wen-Jung Hsin
Joe Hummel, Bill Kerney, Brian Krupp

Student access to Generative AI tools stands to alter the way we teach as well as the way we assess our student's learning. ChatGPT has only been available for a few months, but already instructors are concerned about its wide use and implications. Love it? Hate it? Embed it in your course? Ban its use? Will this change not just how we teach but what we teach, when we teach it and even who we teach? Most of us have been wrestling with these questions, and more. Panelists will speak of how they altered their pedagogy, and the results, in both in-person and online courses.

Moderator:
Dr. Jean Mehta, Professor, Saint Xavier University, Chicago, IL.
`mehta@sxu.edu`
Participants (in alphabetical order):
Dr Brett A. Becker, School of Computer Science, University College, Dublin.
`brett.becker@ucd.ie`
Dr. Wen-Jung Hsin, Park University, Parkville, MO (Missouri)
`wen.hsin@park.edu`
Dr. Joe Hummel, Northwestern University, Evanston, IL.
`joe.hummel@northwestern.edu`
Bill Kerney, Clovis Community College, Fresno, CA.
`william.kerney@cloviscollege.edu`
Dr. Brian Krupp, Baldwin Wallace University, Berea, OH.
`bkrupp@bw.edu`
Dr. Jean Mehta, Professor, Saint Xavier University, Chicago, IL.
`mehta@sxu.edu`

Developing a Twitter Clone by using a No-Code Software Platform*

Conference Workshop

Huseyin Ergin
Computer Science Department
Ball State University
Muncie, IN 47306
hergin@bsu.edu

The rise of no-code and low-code tools has revolutionized software development, making it faster and accessible to all. In this workshop, we'll explore the history of these tools and dive into building a Twitter clone using the powerful no-code platform, Bubble (`https://www.bubble.io`). With Bubble, participants will learn to design software in a domain-specific language, eliminating the need for traditional coding. We'll start by creating a phone book application to grasp the basics of the platform. From there, we'll progress to a more sophisticated to-do list application, exploring advanced features along the way. Finally, we'll build a fully functional Twitter clone, demonstrating the immense possibilities of Bubble's code-free approach. Throughout the workshop, participants will experience an interactive and hands-on learning environment, where they'll develop applications simply by dragging, dropping, and connecting elements. No knowledge of a general-purpose programming language is required. In addition to the practical hands-on exercises, we'll discuss the future of software development and explore how no-code tools can be integrated into educational curricula. By introducing high-level software development concepts to beginner students before delving into complex coding, we can empower a new generation of software creators.

Schedule of Activities

1. Introduction: The history of no-code/low-code platforms (30 minutes)
2. Bubble introduction and account creation (15 minutes)
3. Building a phone book application (20 minutes)
4. Break, Q&A, and troubleshooting (15 minutes)
5. Building a to-do list application (20 minutes)
6. Break, Q&A, and troubleshooting (5 minutes)
7. Building a Twitter clone (35 minutes)
8. Break, Q&A, and troubleshooting (10 minutes)
9. Discussion: Future of no-code tools and next steps (30 minutes)

Notes

- Participants will need to bring their own laptops for this workshop.
- Bubble is free for personal use and a free plan will be enough for this workshop.

Interactive Finite Group Explorations Generated by Quantum Gate Matrices*

Nifty Assignment

John Pais
Mathematics and Computer Science
Ladue Horton Watkins High School pais.john@gmail.com

Recently, we developed an interactive introduction to quantum computing for high school students, which was implemented in several Maplesoft programmable texts. Basic quantum gates are unitary matrices in U(2, C). So, in the process of this development a natural question arose, namely, “what mathematical groups are generated by the standard quantum gate matrices and their combinations?" This was an engaging and interesting spin - off which could be addressed partially through the Maplesoft matrix object and its methods, and which motivated creating additional matrix group generation and identification code. This lead to the following computation related puzzle.

Group Generation Puzzle. Given qubit gate matrices determine what group they generate, using the given code enclosed in this Maplesoft text and/or developing additional code yourself. You may use ONLY the information you obtain through:

1. running this code, which will yield the order of the group, its order portrait and its currently known/unknown subgroups, and
2. comparison using the extensive database located here: https://people.maths.bris.ac.uk/~matyd/GroupNames/index120.html

Sometimes this approach will quickly be sufficient to identify the group. When you do so, you should also modify the code to include the identity of the newly found group. Often you will find too much ambiguity to make much progress initially. A good strategy is to focus on identifying the unknown subgroups of the unknown group in question, by hereditarily applying this same process to the unknown subgroups until enough are identified that lead to the identification of the unknown group. In this process you should, in addition, strive to identify all the unknown subgroups of the unknown group and also add their identities to the code.

In the Group Generation Puzzle, the group elements are sets of matrices under the operation of matrix multiplication. It turns out that this puzzle can be a very fertile ground for exploring a large variety of different interesting groups and understanding their basic properties and subgroup structures. This is an eclectic, computation-based approach to learning/exploring basic group theory, in contrast to the standard abstract approach given in most group theory textbooks. In this talk we will discuss several specific examples of groups discovered and explored as the result of this eclectic approach.

Configuring the Firewall on a Web Server Using Cisco Packet Tracer*

Nifty Assignment

Imad Al Saeed
Computer Science Department
Saint Xavier University
Chicago, IL 60655
alsaeed@sxu.edu

This assignment is for a course on Introduction to Cybersecurity for students with no prior knowledge with Cybersecurity, but they should have prior knowledge about computer networks. The course instructor explains how to build a LAN and configure the firewall on the web server as a solution for ensuring security when accessing to web server. These are the main requirements of configuring the firewall on a web server in computer networking and how to allow access to a certain site and deny the ping of that site. This assignment requires using special software called Cisco Packet Tracer where students can download for free from the Cisco Academy site (Netacad.com) and install it on their computers. Students should use the newest version of the software to do this assignment. This process should be done in the following steps:

1. Assign IP address to the Server.
2. Activate the DHCP service on the server.
3. Activate the HTTP services on the server.
4. Configure the firewall within the server by Denying the ICMP and Allow the IP.
5. Reconfigure the server by allowing the ICMP and IP on the server.
6. Ping the server and access the URL for each computer. Students should successfully access URL but should receive a “Request time out" when pinging that site.

CS-458 Software Engineering Practicum: Senior Project Pitch Showcase*

Nifty Assignment

Tyler Thomas
Math, Stat, Computer Science Department
University of Wisconsin-Stout
Menomonie, WI 54751
thomastyl@uwstout.edu

This work showcases an educational intervention aimed at promoting creativity, innovation, and sustained motivation in senior software engineering students. The project pitch assignment, a component of a semester-long senior capstone project, spurs students to break free from the norm and embark on projects that resonate with them personally. Students pitch ideas in one of five categories, each with varying degrees of practicality, realism, and humor. Projects may be either 1: A useful tool that solves a real problem in a realistic way, 2: A useful tool that solves a real problem in a simplified or fictional way, 3: A useful tool that solves a fictional problem in a realistic way, 4: A useful tool that solves a fictional problem in a simplified or fictional way, or 5: A humorous tool that solves a questionable problem. Students then cast private ballots for the ones they like the best and are grouped into teams by the instructor around those with the most votes. Students vote privately for the most appealing ideas, and the instructor groups students into teams based on the most popular ideas. This versatile project proposal format encourages the generation of innovative project ideas, with the voting process providing refinement. The increase in creativity among projects and engagement among students this semester indicates a positive response from students to this progressive approach. This approach could serve as a model for fostering creativity and engagement in software engineering education or other related fields.

Towards an AI-Aware Pedagogy in IT Education*

Work in Progress

Suguna Chundur and Kristi Hall
University of Cincinnati - Clermont College
Batavia, OH 45103
{chundusa,hall3kj}@ucmail.uc.edu

With the recent sudden explosion in the development of AI-based tools there is uncertainty among educators on its impact in teaching and learning as well as concerns about integrity of student work. Simultaneously, AI tools present a unique opportunity to personalize education and encourage self-driven learning. Hence, there is an urgent need to understand the impact of AI-based tools on teaching and learning and to find a way forward in building a framework that directly addresses the challenges raised by generative AI in three main areas: strategies for AI-aware teaching, reliability of AI-generated knowledge and design of assessments. With a view towards acquiring a clearer picture of the new AI-aware teaching landscape, we have attempted to incorporate AI-tools in the classroom. We have experimented with using various AI-tools in designing assignments that co-opt AI for teaching and learning, whether it is a written essay on an IT topic, a programming assignment in Java or trouble-shooting a systems administration issue, topics that cover the breath of our Information Technology program. The assessments provide clear instructions to students on acceptable and unacceptable ways of using specific AI-tools in completing the work. There is a feedback question in each assessment that seeks students' perception of AI-tools used, its strengths and weaknesses in learning. We plan to continue the experiment for an entire academic year using this methodology in various IT courses. In this work-in-progress presentation, we will discuss the results of the assessments administered so far and provide an overview of the resulting draft AI-aware framework of teaching and learning in the IT discipline.

A Teaching Strategy for Undecidable Problems*

Work in Progress

Pradip Peter Dey, Mohammad Amin
Bhaskar Raj Sinha, Hassan Badkoobehi
School of Technology and Engineering
National University
San Diego, CA 92123
{pdey, mamin, bsinha, hbadkoob}@nu.edu

Important computational studies of undecidable problems started with Allen Turing's 1936 proof that the halting problem is undecidable. This means that there is no algorithm that can correctly answer the halting problem for all possible computer programs. Subsequently many interesting problems are added to the list of undecidable problems. Understanding the concept of un-decidability is important for learning the scope and limitations of computing. One of the requirements for teaching undecidable computing problems is to go through the major steps of the proof in order to convince the learners that the proven theorem is true forever demonstrating that the problem is not decidable. The rigorous proofs of undecidable problems are composed of logical steps. We suggest that discussion of intuitive abstractions with explanations of some examples before presenting logical abstractions works better for teaching/learning purposes. An overview of contrastive differences between classes of languages or sets such as Turing recognizable, decidable and undecidable sets may promote learning in most environments. Since some of the steps of the proof are lengthy, an intuitive overview of the logical proof may help students. For the emerging online synchronous teaching-learning environments with standard Learning Management Systems (LMS) and video conference tools, establishing cross references between related parts of the proof may become problematic for some of the important computational problems. Intuitive abstractions about membership of some undecidable sets are explained with some interactive examples at the following site: `http://www.asethome.org/mathfoundations/asynchronous/`. Most learners initially depend on intuitive abstractions about examples because it is not evident that no algorithmic solution can exist for an undecidable problem. Our research critically examines consequences of some of the traditional teaching strategies for undecidable problems in various environments. An effective strategy is to explain with examples that a Turing Machine (TM) as a decider cannot be built for an undecidable problem. When a TM is started on an input, three alternative results are possible. The machine may halt and accept the input, or it may halt and reject the input, or it may loop forever by failing to halt [1, 3, 4].

If learners are familiar with integral polynomials then decidability problems can be explained with the help of sets of integral polynomials. One-variable integral polynomials are decidable whereas multivariable integral polynomials are undecidable [2, 4]. For one-variable integral polynomials, a decider-program (or Turing machine) can be constructed, in addition to a recognizer-program. For multivariable integral polynomials, only a recognizer program can be constructed, that is, a decider-program can never be constructed. Using example sets of polynomials an instructor can effectively explain intuitive ideas behind the contrastive differences between decidability and un-decidability issues. We hope that most learners will be able to exploit the knowledge gained from examples of one environment to improve generalization about decidability in other environments. In the next phase of our research, we will collect scientific data in order to examine evidential results for or against our hypothesis.

References

[1] Daniel IA Cohen. *Introduction to computer theory.* John Wiley & Sons, 1996.

[2] P. Dey. *Multivariable Integral Polynomials are Undecidable.* Retrieved July 15, 2023. URL: `http://www.asethome.org/mathfoundations/undecidable/`.

[3] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. *Introduction to automata theory, languages, and computation.* Pearson, 2007.

[4] Michael Sipser. *Introduction to the Theory of Computation.* Cengage Learning, 2012.

Group Review Sessions: Online Tutoring for Success in Programming Courses*

Work in Progress

Kristi Hall, Suguna Chundur, Valentine Johns
University of Cincinnati - Clermont College
Batavia, OH 45103
{kristi.hall,chundusa}@uc.edu
chundusa@ucmail.uc.edu

At the University of Cincinnati Clermont College, the online Information Technology degree gives students an opportunity to delve into the areas of networking, web development, programming, and technical support. Two courses that are key to student success when they enter the program are IT1090 Computer Programming I and IT2040 Fundamentals of Web Development. Student success in both courses is indicative of their success in the program. Historically, both courses have had a high drop, fail, withdraw rate. The University of Cincinnati has established a Learning Commons which offers free academic support to students through academic coaching, math and writing support, and peer tutoring, among other services. Starting in the spring of 2023, the Information Technology program began collaborating with the UC Clermont Learning Commons in offering Group Review Sessions for both IT1090 and IT2040. Group Review Sessions are led by peer tutors. These peer tutors are second year students who previously took these courses and were outstanding. Group Review Sessions are scheduled several times a week, and the peer tutors work with students synchronously through Microsoft Teams. The project started out with Group Review Sessions being voluntary and students receiving extra credit points for attending. We have now made Group Review Sessions mandatory and part of the students' grades. The goal of this project is to provide students with more academic support to ensure not only their success in these courses, but also a better understanding of these concepts as they continue to move through the program.

Strategies for Introducing Mathematical Concepts to Computer Science Students*

Work in Progress

Mohammad Amin, Bhaskar Sinha
Pradip Peter Dey, Laith Al Any
National University, San Diego, CA 92123
{MAmin, BSinha, PDey, LAlAny}@nu.edu

In Computer Science, mathematics plays a fundamental role in developing analytical thinking and algorithmic reasoning skills. However, students often struggle to grasp the significance and applicability of mathematics in this field. They need a clear understanding of math concepts and skills for the application of mathematical constructs in computational problem-solving. One of the challenges for math teachers is to help poorly prepared students at introductory levels. This requires an innovative approach that bridges the gap between theory and application. Computational problem-solving requires more than just memorizing formulas. Examples of real- world problem-solving applications of formulas can be integrated into computational solution techniques to develop students' intuitive concepts of math at introductory levels. Proper integration of motivational activities with practice-based math instructions in multiple classes requires careful planning and execution.

Instructors strive to teach math effectively and enhance students' engagement in math learning. We are examining and evaluating different learning tools and alternative methods for improving the learning experience. Available AI tools (ChatGPT, Google Bard, YouChat, etc.), and YouTube Channels for Learning (Khan Academy, Crash Course, TED-Ed, ASAP Science, Wolfram Alpha, etc.) are being examined for augmented math learning. We try to find a reasonable balance in our teaching approaches about concepts, theories, and practice in application areas. We are experimenting with the following strategies for math teaching: explicit instructions, use of visual tools, learning by doing, conceptual understanding, reflection on learning, deep collaborative engagement, and math-language skill development. We have set the stage for collecting scientific data from our experiments on teaching math at introductory levels. Our initial observations suggest that students can reason abstractly and quantitatively, ask questions and define problems, plan and carry out investigations, apply mathematical models and tools, construct viable solutions with justifications, and code solutions in a programming language.

Undoubtedly, computer science students need a clear understanding of mathematical concepts, especially the true essence of algebra and its applications. Most computer programs are written using algebra for repeating the computation of similar items. In this case, students must know how to apply math in various areas of real-world business, scientific, and engineering applications. In our observations from our long teaching experiences, most students memorize some algebraic theories and try to solve the assigned homework and exercises, and attempt to pass the exams to get good grades. A few students pay serious attention to the overview of the ultimate goals of math concepts and their application techniques in computational science. Indeed, every student learns differently. One teaching method does not work for all. Instructors must use different teaching methods to reach all learners. This might include using formal lectures, interactive discussions,teamwork, problem-solving tutorial sessions, hands-on activities, short videos, animations, graphical explanations, and audio-visual descriptions.

We are evaluating several teaching strategies for math topics related to computer science programs, including but not limited to the following:

1. Start with the basics of math to make sure students have adequate preparation before introducing intermediate and advanced topics.
2. Apply math to solve some common and real-world problems and emphasize its relevance in developing algorithms and writing codes.
3. Challenging students with more advanced math and creative applications is helpful for a deeper understanding of the concepts.
4. Provide motivational examples related to math applications that spark students' curiosity and interest in deeper learning that helps them develop logical constructs for computer science.
5. Break down a big and complex math topic into smaller and simple pieces which are easy to manage and comprehend. Always remember your past, and how you learned, put yourself into students' shoes, and explain from their platform. It is not important to show how much you know, but important how simple ways you can explain a complex topic that your students can digest and apply.
6. Encourage students to know what the reasons are behind some specific learning of math topics and how they will get benefits from these after learning.
7. Learning has no end; motivating students to continue learning is important. Computer Science is an ever-growing discipline, it is always addressing new challenges and developing new fields.

Exploring the Quality and Vulnerabilities of Open-Source Systems: An Empirical Investigation*

Work in Progress

Ahmed A. Elmagrous and S. M. Alnaeli
Department of Computer Science
University of Wisconsin-Stout
Menomonie, WI 54751
elmagrousa@uwstout.edu salehalgadry@gmail.edu

An empirical study is presented that examines the source code quality in general-purpose open-source systems. The study is conducted on nine open-source systems comprising over 4.9 million lines of code (MLOC). The main focus was on some good programming practices affecting practical coding aspects such as parallelizability, security, and analyzability. We assess prevalent coding elements: function side effects, jump statements, vulnerable and recursive functions. These elements gauge adherence to best practices by developers in most system domains, in academia and industry. The investigation is exposing patterns in side effects, recursive functions, jump statements, and unsafe functions, echoing findings in systems from different business domains. Such coding elements increase software complexity, impacting some of the important software engineering processes e.g., static analysis, multithreading, and security. Historical analysis over five years reveals an enduring presence of these elements, compounding challenges in software analysis for the studied systems. Additionally, this research highlights coding practices' impact on software quality, underlining the urgency of addressing these issues within the software development community in general and in computer science and software engineering academic programs in particular.

Software Design Comprehension Using Function Point and Artificial Intelligence in Computer Science Programs*

Work in Progress

Bhaskar Sinha, Pradip Peter Dey, Mohammad Amin
School of Technology and Engineering
National University, San Diego, CA 92123
{BSinh, PDey, MAmin}@nu.edu

Software design complexity estimation is a critical aspect of the software development process, determining project timelines, budgets, and resource allocation. Accurate estimations are essential for successful project planning and management. Traditional estimation methods often face challenges, resulting in inaccuracies and project delays. Some of the limitations of traditional estimation approaches include 1) Design complexity, where the numerous variables make the manual calculation and comprehension very cumbersome and prone to errors. 2) Project requirements, customer design, or technology requirements can change during the development process, rendering initial estimates outdated and unreliable. 3) Available data limitations, where the accessible historical data might not be representative of the current project, making it challenging to extrapolate accurate estimates. 4) Subjective decisions, where traditional estimation methods often rely on the expertise and intuition of human estimators, leading to biased and inconsistent results. 5) Finally, the customary manual estimation is time-consuming, impacting the overall project timeline.

This first part of this research is an extension of the work by McCabe, Ousterhout, and Albrecht, and investigates the prospect of using a combination of Function Points and Code Complexity to assess the design in a logical manner. Use of these practices will enhance the traditional teaching approaches and enhance student learning. This analysis starts with the conventional Unified Modeling Language (UML) diagrams and merges functional dependency, system security, database structures, and computational complexity. The notions of Unadjusted and Adjusted Complexity Indexes are used and related to the existing concepts of Influence Factors and Complexity Multiplier. The second part of this research involves the use of Artificial Intelligence (AI) to augment the comprehension of the design, leading to enhanced accuracy and efficiency. This explores the application of AI in software design complexity estimation, its benefits, challenges, and future prospects. AI has the potential to transform software complexity estimation, revolutionizing the way projects are planned, executed, and managed. While challenges exist, ongoing research and developments in this field promises a bright future for AI- driven software estimation. Utilizing the power of machine learning and data analysis, AI can significantly improve estimation accuracy and efficiency with respect to customary approaches.

The integration of the above two elements will offer numerous advantages, and this research has the potential to transform software design complexity estimation, revolutionizing the way projects are planned, executed, and managed. Ongoing research and development in this field promises a bright future for software design estimation and comprehension.

Old MacDonald Had a Farm*

Nifty Assignment

Stephen Larson
Information Systems and Technology Management
Slippery Rock University of PA
Slippery Rock, PA
stephen.larson@sru.edu

This Nifty Assignment is an optimization problem. Optimization problems can be solved in a variety of different ways. High school students will most likely use Algebra to solve them. College students may use higher math. As shown by the graph in the Introduction worksheet of the workbook, once a person enters the working world, they begin to use spreadsheets to solve problems.

Old MacDonald has a farm and many animals to feed. He is trying to find the best combination of types of feed to give his animals to maximize nutrients and minimize cost. Old MacDonald plans to mix two types of food to make a mix of low cost feed. A bag of food A costs \$10 and contains 40 units of proteins, 20 units of minerals and 10 units of vitamins. A bag of food B costs \$12 and contains 30 units of proteins, 20 units of minerals and 30 units of vitamins. He must figure out how many bags of food A and B should be consumed in order to meet the minimum requirements of 160 units of proteins, 100 units of minerals and 60 units of vitamins at a minimum cost? This problem is easily solved using Algebra. In your solution, include the graph of your feasible region with all vertices labeled.

Now that we have solved this using Algebra, we will now solve it using Excel's optimization tool called Solver. Solver lets you specify an objective (in this case, minimize the cost), while meeting the business constraints (in this case, nutrient requirements).

The feed problem is explained in the Feed Problem worksheet in the Excel workbook. The assignment assumes the students know Algebra, and are familiar with MS Excel's solver tool to solve the optimization problem. Both the Algebraic and Solver solutions are given.

Reviewers — 2023 CCSC Midwestern Conference

Dave Surma .... Indiana University South Bend, South Bend, IN
David Bunde .... Knox College, Galesburg, IL
David Largent .... Ball State University, Muncie, IN
Deborah Hwang .... University of Evansville, Evansville, IN
Edward Lindoo .... Regis University, Denver, CO
Imad Al Saeed .... Saint Xavier University, Orland Park, IL
Jean Mehta .... Saint Xavier University, Orland Park, IL
James Vanderhyde .... Saint Xavier University, Chicago, IL
Khaled Elzayyat .... Salt Lake Community College, Salt Lake, UT
Michael Rogers .... University of Wisconsin Oshkosh, Oshkosh, WI
Paul Gestwicki .... Ball State University, Muncie, IN
Paul Talaga .... University of Indianapolis, Indianapolis, IN
Fadi Wedyan .... Lewis University, Romeoville, IL
Hector Hernandez .... Benedictine University, Waukegan, IL