Five Years of Graduate CS Education Online and at Scale

David A. Joyner
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA
david.joyner@gatech.edu

Charles Isbell
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA
isbell@cc.gatech.edu

Thad Starner
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA
thad@cc.gatech.edu

Ashok Goel
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA
goel@cc.gatech.edu

ABSTRACT

In 2014, Georgia Tech launched an online campus for its Master of Science in Computer Science program. The degree, equal in stature and accreditation to its on-campus counterpart, offered a notably lower cost of attendance. Its design emphasized flexibility in both geography and time, allowing students from around the world to earn a highly-ranked MSCS without taking time off work or moving to campus. Five years later, the program enrolls over 8000 students per semester and has graduated 1500 alumni. It is believed to be the largest program of its kind and has received recognition from national organizations on professional education. Existing research on the program has focused on challenges and opportunities to scale that are agnostic to the content itself. In this reflection, we look at the creation and growth of the program as it relates to graduate-level CS instruction. In particular, we note a unique and powerful unity of content and platform: the online delivery of the program dovetails with the technical skill-sets of the professors and students that it draws, putting both in the position to contribute and innovate.

ACM Reference Format:
David A. Joyner, Charles Isbell, Thad Starner, and Ashok Goel. 2019. Five Years of Graduate CS Education Online and at Scale. In ACM Global Computing Education Conference 2019 (CompEd '19), May 17–19, 2019, Chengdu,Sichuan, China. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3300115.3309534

1 Introduction

Higher education in the 2010s has been characterized in large part by two major trends. The first has been the arrival and widespread deployment of Massive Open Online Courses, or MOOCs. MOOCs first appeared in 2010 and quickly rose in visibility, culminating in New York Times dubbing 2012 the "Year of the MOOC" [28]. That early hype was met with early skepticism as well, and MOOCs have since followed a traditional hype cycle, but in more recent years they have begun to find applications in for-credit educational environments [16][17][28].

A second well-documented trend has been the rising cost of college [18], and the accompanying meteoric rise in student debt burdens [25]. On the surface, it appears that one of these trends may address the other: MOOCs aim to make higher education far more accessible and affordable.

It is against the backdrop of these two trends that in 2014, Georgia Tech, a major public research university in the United States created the first MOOC-based graduate degree. By design, the program was more like traditional distance learning than MOOCs: students applied for admission, paid to enroll in semesters, were evaluated by human graders, received letter grades, and earned a fully-accredited diploma. The program capitalized on the strengths of MOOCs; however: instruction was delivered on a popular MOOC platform and designed to be consumed asynchronously, and enrollment cost was lowered to less than one-ninth of the out-of-state cost (one-third of the in-state cost), regardless of the state or country in which the student resided.

Five years later, the program has thrived. It has grown to over 8,000 students in the Fall 2018 semester, and it is projected to increase the world's output of MSCS graduates by 8% annually [10]. It was recognized by the University Professional and Continuing Education Association for program excellence, and has spawned numerous similar programs, both at the university and on other platforms such as edX and Coursera.

In this paper, we trace back the motivation and creation of this new curriculum initiative. While much has already been written about the program, most existing research has focused on elements of the program that are domain-neutral. In this overview, we will concentrate as much as possible on those elements of the program unique to its nature as a computer science graduate degree. Of special note will be the frequency with which innovative initiatives occur specifically due to the technical backgrounds of the students and instructors in the program.

CompEd '19, May 17–19, 2019, Chengdu, Sichuan, China

ACM ISBN 978-1-4503-6259-7/19/05…$15.00
https://doi.org/10.1145/3300115.3309534

2 Program Motivation

As noted above, the primary context under which the program was developed was the rise of college prices and the emergence of MOOCs as a possible means through which to deliver education affordably by leveraging scale. These trends may address any similar affordable online program, and indeed some of the new initiatives that have followed on this program's success are in less technical fields like public health, business administration, and accounting. Specifically with regard to computer science education, however, two additional trends contributed to the inception of this program.

The first is the increasing need for additional lifelong learning, especially within technical fields. The "fourth industrial revolution" as it has been dubbed [32] has seen an explosion in job opportunities in computing-oriented fields [22]. This trend, coupled with automation and outsourcing driving additional changes in the job market [2], has led to a dramatic increase in computing as a new career choice. Even for those already within technical fields, the rapid changes in the technology industry mean that employees need to re-skill regularly to stay current. A Bachelor's degree in Computer Science from many schools in the early 2000s would likely not have covered machine learning, cloud computing, computer vision, and more topics that are spawning entire new job categories today. These learners, however, are often in the middle of their careers with families: they cannot afford the opportunity cost of going two years without a salary to obtain a graduate degree, let alone the high price of enrollment.

There was thus a demand for rigorous, respected lifelong education specifically in computing from both students and industry, but too much friction in the existing mechanisms to meet either demand. Startups have been quick to provide bootcamps, MOOCs, and other non-traditional credentials to address this gap, but research has found that these alternate methods typically focus only on a small subset of the skills delivered by traditional programs [35].

A second significant motivating trend in the program's inception was a more straightforward desire to continue to innovate in the emerging area of online learning. Georgia Tech had been at the forefront of these initiatives, highlighted by its early participation in one of the first MOOCs as well as its decades-old distance learning division. Given the nature of the delivery mechanism, an initial focus on computer science was logical as the individuals involved in teaching in the program could also contribute to its technological development.

Thus, the program can be seen as largely motivated by four trends: the rising cost of higher education; the emergence of MOOCs as a possible new delivery mechanism; the increasing need for lifelong CS education (driven by both student and industry demand); and the desire to continue to innovate on technological mechanisms for delivering high-quality for-credit education. This motivation, of course, is in some ways teleological: the full nature of the motivation behind the program's inception is more complex; however, these motivations were all present, and moreover, all directly connect to the program's eventual success.

3 Program Development

It is useful to think of the development of the program in terms of three major stages. The first stage, creation, occurred before the initial students began the program and provided the initial foundation. The second stage, experimentation, covered the first few years and saw expansion coupled with experimentation in production and delivery methods to find the procedures most effective for scale and quality. The third stage, normalization, began within the past year and sees an increasing focus on standardizing practices, maintaining growth, and optimizing existing systems. Although there are formal milestones that may be marked along the way—such as the matriculation of the first class in January 2014 and the graduation of the first students in December 2015—these stages are interpretive. Experimentation continues in the program today, but now there exists a set of principles and practices for delivery that did not exist when the program began and were derived based on that early experimentation.

3.1 Creation

Creation of the program began in the summer of 2013. The university partnered with a startup specializing in creating MOOCs to develop its initial set of five courses to launch in Spring 2014, including one taught by one of this paper's co-authors, Isbell. Courses were generally developed by three-person teams: a professor, a course developer, and a video producer. The professor was responsible for authoring and filming all course content. The course developer worked with the professor to convert the content to work in the traditional MOOC presentation style, characterized by short videos, frequent interspersed exercises, and pen-based presentation (wherein professors would write on a virtual "whiteboard" while narrating for students). The course developer also assisted the professor in developing the syllabus and assignments, authoring autograding tools, and typically served as the course's teaching assistant during its inaugural semester. The video producer filmed and edited sections of the course where professors were on camera, as well as edited the virtual whiteboard portions of the course.

The majority of course content was presented in this virtual whiteboard format, which carried several benefits. First, because the recording setup required only a single pre-configured setup, professors were able to film on their own schedules rather than coordinate with the video editor to be present and working during every filming session. This approach allowed professors more autonomy over production scheduling and maximized video producers' time. Second, because the professor was typically not on screen, content could be filmed in very small chunks. A professor could attempt a sentence several times until they were satisfied, and then move on to the next sentence. A single bad take meant re-filming only a few seconds, not multiple minutes, allowing the finished product to be more professional.

Pre-production allowed additional benefits as well. In one course from the original set of five, Isbell presented the course along with a prominent colleague from another university. In an on-campus class, it would be entirely infeasible to have another professor—especially from another university—co-teach a class

every semester for five years, but by pre-producing the material and delivering it online, those different perspectives could not only be included, but included in a conversational style. The asynchronous and remote nature of the course could allow another professor to be involved in the delivery as well, an opportunity realized to a greater extent during the subsequent experimentation phase.

3.2 Experimentation

The experimentation phase began with the inaugural semester of the program in spring 2014, where 300 students enrolled in that initial batch of classes. This semester was intentionally constructed as a "beta" semester to ensure the feasibility of the program under optimal conditions, with classes no larger than their on-campus counterparts. Upon the success of the initial semester, the program was opened more broadly in fall 2018, including students deferred from the first semester. In the lead-up to this semester, five additional courses were developed, including one by two additional co-authors of this paper (Goel and Joyner).

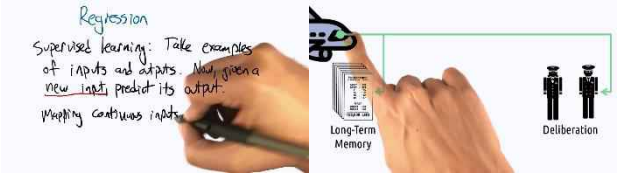

Figure 1. A still from one of the early courses produced in the program (left), and one from a later course (right). Over time, the production workflow shifted from the virtual whiteboard with a working pen to an emphasis on produced diagrams, visuals, and animations.

As course delivery began, experimentation focused on ways to address the myriad of challenges to scaling a program while keeping all of the rigors and procedures associated with an accredited degree. Likely the most significant development, though, was the discovery that online students could be relied upon to work as teaching assistants to support the program's growth. Early semesters relied upon on-campus students, but not enough such students were available to support the program's growth, and the cost of their tuition waivers and stipends was prohibitively expensive compared to the inexpensive price paid by online students. We hypothesized that online students would be too occupied with work, family, and coursework obligations to work in the role, but found that enough were interested to support the program's growth. Additionally, they were more suited to give better feedback given their professional backgrounds, and they were motivated by more intrinsic and altruistic factors than extrinsic [15]. Compounding this benefit was the realization that online students never truly "leave" campus: students may consider working as teaching assistants after graduation. Of the 250 teaching assistants hired for fall 2018 at time of writing, 19% are alumni of the online program, and 43% are present students in the program. 28% of the remaining teaching assistants are on-campus MS students, and 10% are on-campus PhD students. This ability to hire alumni—and apparent interest from alumni in being hired—provides a pool of potential teaching assistant candidates that will continue to grow over time even as program enrollment ultimately stabilizes.

Additionally, early feedback from students in the initial semesters led to changes in the course production process. One course in this second phase of development experimented with avoiding the virtual whiteboard and instead pre-producing all course visuals. During recording, the professor retained the ability to point to elements on the screen with a recorded hand, similar to an instructor pointing at elements on a slide. Although there was skepticism that this change would lead to a dry presentation of bullet-point slides, the approach was ultimately successful as it allowed significant resources to be invested into producing high-quality, previously reviewed, engaging visuals [27]. This approach has since become the standard approach for creating new courses for the program, and video producers—previously responsible solely for post-production—began to take on the roles of graphic designers and artists. Course developers, in turn, began focusing more attention on translating the professors' vision into descriptions of visuals for the video producers to create even without subject matter expertise. This change then freed professors to focus entirely on their manner of presenting during recording rather than having to attend to screen layout, virtual ink color, and other distractions from speaking. Figure 1 shows typical visuals from an early course and a more recent course produced for the program.

Other areas of experimentation include forum management, office hour delivery, and grading management. These early experiments have given way to a "toolbox" of approaches to delivering different classes based on the specific requirements of their content, students, and teaching teams. A comprehensive view of the variety of different approaches developed through this phase is available in prior work [16]. The initiatives described in section 4 below also took place during this phase.

In many ways, this experimentation phase is a growth phase as well: Figure 2 shows the growth of the program during these semesters, measured in seats. A seat is a single student enrolling in a class: if one student takes two classes, they count as two seats. The figure portrays seats because a single student taking two classes requires no less work for the professors and teaching assistants than two students each taking one class. Thus, the program has grown from 10 courses and 1,828 seats in its first open semester in fall 2014 to 29 classes and 8,911 seats in spring 2018.

3.3 Normalization

The current phase in the program is modeled here as normalization. This label is not to suggest that experimentation has ended, but rather that a technological and procedural foundation has been laid. There are still challenges to be addressed and improvements to be made, but unlike in the experimentation phase, those are no longer seen as existential: whereas inability to scale feedback and assessment would threaten the very viability of the program, the challenges to be addressed now concern improving student outcomes and experiences, maintaining growth, and leveraging emerging opportunities.

Supporting that notion, the program continues to grow: this term, over 1500 students are expected to begin the program, more

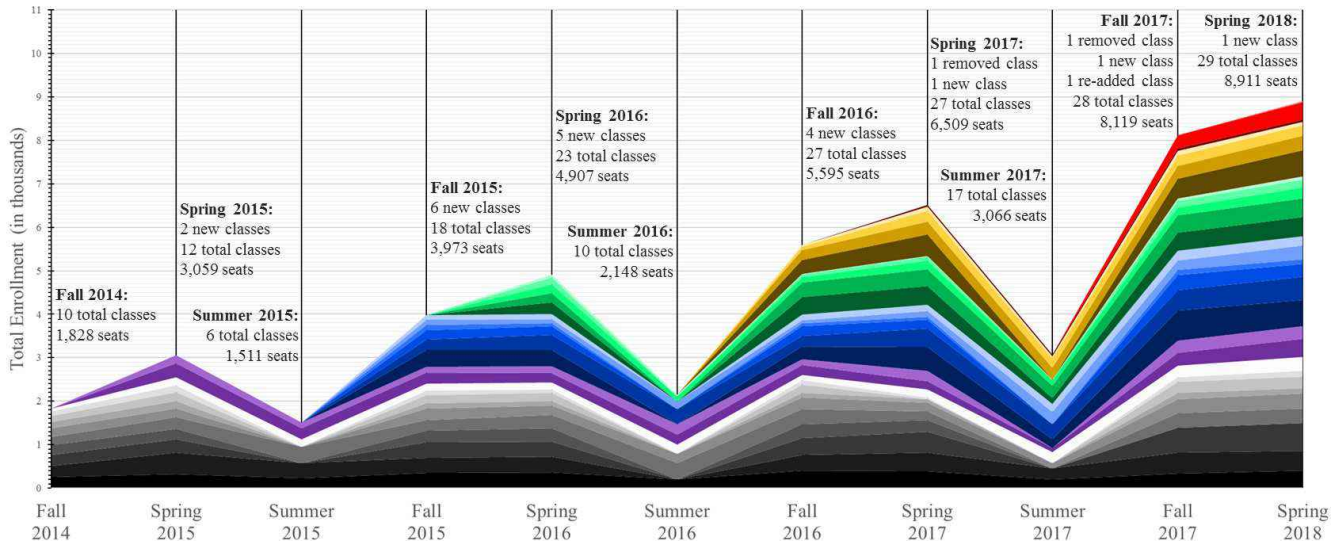

Figure 2. Growth of the program over time. Summer semesters are shorter, and thus some courses are not offered during summer. Each color represents a different course; color families represent the semester of launch for each course.

than the total number of alumni the program has generated so far. Graduation numbers have risen each semester since the inaugural class graduated in December 2015, and it is likely that matriculation and graduation numbers may ultimately balance out as they do on-campus (where the total enrollment capacity is dictated more strictly by housing and lecture hall capacity). That point has not been reached yet, however.

Thus, normalization focuses pedagogically on setting and communicating best practices across the program, and administratively on building out the supporting infrastructure behind the program. On the administrative side, the later years of the program's history have seen the hiring of a half-dozen academic advisers; multiple associate directors ded icated to topics like admissions, grievances, and the student experience; and greater relationships with other departments of campus covering academic integrity, student life, and alumni relations.

Pedagogically, much of this initial success is owed to the entrepreneurial nature of the courses and their professors: each has been active in experimenting and revising their own classes, discovering procedures and developing technologies that likely would never have been realized had requirements been communicated in a more top-down fashion. However, enough has been learned that a set of best practices has emerged: these may be challenged by future experiments, but such experiments should build from the experiences of these previous semesters. Moreover, there are specific details that must be filled in for every course: every course has its own deadlines, late policy, regrade policy, and so on. We are encouraging more standardization across these criteria so students may spend less time learning a course's specific delivery policies and more time learning the course's content.

The experimentation focuses now on building from that solid and shared foundation, but experimentation is underway. In fall 2018, for example, the program's peer review tool has been adjusted to share reviews across all reviewers of a particular assignment to gauge whether that shared foundation sparks more conversation or if the quality of students' feedback rises due to any perceived social pressure. In spring 2019, work is planned to quantify the biases that may emerge based on the name attached to a student in peer review; this controlled experiment would be able to quantitatively answer the extent to which a name associated with a certain gender or demographic group is prone to receive systematically better or worse numeric evaluation and feedback.

Other ongoing experimentation looks at the potential use of virtual reality to facilitate social presence in online learning and the increasing role that AI serves in providing students rapid feedback and answers to their questions. Those initiatives bring the program's development full circle back to one of its original motivations: the desire to innovate, and the unique ability of computing professors and students to contribute to both the program's content and its platform. This innovation echoes one of the major new opportunities in the program: how do we scale research opportunities for online students as we have scaled learning opportunities?

4 Technological Initiatives

Significant research has already been done on this program. However, much of that research has focused on elements of scale that are not unique to a computer science program, such as scaling human grading, office hours, peer grading, and forum management [3][12][15][16][19]. In teaching CS, however, there are unique additional demands absent from other fields, such as increased technological needs (e.g. cloud computing resources), unique types of integrity violations (e.g. code plagiarism), and increased focus on projects.

Those needs have given way to one of the program's distinct features: a unity of platform and skillset. As an online program,

all elements of instruction and assessment are delivered via technology. As a CS program, all faculty members and students are (or must become) proficient with the design and development of technology. As a result, we note that several of the solutions to the demands documented above have come from students or faculty teaching in the program rather than external partners, and moreover, that there are several opportunities in teaching online that these professors and students have been uniquely positioned to exploit.

4.1 AI for Forum Administration

Significant attention has been paid to forum administration, whether it be in for-credit online courses (e.g. [23]), MOOCs (e.g. [26]), or non-academic settings (e.g. [5]). Like MOOCs, for-credit at-scale courses have a massive number of interactions, but like for-credit courses, they carry the expectation that all questions will receive answers from official course staff. As graduate-level CS courses, these questions are often back-and-forth discussions to debug errors or expand knowledge, not solely isolated questions with singular answers.

While non-technical human-based solutions have been developed [16], several projects have applied AI to this challenge. In one of Goel's projects, an AI agent was developed based on previous semesters' Q&A patterns as well as a structured knowledge representation of course information to proactively answer student questions when a certain level of confidence in the answer could be achieved [9]. Starner has developed a tool that intelligently recommends related questions to students based on new questions they are asking in order to offload responsibility for finding repeated questions from students or teaching assistants. Both projects aim to allow students to get answers more quickly, either by proactively delivering an answer or making an existing answer easier to find, while also reducing the number of questions teaching assistants must answer individually.

Forums play an additional role regarding community-building in online courses and may be analyzed to signify overall student sentiment or identify students in need of individual intervention. For a class project, two students in the program individually leveraged sentiment analysis to accomplish these two tasks: one AI system flags individual students whose tone is seen trending negatively for instructor intervention, while another gauges overall classroom sentiment to synthesize for instructors points of contention or difficulty in the semester [31]. Another student used sentimental analysis to evaluate longitudinal changes in discourse in one of the program's courses [3].

4.2 Code Plagiarism Detection & Deterrence

Detecting code plagiarism is a commonly referenced problem in computer science education that has given rise to several projects (e.g. [1][4][21], among many others). While many are narrowly tied to a particular programming language or context, MOSS [29] covers multiple languages. However, detecting if two code files are similar is only the last stage of plagiarism detection at scale. The ubiquity of tools like Github for sharing work demands that plagiarism detection occur across semesters rather than just within singular assignment submissions, but these courses at scale retain thousands of submissions after multiple semesters. To support organizing and filtering the results for only pertinent pairs, one student for her class project developed a tool specifically for organizing archives of previous submissions, deliberately uploading pertinent sets for evaluation, and presenting results for efficient confirmation and action [32].

Detecting plagiarism relies on possessing both the original and the copied material, but no plagiarism detection solution can address the case of homework-for-hire services where the original is never available to anyone but the student ultimately submitting it as their original work. To combat this tactic, Starner has led a project to construct an AI agent to proactively identify suspected requests for homework-for-hire on a popular freelancing web site, with plans to extend that work to other services as well.

As an international program, the courses are also notably impacted by the documented difference in cultural perceptions of plagiarism [11]. To address this dynamic, one course has—in addition to employing MOSS to detect plagiarism and proactively having public code repositories removed—authored content with embedded formative assessment to instruct students on what behaviors are permitted and forbidden.

4.3 Peer Evaluation and Participation

Peer assessment has been heavily used in MOOCs, for both summative (e.g. [33]) and formative feedback (e.g. [20]). Research on this program, however, has noted the need—for accreditation and reputation—to rely on expert review rather than peer review for generating grades [15]. Peer review may nonetheless play a pedagogical or supporting role, however. Isbell has led a project wherein peer review of short answer responses is seeded with responses whose assessment is known. A probabilistic graphical model is then constructed to establish individual students' proficiency in evaluation, which can be used to generate grades for their peers or to evaluate those students directly [19]. Joyner led a project to modify an existing peer review platform to equip graders with the results of a round of peer review while grading, finding that doing so increased students' perceptions of the quality of feedback they received [12].

As part of a graduate-level CS program, the program also has courses focused on design or evaluation of interfaces. Toward this end, students may need to conduct surveys, interviews, or demonstrations and gather feedback. Classmates' participation in these is encouraged and scored for course credit. To support this process, a student under the direction of Joyner developed a platform for students to create surveys or evaluations and share them with classmates, whose participation is then recorded automatically for inclusion in their grade.

4.4 Automated Evaluation and Feedback

One major area of opportunity in online learning is the opportunity for AI-driven automated feedback. As instruction and assessment are already occurring in a computational interface, the platform is present to give students immediate evaluation in support of a rapid feedback cycle. To realize this potential, Goel and

Joyner developed the notion of nanotutoring, wherein small, highly specialized AI agents are developed for specific problems [6]. These nanotutors, inspired by the broad literature on intelligent tutoring systems, address problems narrow enough that the entire potential answer space may be mapped to feedback, but broad enough that student answers may have hundreds of variations. Goel and Joyner constructed from scratch over 100 such nanotutors in their course and have received positive feedback on the role this feedback plays in students' learning [7].

On the other end of the spectrum, automated evaluation is also possible on much larger projects of the outcome measures are objective. In one class taught by Goel and Joyner, students are asked to construct cognitive agents that attempt a human intelligence test [8]. Upon submission to the autograding platform, the students' agents are tested against a battery of problems they did not see while writing the agent. Students are then given the results of this test immediately so they may continue to iterate on and improve the agent. This project was used before in the course's on-campus counterpart, but the scale of the online program provided the incentive and resources to develop a sophisticated solution.

The platform on which this solution resides is general and available for use by other classes as well, which has given way to several new opportunities for innovation. Goel and Joyner capitalized on the formal structure and automated execution to study crowdsourced ideation for complex problems featuring novel solutions from students [13]. In Starner's course, students construct separate agents for search and for game-playing, the latter of which are then matched against each other in a tournament. Similar to crowdsourced ideation, this infrastructure combined with the program's scale and the instructors' research background has given rise to valuable research on the content itself. Notably, this infrastructure was all constructed by program staff for the program; no out-of-the-box solutions are used.

4.5 Student-to-Student Advising

Professors in this online program have noted the significant impact that the online platform allows individual students to have [14]. The forum-based environment allows every student to effectively have unlimited time "in front of" the class to share their thoughts without taking time away from the planned lecture. As a result, a powerful student community has emerged. While this community could be present in any similar program regardless of subject matter, the technical ability of these students has given rise to projects to support their classmates. The most significant among these is an unofficial student-run web app in which students review courses and read reviews of other students. The site evolved from a collaborative spreadsheet shared in the early days of the program, and although similar sites exist (e.g. RateMyProfessor), this site's close tie with the program's structure and student body drive significantly more traffic. To date, the site has received over 2,700 reviews, most connected with numeric assessments of the quality, difficulty, and workload for quick summarization. Students may filter reviews by semester, rating, and difficulty, and the course pages automatically pull from the school's publicly-available grade database to augment the reviews. The entire effort has been performed completely independently by the student community with no support from the school, and the project has changed hands multiple times since its inception. The platform has even served as the target for follow-up studies; one student used sentiment analysis on the platform's reviews and compared them to official university reviews, finding that these public classmate-targeted reviews showed more negative sentiment than private professor-targeted reviews [24].

5 Persistent Challenges

While the program has largely addressed existential threats, there are still challenges and opportunities. One major issue is course maintenance. Most courses in the program are built on top of a MOOC with hundreds of videos. Over 14,000 individual videos have been produced, most around 3 minutes in duration. Most professors will record a lecture (which is then split into smaller videos) in a single sitting, lending continuity to the presentation. If later content is to be added or revised, the process typically demands re-recording the entire lecture. Joyner experimented with focusing on video independence and modularity and has found the course easier to maintain. This solution requires a fundamental shift in the initial development paradigm that is less natural than the standard presentation style.

Forum administration remains an issue. Although the forum workflows documented in prior work [16] and the AI tools described in section 4 help, forums for large classes still become difficult for both students and teaching teams. While AI may find duplicate questions within a semester or proactively answer routine questions, a graduate CS program is characterized in part by discussion. Constructing an AI to participate in discussion is more difficult challenge than clustering or pairing answers to questions. Developments are underway to create a forum built with the needs of a large, for-credit online course in mind.

Finally, much of this analysis has focused on delivery of individual courses. However, the program exists as part of a broader ecosystem, from which it draws value. Elements of that broader ecosystem are strained by the program's growth. Academic advising, for example, is needed by the program's students. Institute policies exist for documenting family emergencies or illnesses, tracking and investigating integrity violations, and addressing students' grievances. In many cases, technological solutions are possible, but many of these require a human's direct involvement in each case. Scaling these processes remains a challenge.

6 Conclusion

This paper has recounted the motivation, creation, expansion, and normalization of an online graduate program in computer science. While in some ways it is a straightforward evolution of traditional distance learning, it is distinct in its approach to leveraging the lessons learned from the rise of MOOCs and the extent to which it takes advantage of the online environment to improve the student experience. In many ways, the scale, asynchronicity, and remoteness of the program have proved to be opportunities rather than obstacles: they allow the creation of a 24/7 classroom where students may interact with one another any without permanently missing any lecture or discussion.

REFERENCES

[1] Aleksi Ahtiainen, Sami Surakka, and Mikko Rahikainen (2006). Plaggie: GNU-licensed source code plagiarism detection engine for Java exercises. In *Proceedings of the 6th Baltic Sea Conference on Computing Education Research: Koli Calling 2006*, 141-142. ACM Press.

[2] Marc Beylerian & Brian H. Kleiner (2003). The downsized workplace. *Management Research News, 26*(2/3/4), 97-108.

[3] Ida Camacho & Ashok Goel. (2018, June). Longitudinal trends in sentiment polarity and readability of an Online Masters of Computer Science course. In *Proceedings of the Fifth Annual ACM Conference on Learning @ Scale*. ACM Press.

[4] Georgina Cosma & Mike Joy (2012). An approach to source-code plagiarism detection and investigation using latent semantic analysis. *IEEE Transactions on Computers, 61*(3), 379-394.

[5] Kushal Dave, Martin Wattenberg & Michael Muller (2004, November). Flash forums and forumReader: navigating a new kind of large-scale online discussion. In *Proceedings of the 2004 ACM conference on Computer-Supported Cooperative Work*, 232-241. ACM Press.

[6] Ashok Goel & David A. Joyner (2016). An Experiment in Teaching Cognitive Systems Online. In Haynes, D. (Ed.) *International Journal for Scholarship of Technology-Enhanced Learning 1*(1).

[7] Ashok Goel & David A. Joyner (2017). Using AI to Teach AI: Lessons from an Online AI Class. *AI Magazine 38*(2), 48-59.

[8] Ashok Goel, Maithilee Kunda, David A. Joyner & Swaroop Vattam (2013). Learning about Representational Modality: Design and Programming Projects for Knowledge-Based AI. In *Fourth AAAI Symposium on Educational Advances in Artificial Intelligence*, 1586-1591.

[9] Ashok Goel & Lalith Polepeddi (2016). Jill Watson: A Virtual Teaching Assistant for Online Education. Georgia Institute of Technology Technical Report. Retrieved from https://smartech.gatech.edu/handle/1853/59104

[10] Joshua Goodman, Julia Melkers & Amanda Pallais (2019). Can online delivery increase access to education? *Journal of Labor Economics, 37*(1), 1-34.

[11] Niall Hayes & Lucas D. Introna (2005). Cultural values, plagiarism, and fairness: When plagiarism gets in the way of learning. *Ethics & Behavior, 15*(3), 213-231.

[12] David A. Joyner, Wade Ashby, Liam Irish, Yeeling Lam, Jacob Langston, Isabel Lupiani, Mike Lustig, Paige Pettoruto, Dana Sheahen, Angela Smiley, Amy Bruckman, & Ashok Goel (2016). Graders as Meta-Reviewers: Simultaneously Scaling and Improving Expert Evaluation for Large Online Classrooms. In *Proceedings of the Third Annual ACM Conference on Learning at Scale*. Edinburgh, Scotland. ACM Press.

[13] David A. Joyner, Darren Bedwell, Chris Graham, Warren Lemmon, Oscar Martinez & Ashok Goel (2015). Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems in Intelligence Tests. In *Proceedings of the Sixth International Conference on Computational Creativity*. Provo, Utah.

[14] David A. Joyner, Ashok Goel & Charles Isbell (2016). The Unexpected Pedagogical Benefits of Making Higher Education Accessible. In *Proceedings of the Third Annual ACM Conference on Learning at Scale*. Edinburgh, Scotland.

[15] David A. Joyner (2017). Scaling Expert Feedback: Two Case Studies. In *Proceedings of the Fourth Annual ACM Conference on Learning at Scale*. Cambridge, Massachusetts. ACM Press.

[16] David A. Joyner (2018). Squeezing the Limeade: Policies and Workflows for Scalable Online Degrees. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. London, United Kingdom. ACM Press.

[17] David A. Joyner (2018). Toward CS1 at Scale: Building and Testing a MOOC-for-Credit Candidate. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. London, United Kingdom. ACM Press.

[18] Rita Kirshstein (2013). Rising tuition and diminishing state funding: An overview. *Journal of Collective Bargaining in the Academy*, (8), 14.

[19] Pushkar Kolhe, Michael L. Littman & Charles L. Isbell (2016, April). Peer Reviewing Short Answers using Comparative Judgement. In *Proceedings of the Third Annual ACM Conference on Learning at Scale*. Edinburgh, Scotland. ACM Press.

[20] Chinmay E. Kulkarni, Michael S. Bernstein, & Scott R. Klemmer (2015, March). PeerStudio: rapid peer feedback emphasizes revision and improves performance. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, 75-84. ACM Press.

[21] Cynthia Kustanto & Inggriani Liem (2009, May). Automatic source code plagiarism detection. In *Proceedings of the 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, 481-486. IEEE.

[22] Frank Levy & Richard J. Murnane (2005). *The new division of labor: How computers are creating the next job market*. Princeton University Press.

[23] Margaret Mazzolini & Sarah Maddison (2003). Sage, guide or ghost? The effect of instructor intervention on student participation in online discussion forums. *Computers & Education, 40*(3), 237-253.

[24] Heather Newman & David A. Joyner (2018). Sentiment Analysis of Student Evaluations of Teaching. In *Proceedings of the 19th International Conference on Artificial Intelligence in Education*. London, United Kingdom. Springer.

[25] Elvira Nica & Catalina-Oana Mirica (2017). Is higher education still a wise investment? Evidence on rising student loan debt in the US. *Psychosociological Issues in Human Resource Management, 5*(1), 235.

[26] DFO Onah, Jane E. Sinclair & Russell Boyatt (2014, November). Exploring the use of MOOC discussion forums. In *Proceedings of London International Conference on Education*, 1-4.

[27] Chaohua Ou, Ashok Goel, David A. Joyner & Daniel Haynes (2016). Designing Videos with Pedagogical Strategies: Online Students' Perceptions of Their Effectiveness. In *Proceedings of the Third Annual ACM Conference on Learning at Scale*. Edinburgh, Scotland.

[28] Laura Pappano (2012, November 4). The Year of the MOOC. The New York Times (p. ED26).

[29] Cathy Sandeen (2013). Integrating MOOCs into traditional higher education: The emerging "MOOC 3.0" era. *Change: The Magazine of Higher Learning, 45*(6), 34-39.

[30] Saul Schleimer, Daniel S. Wilkerson & Alex Aiken (2003, June). Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, 76-85. ACM Press.

[31] Michael Schubert, Damian Durruty, & David A. Joyner (2018). Measuring Learner Tone and Sentiment at Scale via Text Analysis of Forum Posts. In *Proceedings of the 8th Edition of the International Workshop on Personalization Approaches in Learning Environments (PALE)*. London, United Kingdom.

[32] Klaus Schwab (2017). *The fourth industrial revolution*. Crown Business.

[33] Dana Sheahen & David A. Joyner (2016). TAPS: A MOSS Extension for Detecting Software Plagiarism at Scale. In *Proceedings of the Third Annual ACM Conference on Learning at Scale*. Edinburgh, Scotland.

[34] Hoi K. Suen (2014). Peer assessment for massive open online courses (MOOCs). *The International Review of Research in Open and Distributed Learning, 15*(3).

[35] Leslie Waguespack, Jeffry Stephen Babb & David Yates (2018). Triangulating Coding Bootcamps in IS Education: Bootleg Education or Disruptive Innovation?. *Information Systems Education Journal, 16*(6), 48.