ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #40889

Instructor-Facing Graphical User Interface for Micro-Credential Designation and Refinement in STEM Curricula

Mr. Paul Amoruso, University of Central Florida

Paul Amoruso earned his B.S. in 2021 and his Master's degree in 2023. Currently, he is continuing his education in the same field and pursuing a Ph.D. in Computer Engineering at the University of Central Florida, Orlando, FL. Since 2022, he has served as a Graduate Research and Teaching Assistant for the Department of Electrical Engineering and Computer Science. His educational interests are innovations in machine learning implementations in everyday aspects particularly in education, technology-enabled learning, and feedback driven grading approaches. His thesis was titled "Micro-credentialing with Fuzzy Content Matching: An Educational Data-Mining Approach." ECE Best TA Award, Three Minute Thesis finalist, and Senior Design Day Showcase winner of Fall 2021, are a handful of many awards obtained within this pursuit. In addition, the PhD research led by Mr. Amoruso is aiming to integrate machine learning in the collected educational data gathered in his Master's research.

Dr. Ozlem O Garibay, University of Central Florida

Dr. Garibay has over 20 years of combined experience in creating and developing multimillion-dollar complex software systems and research in artificial intelligence. She is an assistant professor in industrial engineering and director of the UCF Human-Centered AI Research Laboratory, a cross disciplinary team dedicated to investigating artificial intelligence from a human-centered context. Prior to that, she served as the director of the UCF's Research Technology Office. Her areas of research are applied machine learning in drug discovery and molecule design, social media analysis, social cybersecurity, social and economic networks, network science, and evolutionary computation.

Dr. Laurie O Campbell, University of Central Florida

Laurie O. Campbell, Ed.D., is an Assoc. Professor in Learning Sciences and Educational Research, at the University of Central Florida. She pursues research to better serve underrepresented populations in STEM.

Dr. Ronald F. DeMara P.E., University of Central Florida

Ronald F. DeMara is Pegasus Professor in the Department of Electrical and Computer Engineering, and joint faculty member of Computer Science, at the University of Central Florida, where he has been a full-time faculty member since 1993. He has completed over 325 articles, 50 funded projects as PI or Co-PI, and 56 graduates as Ph.D. dissertation and/or M.S. thesis advisor. He was previously an Associate Engineer at IBM and a Visiting Research Scientist at NASA Ames, in total for four years, and has been a registered Professional Engineer since 1992. He has served ten terms as a Topical Editor or Associate Editor of various IEEE Transactions and in many IEEE/ACM/ASEE conferences including General Co-Chair of GLSVLSI-2023. He has received the Joseph M. Biedenbach Outstanding Engineering Educator Award from IEEE and is a Fellow of AAAS.

Instructor-Facing Graphical User Interface for Micro-Credential Designation and Refinement in STEM Curricula

Abstract

An instructor-facing Graphical User Interface (GUI) for micro-credential formation and tracking within STEM curricula is designed, prototyped, and evaluated. It enables instructors to readily associate skills with response-level elements within digitized assessments of Canvas LMS for issuance of micro-credential badges, without the need for computer programming expertise. An instructor survey of preference and acceptance levels of skill tracking was conducted. Survey results were used to design an input flow and interface to tag skills. The Tkinter Python-based computer coding library was utilized to create rapid check buttons and the Matplotlib backend was deployed to generate pie chart views of instantaneous course performance against tagged skills. A case study demonstrates the process for highest and lowest achieved skills via an auto-appending CSV file, allowing the instructor to identify and react to semester-long performance of learners by tracking badge issuance rates up to 17.3% within a required Computer Engineering undergraduate course.

Keywords

Micro-credentialing, Learner Outcome Tracking, Digitized Assessment, User Interfaces for Information, Communication Technology in STEM Education

Related Work

Awarding badges or the process of micro-credentialing students continues to evolve, especially in higher education where course letter grading has had a predominant role for centuries. More importantly various employers who seek to hire graduates have been seeking employer-facing solutions that can provide more confidence in selecting future employees that match the desired skillsets. Therefore, some educational corporations have aimed to provide solutions to fill the niche to provide badges to “validate competencies, drive engagement, improve completion, and increase enrollment”, as stated in the Canvas credentials digital badges for higher education site [1]. The current market is primarily led by companies like Badgr and Accredible, prioritizing activity-centered credentialing methods rather than the automation of assessment-based skill evaluation through an interface directed towards instructors, an approach explored and improved upon in this paper.

The Badgr framework grants instructors and course designers the ability to manage badge requirements in the courses and review privacy protected leader boards and other features such as learning pathways to motivate students through the course. Badgr provides a leaderboard that helps to track individual student performance and provides a gamified view of acquiring badges. Considering the diverse range of features supporting badge creation, distribution, and assessment provided by the company, the system can restrict instructors to award badges solely based on criteria such as overall course grade, completion, module fulfillment, and assignment performance or completeness [2].

The Accredible framework highlights its product through three primary features: automating badges and certificates, enabling students to view badges within the Canvas LMS, and employing single sign-on (SSO) for streamlined authentication [3]. Similar to other products, Accredible relies on the instructors or course designers to employ topic-specific modules to assess a student's eligibility for credentials. The company states that they have worked closely with customers for easy integration with the LMS.

Most products and services within the industry that are in the same or similar realm often necessitate instructors to utilize multiple websites/platforms and depend on using modules to track student performance on a skill or topic that requires a badge. There are three key features that sets the AchieveUp user interface and experience presented in this paper apart from both Badgr and Accredible, and that is minimal extension to implement in a preexisting course, the ability to tag multiple skills throughout the semester standard digital quiz-based assessments, and a simple interface that needs very little learning to operate [4]-[5]. Future work on AchieveUp will focus on extending the user domain to more curricula and integration with machine learning techniques to help instructors identify and track skills with minimal expenditure of time and effort.

Introduction to Automating Micro-Credentialing Framework for STEM Curricula

Micro-credentialing has been an active and fruitful area of educational research within recent years. The drive to utilize micro-credentialing is due to its ability to innovate the field of education by assessing students' acquired skills beyond the scope of conventional grades or assessment scores available to instructors. The user-centric interface implementation builds upon an open-source NSF project aimed to measure student performance and disperse badges in a manner that has not been achieved prior. The context in this framework is to data mine various key information from the assessments in the learning management system (LMS) to award digital badges to students that show comprehension of topics seen in a course [6]-[9]. There are opportunities to understand and broaden participation in undergraduate curricula, especially within courses which are required to earn a degree within STEM fields [10].

In response to remediating at-risk students, various methods have been explored, such as C. L. Bruder's approach to develop a gamified curricula for engaging student participation and garnering digital badges [11]. Additionally, other forms of improving modern education in courses can be demonstrated by implementations of an evaluation and proficiency center (EPC) or Electronic Laboratory Assessments (ELAs), where students take proctored digitized quizzes that facilitate in remediation via personalized tutoring with graduate teaching assistants [12] – [13]. In building upon various implementations, AchieveUp was coined as the overarching name for the framework upon which the instructor-facing graphical user interface for micro-credential was applied to. AchieveUp consists of various phases, consisting of tagging skills to questions, acquiring student performance, and dispersions of digital badges to represent skill competencies in courses.

This paper focuses on the feasibility, design requirements, prototype development, and refinement of user interface to support instructor-facing micro-credentialing flow. The employed approach expands upon foundational research that leverages student submissions within Canvas to evaluate and award badges corresponding to skills attainable within the course. Instructors seeking to incorporate micro-credentialing into their courses should be able to assign skills to individual questions as well as receive useful data in an easy and useful manner. Thus, making it integral that this project can allow the foundational research of micro-credentialing port into the real-world undergraduate curriculum. The work herein extends the authors' Python-based analysis toolset, which was developed to identify achievable skillsets of students, to extend it into a fully user-accepted micro-credentialing tool. In the following sections the details in regard to steps taken for the extension is presented. The outcome of the project will allow instructors to not only better manage how or what they focus on when teaching a course, but also allow for more targeted tutoring by being able to identify and fill knowledge gaps of at-risk learners and disperse badges to reward high achieving students to display the skills received in the course. The development of said project is dependent on taking the already developed work on micro-credentialing and tailoring it for utilization among various STEM instructors. As described by G. Ghaith and H. Yaghi, instructors often are more willing to implement new implantations to their classroom when it aligns with their current workflow, signifying the importance that an implemented framework should not add many steps to the already busy workflow of an instructor [14].

At the commencement of the project, the primary step as with any user-interface design endeavor, entails an examination of the requirements and desires of the idealized user. In design brainstorming sessions with prospective instructors, a prototype user interface was formulated based on their user needs and preferences. In the preliminary informal survey aimed at enhancing the micro-credentialing process, the authors gathered feedback from prospective instructors regarding the feasibility of a graphical user interface to tag skills with questions. Ideally, this would display a visualization of the overall performance per skill after students have completed the quiz-based assessment. The application that tags skills should have a very simple interface which consists of traversing all the questions in each quiz and having the functionality to select multiple skills per question. The project was further refined for master's thesis work and unpublished as of yet [5].

Next, the user interface comprising the feedback portion, would rely on the development of an application that that visually illustrates areas where the class excels or needs improvement in attainable skills. The initial version of the applications became refined during a student-led user interface design project. The recommendations obtained centered on providing a more user-centric design, as well as pushing the feedback of skills in a more concise manner, rather than simply a pie chart. The primary challenge that the project aims to resolve is the fact that many instructors are not inclined to interact with Python scripts, and or mundanely go through a matrix of questions to tag skills to each question. Thus, the primary objective was to develop a user-friendly application that demands minimal learning curve for successful operation.

During the phase of application development, the foundation was laid by utilizing the pre-existing coded algorithm from previous iterations of the research. The inclusion of the Tkinter package, chosen for its status as a standard Python interface to the GUI toolkit, extended the functionality [15]. A segment of the project's development phase involved acquiring the proficiency to seamlessly incorporate the graphical user interface (GUI) framework into the existing Python scripts, which contain the logic for generating a matrix and assigning badges to students. Under the constraints leading the knowledge needed for crafting a user interface in the specified programming language, the design incorporates a JSON file for instructors to input an access token and skill names, obviating the necessity for the application to request basic information upon each instructor initiation. The system presented embodies the definitive structure for the foundational version of the user interface, positioning it for beta testing in advance of full-scale deployment.

The applications that structure the AchieveUp framework consist of two main programs utilized throughout the semester, which are to extract questions to tag, and calculate student performance in skills. The overall process can be streamlined through three primary phases. The first phase is the extraction phase, which is required to extract the questions from Canvas LMS, to develop a data matrix that is further used to tag with skills. The second phase is the credentialing phase, this is the backbone of the project. The credentialing encompasses the traversal over the matrix of questions and student performance to assign the skills for each assessment to the students. Then the most recognizable phase is the reporting phase; this encompasses the badge dispersals with overall student-by-student report of each skill. Given the AchieveUp's ability to tabulate, calculate, and present student performance is highly dependent on matrixes that the user interface is hiding, the title "MUI" was coined as the title for the matrix user interface.

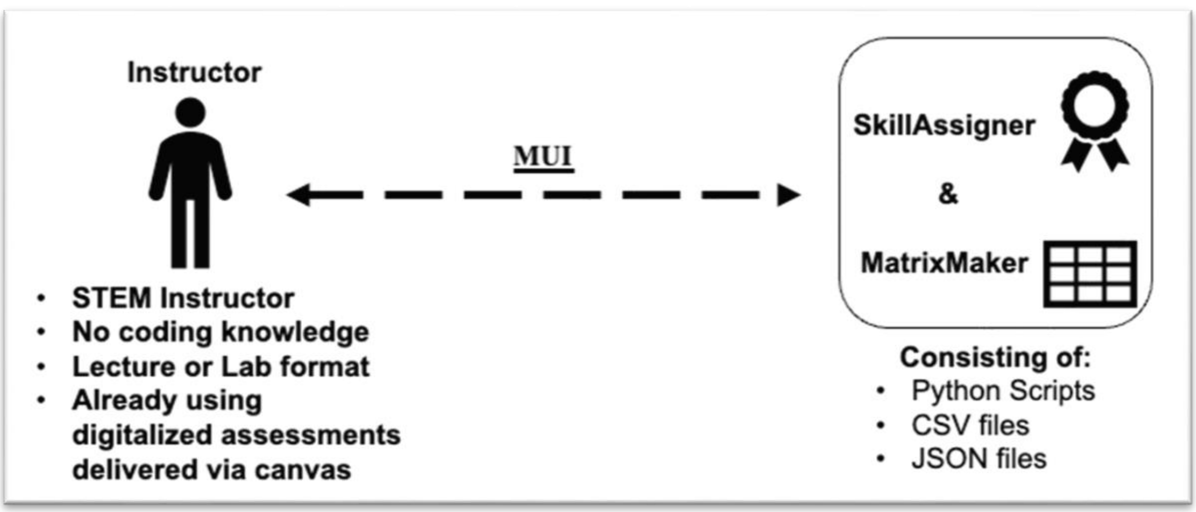


Figure 1: The Matrix User Interface (MUI)

MatrixMaker

In developing the application that is synonymous for the extraction phase the name "MatrixMaker" was coined, due to the systems reliance on a data matrix that is used to tag skills to each question within the assessments. The idea for developing the application was to have some way of extracting questions from canvas to store in a way that skills can be visually tagged to questions. As with most projects, the goal is to achieve a functionable product as a proof of concept, prior to putting it into ease of use for non-technical users. Therefore, the initial design was dependent on a "script overseer" such as a graduate teaching assistant with programing experience, to run the applications via accessing the source code. For data mining to occur, the application necessitates access to the Canvas LMS. This means that the user would have to input a bearer access token, and URL course specific values to the source code. In addition to this, the user would have to manually open the matrix of questions to tag each question with skill(s). There were two main concerns with this, first, the user would have to feel comfortable with modifying Python programming language code, and the second, is that the person who is tagging the questions must make sure that the skills are typed in correctly without any accidental characters. Prior to the development of the MUI, functional requirements of tagging skills were achievable, but demanded the expertise of an experienced individual with enough time to verify that all tagged skills were inputted correctly [4]-[5].

This paper introduces the implemented design, presented in Figure 1. It aims to deliver a streamlined approach that makes the application viable to a wide range of instructors. While designing it became clear that the most time consuming and most error-prone portion of the system is tagging the skills to the questions. Therefore, since the MatrixMaker is directly responsible for creating the data matrix upon which skills are tagged to questions it made sense, from both the user and developer points of view, to build a user interface that integrates with the tagging of the skills to each question. This is important for two main reasons, one the user no longer has a potentially huge data matrix that is relatively unpleasant to view on a question-by-question basis, as well as mentioned earlier, the need to input text into the cells in the matrix to correlate skills to questions.

When modifying the application, there were a couple main focuses in mind, the first was to make sure that the application was easy to use, and that it was relatively simple in design. With the demand to have the first version available in less than four months, it was integral to have a simple design for development. After discussing with advisors that are the targeted users of the product, as well as reviewing what was most reasonable on the development side, the product would consist of a simple page that displays a question and an option to select the skills that apply. Therefore, the user interface only needed to display the question, present all the skills as check buttons to select all that apply, a next button, and a quit button.

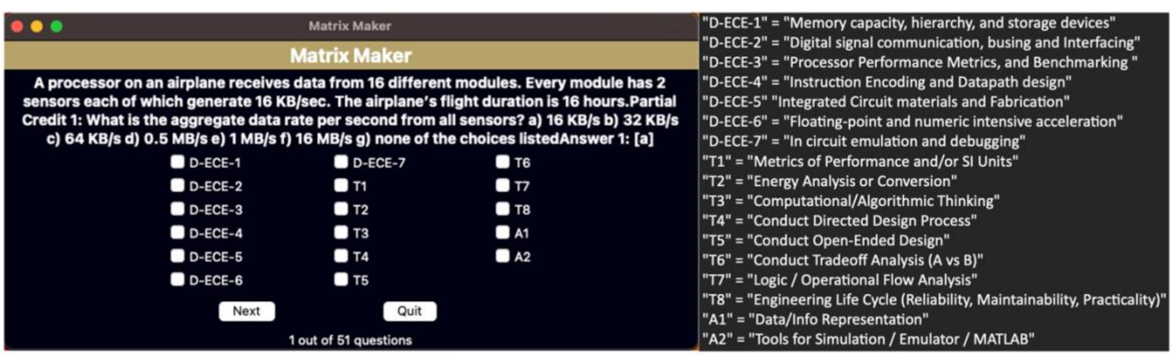


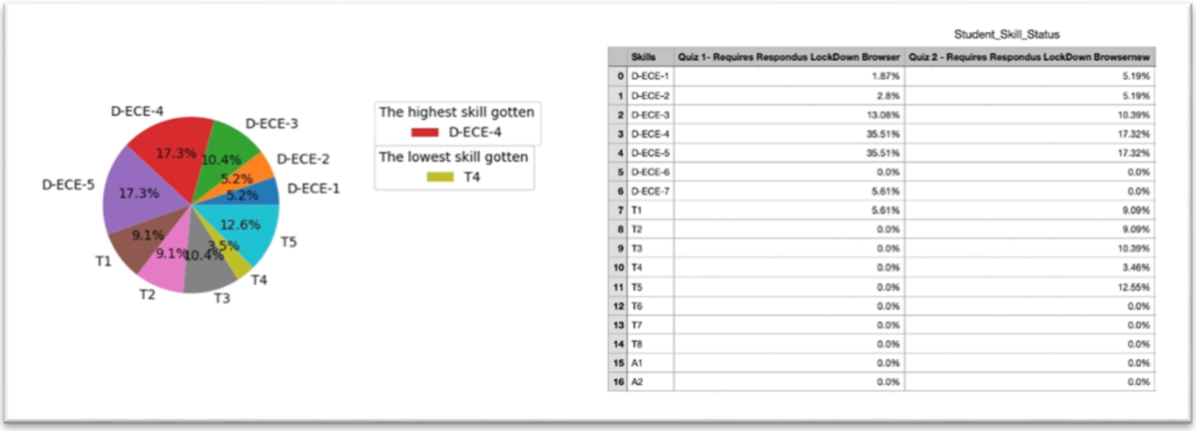
Figure 2: MatrixMaker Interface

Figure 2 illustrates the MatrixMaker application utilizing a universally understandable interface with acceptable colors. The flow that the program uses is that it begins with the first question in the assessment and lists all the instructor specified skills for the course with check buttons. The idea is to have the user iteratively traverse each question in the specified assessment through the interface scene in Figure 2. Keeping with the initial idea for tagging questions, the user interface is continuously updating the matrix of questions with their specified skills behind the scenes. If the user wishes to edit or view the matrix, they are free to do so, however this graphical user interface was designed with the intent of minimizing direct user interaction with the matrix or code. When developing the interface and conducting informal interviews with target audiences, there were some variabilities in opinions on how the design should have been. Nevertheless, in the pursuit of creating a product that satisfies a broad range of users, those consulted were able to reach a mutually agreeable compromise. For instance, one of these individuals wished to utilize a drag-and-drop feature to assign skills to questions individually by moving them to a specific column on the screen, while another preferred the convenience of a checklist displaying all skills alongside the questions. As a result, the compromise involved presenting questions one at a time with the inclusion of checkbox buttons.

SkillAssigner

The next module was named "SkillAssigner" that is continuously executed throughout the duration of the semester for each assessment, as it follows the MatrixMaker after students have taken the given assessment. In the foundational work, the SkillAssigner was developed to mine the student statistics on performance for a given assessment, calculate the skills if any for each assessment, and output a matrix with each student and what skills they demonstrated (if any) for a particular quiz-based assessment. As mentioned with the MatrixMaker, the SkillAssigner also required the user to have experience to specify the given assessment in the source code. Overall, the SkillAssigner effectively fulfilled its primary function, which was to compute and provide an assessment of a student's performance based on the skills demonstrated in a specific quiz-based assessment. Nevertheless, it is worth mentioning that while having direct access to the skills of each student in the assessment is valuable, it presents a challenge for instructors when it comes to assessing the overall performance of the course using a list of students and their respective skills [4]-[5].

When enhancing the provided framework, the objective is to create a system that is not only user-centric but also one that effectively supports the overarching goal of enhancing the educational experience. A survey was conducted to assess the features that would help instructors manage the course. Out of five instructors that were surveyed in terms of what they would want the SkillAssigner to provide, all of them stated that having an easy to view graphic displaying how the whole class is doing would be most ideal. Established on the feedback received from the participants, a pie chart has been disseminated, accompanied by an update to a dynamic log that provides a display of course performance for each assessment on which the SkillAssigner performed analysis on.



	Skills	Quiz 1- Requires Respondus LockDown Browser	Quiz 2 - Requires Respondus LockDown Browsernew
0	D-ECE-1	1.87%	5.19%
1	D-ECE-2	2.8%	5.19%
2	D-ECE-3	13.08%	10.39%
3	D-ECE-4	35.51%	17.32%
4	D-ECE-5	35.51%	17.32%
5	D-ECE-6	0.0%	0.0%
6	D-ECE-7	5.61%	0.0%
7	T1	5.61%	9.09%
8	T2	0.0%	9.09%
9	T3	0.0%	10.39%
10	T4	0.0%	3.46%
11	T5	0.0%	12.55%
12	T6	0.0%	0.0%
13	T7	0.0%	0.0%
14	T8	0.0%	0.0%
15	A1	0.0%	0.0%
16	A2	0.0%	0.0%

Figure 3: SkillAssigner's Summary Result View

In addition to the automated generation of a pie chart depicting skill achievement distribution, it was stated by an instructor who focuses on human-computer interaction, that pushing the most important data to the user is fundamental to the process of user acceptance. Consequently, aside the pie chart has the highest and lowest achieved skills from the course to make it easier for the user to gather the most important statistics faster than tediously traverse the statistics. Accompanying the pie chart is a data matrix shown in Figure 3. The file consists of a column with all the skills for the course, and the preceding columns consists of the quiz-based digital assessments the application was ran on. The SkillAssigner application was developed to make a dynamic matrix of student achievable skills. Therefore, if an instructor decides to add another skill for an assessment, the newly specified skill will be appended to list of already listed skills. The benefit is to provide a visual presentation of the class performance consolidated into a pie chart and matrix file appended throughout the semester to track class performance as the instructors manage the course.

Evaluation and Results

The evaluation consisted of various STEM instructors that resemble the target audience for the micro-credentialing system. The key features that are shared amongst the instructors that were surveyed are that they are all in STEM, invested in improving higher education, and currently or planning to implement digitalized quiz-based assessments. The responses collected via the instructors consisted of answers to four survey questions, which involved asking the participants that they would rate the user interface from one to ten, the likeliness of implementing the system in their course(s), comments, or suggestions, and if they have seen any similar product or service. The survey revealed that 50% of the participants rated the user experience a 10 out of 10, 37.5% rated it a 9 out of 10, and 12.5% rated it seven out of 10. Overwhelmingly 62.5% of the participants stated that they see themselves "highly likely" and the other 37.5% said "likely" to use something like this in their course. The most interesting responses came from the participants' comments or suggestions. For instance, one participant stated that they see this framework as a tool that could be used to highlight skills in a course to be public to students, for making the course selection process simpler on the student side. More specifically, the participant stated that "this could help to fill the knowledge gap for those students who do not have some of the skills needed for the course and skip concepts that they are already familiar with." Moreover, another participant stated that "the ability to tag questions with skills is novel. Displays created also highlight top-level outcomes which are of interest to an instructor. These features support UI capabilities which would be a positive aspect to utilize and do so in a direct and usable format." All the participants also stated that they have not seen a similar product or service [4]-[5].

In Figure 4, there are three distinguishable skill acquisition trends which AchieveUp can reveal by analyzing the number of students who demonstrated each skill as the course progressed. For instance, Figure 4 shows 'Computational/ Algorithmic Thinking', 'Conduct Open-Ended Design', and 'Conduct Tradeoff Analysis (A vs B)' respectively as three unique skillsets obtainable in a required undergraduate course on Computer Organization, which was utilized in the analysis of student data. Figure 4 (a) presents a case where skill T3 is acquired *linearly* at a constant rate through the semester, signifying that the competency of the skill among students increases somewhat uniformly as the semester proceeds. On the other hand, Figure 4 (b) displays a *front-loaded* skill acquisition flow which occurred for skill T5. A front-loaded acquisition signifies that most, or in this case all, of the students who obtained this skill had acquired it on the first coverage of the topic, and no additional learners subsequently demonstrated the skill. Automated identification of such a scenario can provide a significant opportunity for the instructor to focus on enhancing the acquisition of this skill which employers are interested in the latter phases of the course. As another case-in-point, a *back-loaded* skill acquisition seen in Figure 4 (c) exhibiting T6 represents that nearly all students who acquired the skill did so at the end of the course. This can help to inform instructors regarding the provision of additional resources in the earlier stages of the course to broaden student learning experience. Taken together, skill acquisition reporting facilitated by data mining of digitized assessments can offer new insights for both instructors and future employers.

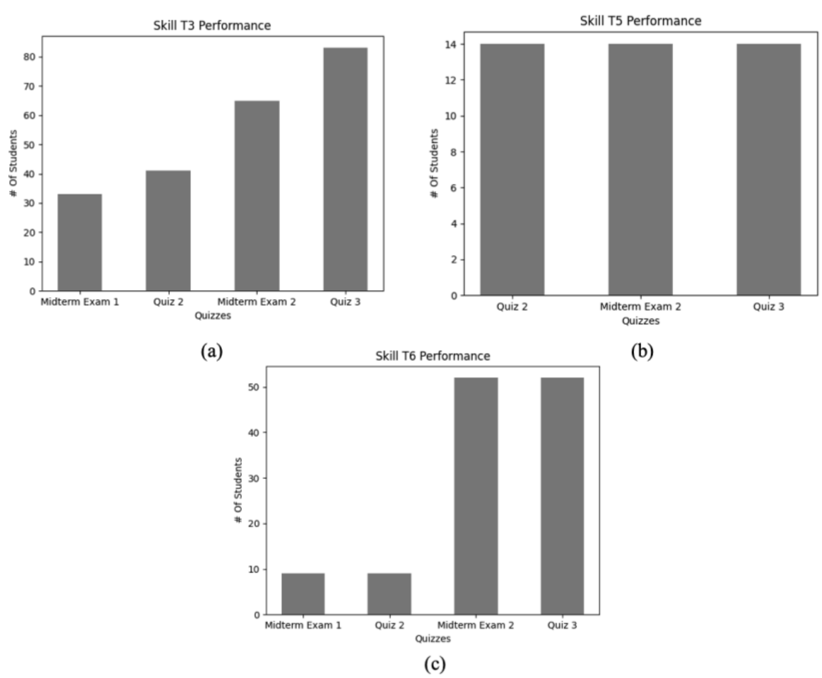


Figure 4: Three Identifiable Trends in Skill Acquisition: (a) Linear Skill Acquisition, (b) Front-Loaded Skill Acquisition, and (c) Back-Loaded Skill Acquisition.

Acknowledgments

This research was supported by NSF HRD/EHR-1953606. The first author thanks participants for reviewing and offering technical insights. Special acknowledgment is extended to the direct advisors, including Dr. Ronald DeMara, Dr. Laurie O. Campbell, and Dr. Ozlem Garibay.

References

[1] "Canvas credentials digital badges for higher ed," Instructure. [Online]. Available: https://www.instructure.com/higher-education/products/canvas/canvas-credentials-digital-badges?domain=badgr. [Accessed: 23-Nov-2022].

[2] Digital Credential Network powered by Badgr Pro. [Online]. Available: https://instructure.badgr.com/public/organization/badges. [Accessed: 23-Nov-2022].

[3] "Canvas Badges & Canvas certificates," Accredible. [Online]. Available: https://www.accredible.com/integrations/canvas-badges-canvas-certificates. [Accessed: 23-Nov-2022].

[4] P. Amoruso, R. F. DeMara, L. O. Campbell, F. Hernandez, and A. Mejia, "Personalizing Digitized Assessments and Remediation using an Automated Micro-Credentialing Framework," Online Learning Consortium (OLC) 2022 Conference on Accelerating Online Learning Worldwide, Orlando, FL, Nov. 15, 2022.

[5] P. Amoruso, 'Micro-Credentialing with Fuzzy Content Matching: An Educational Data-Mining Approach', University of Central Florida Orlando, Florida, 2023.

[6] S. A. Ali and R. Khan, 'Harnessing the Digital Nature of Micro-Credentials in Higher Education', 2023.

[7] M. Mallon, 'Micro-credentialing', *Public Services Quarterly*, vol. 15, no. 2, pp. 116–125, 2019.

[8] B. Fishman, S. Teasley, and S. Cederquist, 'Micro-credentials as evidence of college readiness: Report of an NSF workshop', 2018.

[9] D.-K. Mah, N. Bellin-Mularski, and D. Ifenthaler, 'Moving forward with digital badges in education', *Foundation of Digital Badges and Micro-Credentials: Demonstrating and Recognizing Knowledge and Competencies*, pp. 511–517, 2016.

[10] J. Rujimora, L. O. Campbell, and R. F. DeMara, 'Exploring the Student-to-Faculty Ratio and Degree Attainment in Florida', *Journal of Hispanic Higher Education*, p. 15381927231172583, 2023.

[11] C. L. Bruder, '101 UP: Implementation of a Gamified Curriculum to Increase Self-Regulated Learning Skills and Motivation for At-Risk Students in a First-Year Experience Course—An Action Research Study', University of South Carolina, 2020.

[12] R. F. DeMara, N. Khoshavi, S. Pyle, J. Edison, R. Hartshorne, B. Chen, M. Georgiopoulos, "Redesigning Computer Engineering Gateway Courses using a novel Remediation Hierarchy," in Proceedings of American Association for Engineering Education National Conference (ASEE-16), New Orleans, LA, USA, June 26 – 29, 2016.

[13] B. Chen, R. F. DeMara, S. Salehi and R. Hartshorne, "Elevating Learner Achievement Using Formative Electronic Lab Assessments in the Engineering Laboratory: A Viable Alternative to Weekly Lab Reports," in *IEEE Transactions on Education*, vol. 61, no. 1, pp. 1-10, Feb. 2018, doi: 10.1109/TE.2017.2706667.

[14] G. Ghaith and H. Yaghi, 'Relationships among experience, teacher efficacy, and attitudes toward the implementation of instructional innovation', *Teaching and Teacher education*,

vol. 13, no. 4, pp. 451–458, 1997.
[15] "Tkinter - Python interface to TCL/TK," tkinter - Python interface to Tcl/Tk - Python 3.11.0 documentation. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed: 23-Nov-2022].