A Total Quality Management Tool for Experiential Engineering Education

Dr. Nicholas DiZinno, New York University

Nicholas DiZinno holds a B.S.M.E. from the University of Central Florida, an M.S.M.E from Polytechnic University and a Ph.D. from New York University. He has taught at the community college and senior college levels prior to his current appointment as an Industry Associate Professor at NYU. He is active in AIAA and holds a NYS Professional Engineers license.

Ms. Yona Jean-Pierre, New York University

Yona Jean-Pierre is the Director of Faculty Innovations for Teaching and Learning (FITL) at NYU Tandon School of Engineering. She received her Bachelor of Arts in Mathematics and Philosophy from the University of Rochester and her Master of Science in Mathematics from NYU Tandon School of Engineering. She manages the FITL Center, which encourages and promotes innovative teaching strategies and technologies for learning. The Center provides faculty with training on new classroom technologies, access to technologies and resources, opportunities for research and design, and collaboration on educational and technological projects. Ms. Jean-Pierre has taught Mathematics, Problem- Solving and Academic Success Seminars at Polytechnic University and Columbia University. In addition to her experience in academia, Ms. Jean-Pierre has practical experience in developing online technology and multimedia products having worked in corporate positions at Google Inc. and iVillage Inc.

A Total Quality Management Tool for Experiential Engineering Education

Abstract

The development and deployment of a web-based software tool, the Automated Grading Platform (AGP), will be presented. A collaborative effort between the Mechanical & Aerospace Engineering department and Faculty Innovations in Teaching & Learning (FITL) Center, the AGP allows for the real-time grading of assignments and provides immediate feedback to students in Mechanical Engineering courses. The platform is designed to work within the Total Quality Management (TQM) paradigm. With this system, only exemplary work is accepted. However, students are free to repeat an assignment an unlimited number of times until they produce work that meets this standard. Grade thresholds are established based on the volume of acceptable work submitted. This threshold is distributed to students at the beginning of the semester. The student can determine the grade they wish to achieve, and stop working when they have reached the desired grade threshold.

Applying a Total Quality Management (TQM) Tool in an educational setting will provide a quantum leap forward in providing both flexibility and support to our students. It draws on existing educational paradigms such as constructivism and the experiential theory of learning, yet offers a new and exciting approach to covering course material. Students get increased flexibility to tailor their workflow to achieve specific course objectives while the instructor is freed from routine calculations and allowed to focus on student support. Additionally, it breaks down barriers often encountered by students working remotely and on other campuses. It is expected that TQM and AGP will fundamentally change the way that Engineering is taught for the foreseeable future.

The paper will discuss the development of the Automated Grading Platform and its deployment and testing in an undergraduate Computer Aided Design (CAD) course. This course is typically taken by freshman in the Mechanical Engineering program. We will discuss the need for such an application, its development, and the results of our pilot study. Emphasis will be placed on the effect that the AGP has on student outcomes and important metrics. The role of the AGP within the broader context of student resources will also be considered. Lessons learned from our pilot study will be reviewed in preparation for a full implementation and evaluation of the AGP.

I. Introduction

The field of Engineering Graphics has evolved substantially over the last 50 years. From the days of the drafting board, we have seen the emergence of primitive Computer Aided Design (CAD) tools give way to 2-D drawing programs and eventually to 3-D solid modeling packages. Parametric solid modeling, sometimes known as constraint-based CAD, allows users to capture design intent and to take full advantage of the desktop computer as a design tool. As a result, required courses in Computer Aided Design have become ubiquitous in undergraduate Mechanical Engineering programs. Typically, students are introduced to the basics of spatial visualization, the theory of various projection techniques and the preparation of engineering drawings, all the while mastering the mechanics of using a particular software package. As with many courses in today's credit-starved curricula, teaching a CAD course presents a unique set of challenges. Perhaps foremost is the varying starting abilities of the students [1]. Some may be distracted by computer graphics, trying to make parts that "look good", but fail to incorporate robust modeling strategies [2]. As constraint-based

CAD has shifted the emphasis from documentation to correct part geometry and design intent [3], it is critical that students move beyond simple procedural knowledge and develop proficiency in higher-level strategic thinking [2,4]. How then are educators to take students from varying backgrounds, lead them past the glitter of rendered images, take them through the procedural manipulations of software and arrive at the plane of strategic thinking? The key is experiential learning. As Aristotle said, "*For the things we have to learn before we can do them, we learn by doing them*" [5]. However, experiential learning is more than just learning by doing; it requires reflecting on what has been done [6,7]. To achieve this, students must not only practice a substantial quantity of assignments, they must receive feedback and then reflect on the experience. The value of feedback is well accepted. The authors of [8] state that prompt feedback is essential for students to improve their models and make solid modeling more of an engineering design task and less of an art project; while both [2] and [4] contend that without feedback to remedy their models, students develop and perpetuate poor modeling strategies. In spite of the obvious value, traditionally there is a significant time lag between the introduction of a concept and the receipt of feedback by the student. In some instances, weeks may elapse before a misunderstanding is uncovered [9]. Often, by this time, it is too late to modify teaching and learning activities [9] to remedy the situation. The culprit here is the methodology used to provide the feedback. The desired feedback comes as a result of some form of assessment. That assessment is typically a visual inspection or the electronic measuring of geometry in a computer file [10]. This type of assessment is both labor intensive and time-consuming [2,3,10,11]. And it is antiquated. With the evolution of CAD technology, examining print-outs of solid models or drawings is no longer sufficient to determine the correctness of geometry [3]. As CAD technology has evolved, so must the assessment of CAD coursework evolve. Fortunately, automatic grading can be used successfully to provide consistent feedback on part models and reduce or eliminate grading time [2]. In addition, it provides other benefits such as accuracy, repeatability and objectivity [2,8,11]. While the need for an automated system is well established, the effort to create such a system is in its infancy. The authors of [8] have developed a computer program to automatically grade Siemens NX files. It must be launched from within the NX software, and compares the student file to a reference file. Mass property data is evaluated, along with analyzing sketches and features. Limited data is offered to show that the automated grades compare well with manual grading. [11] describes the logic behind an automatic grading tool for architectural applications, however, their tool is still under development. [12] describes the logic behind an automatic grading tool for Solidworks using macros and visual basic. However, no data is available as their tool is still under development. [13] describes an evaluation algorithm that compares a part to a reference, but it is a manual process that has not yet been computerized. [1] uses personal interviews to assess student learning. [2] describes an automatic grading tool within a Learning Management System (LMS). The student file is compared to a reference file. Limited results were shown. No direct validation of the tool by comparison with other grading methods is presented. While all of these solutions offer some promise, there is still a long way to go toward fulfilling the need for immediate, specific feedback that allows students to reflect on what they have done.

The Automated Grading Platform (AGP) described in this paper intends to contribute toward fulfilling this need. It is a web-based platform for the real-time grading of assignments that provides immediate feedback to the student. The AGP employs a total quality strategy, which will improve student proficiency in fundamental skills, increase student engagement, and break down barriers for students that work remotely. It offers increased flexibility for students to tailor their workflow to achieve specific course objectives. In section II of this paper, we describe the development and functionality of the AGP tool, while section III details the system deployment as well as our internal testing and the results of the soft-pilot study of the system. In section IV, we conclude with a summary of our findings to date as well as our plans for the future.

II. Development of the AGP

Terminology

The following terms are core to the CAD industry: Computer Aided Design (CAD) is a framework where a computer is used as a tool for designing and drafting of mechanical parts. A Solid Model (also referred to as a part) is a 3-D representation of an object within the computer. It has mass, occupies volume and has all the properties of a real object (e.g. center of gravity). A Drawing (also referred to as a two-dimensional drawing or a blueprint) is a flat paper representation of a solid model. It is typically an arrangement of views and contains dimensions and specifications necessary for the manufacture of a part. Solidworks is a particular piece of CAD software used to create solid models and generate two-dimensional drawings from them. It is the current state-of-the-art in CAD software. In the context of the TQM paradigm, work is considered *exemplary* if it meets a standard of excellence as defined by the course instructor. For example, an instructor may define work on a particular assignment as exemplary if the dimensions of a student's part match those of a reference part to within a 95% match.

Background

At the author's University, the CAD course is required of all Mechanical Engineering majors but is also frequently taken as an elective by students in other majors. The course is offered every semester, and approximately 150 students take the course each year. During the course, students use the commercially available *Solidworks* software to create solid models with a computer. Certain assignments will also require the student to generate two-dimensional drawings from their solid models, and also assemble multiple models into an assembly.

Concept and Analysis

The Automated Grading Platform (AGP) is a web-based software system that automates the entire grading workflow. It frees the instructor from routine calculations, allowing an expanded emphasis on student support, and it gives the student immediate feedback about the quality of their work. Students upload their CAD files to a webpage. While the student waits, the system will perform routine administrative checks, such as file name formatting and the type of units used. The system then automatically grades the part by comparing it to an (instructor provided) reference file. This happens in a matter of minutes. Feedback is then provided to the student via on-screen comments. In a future version of the AGP, that information will also be sent to the student via email. The total time from file upload to receiving of comments is on the order of minutes. If the submission does not meet various standards, the student is offered help via traditional help files and links to short video clips to clarify specific concepts. In addition to a general critique of their work, students also see their grade. The AGP system maintains a database of the feedback related to a particular part. This allows the student to login at any time and review their work. The instructor can download grade results as detailed data for each student and also as class averaged statistics. Thus, by querying on a certain error, the instructor can find the distribution of how many students made that error. This allows for a more customized classroom delivery, targeting lessons to the most common errors. The system is designed to dovetail closely with the concept of Total Quality Management (TQM) [14,15].

The Automated Grading Platform is developed in coordination with the Faculty Innovation in Teaching and Learning Center at the author's university. The overall system specification is defined and written by the Mechanical Engineering instructor and the FITL Center assembled a product development group consisting of the course instructor, director of the Center as the project manager, an interactive designer, two graduate student programmers from the Department of Computer Science & Engineering and two graduate assistants from the Mechanical Engineering Department. Undergraduate students who have already taken a CAD course will be asked to perform beta testing of the software at various stages during its development. This structure of the development team is important as we need strong coders and web-development resources paired with the content owners. An additional goal is to provide our graduate students with exposure to a project of real-world complexity and scope.

The system draws on the instructor's extensive experience over the past ten years teaching Computer Aided Design. He searched the market for a product with functionalities to assist him in grading student assignments, however he did not find an existing product that could help him. This system utilizes conventional education models and implements it into an automated platform for successful teaching and learning. And while the idea of automatic grading tools is common in computer programming (see, e.g. [23] and the references therein), it is the combination of the automation technology with the pedagogical paradigm of TQM that makes the AGP unique. The AGP will have a strong impact on the way that Computer Aided Design is taught and assessed in the future.

The Total Quality Paradigm

The AGP integrates into classroom use within the paradigm of Total Quality Management (TQM). While there are many facets to TQM, the key concept that will be applied here is that of *continuous improvement* [14,15]. A student's work is only accepted for a grade if it is exemplary. If there are any flaws, the student will receive feedback and have the opportunity to revise their work. This iterative process of *receive feedback – revise* continues until the work has reached the standard set by the instructor as exemplary. At that point, the assignment is marked as complete. The minimum number of completed assignments required to achieve a given grade will be determined by the instructor. Assignments that are flawed will not count towards the required tally. If a student submits a flawed assignment, he or she has the option of re-doing the same assignment or choosing another one. This gives the student the flexibility to choose assignments that they feel comfortable with. They get the immediate feedback provided by the Automated Grading Platform, and they can choose which grade they are aiming for based on the amount of time that they wish to invest in the course. In this setting, only work that has met the instructor's definition of exemplary is acceptable. In this paradigm, inferior work is not punished. Rather, quality work is rewarded. The instructor sets the thresholds for each grade in a way that the student is guaranteed to achieve a given level of proficiency with the course goals. The student gets the advantage of choosing their own assignments and benefiting through reinforcement via immediate feedback. There is strong evidence to suggest that the TQM paradigm has value in an educational setting [16,17]. In fact, TQM has been shown to influence student attitudes. As [18] points out, when students begin to expect to do quality work, they begin to be less sloppy and pay more attention to details [18]. As an example of the use of TQM, consider an assignment that consists of creating a solid model of one part. At the beginning of the term, the instructor will assign 250 parts, grouped equally into ten bins, twenty-five parts per bin. Each bin represents a fundamental concept and skill that the student must gain proficiency in. In order to earn a particular grade, the student must successfully complete parts according to the schedule shown in table 1 below:

Desired Grade	**Required Number of Accepted Parts from Each Bin**
A	12 – 14
B	9 – 11
C	6 – 8
D	3 – 5
F	2 or less

Table 1: Part Thresholds per Grade

Deadlines are used at the discretion of the instructor. Within the assigned time frame for a particular problem set, students have an unlimited number of attempts to revise their work, getting immediate feedback at every step via the AGP. The instructor will post each student's tally via the Learning Management System (LMS) on a weekly basis. This, combined with the feedback from the AGP, allows the student to know exactly where they stand at all times. The crux of this system is that, *no matter what grade a student achieves, all of the work that they have submitted is of the same high-quality standard.* In this paradigm, we are abandoning the model that once a student submits a piece of work, it is "out-of-sight, out of mind" and that their grade is at the mercy of the instructor. Students will continually work and re-work much greater quantities of problems than in the traditional model. To fully implement the experiential learning cycle, at certain points in the revision process the student will be required to perform a reflective observation. Here they identify any inconsistencies between their current state of knowledge and that required for a successful completion of the assignment. These reflections are used as the starting point for classroom discussions that specifically target student weaknesses. In spite of the scant adoption of TQM in education, it is our contention that *a total quality teaching strategy with immediate feedback will improve student proficiency in fundamental skills, increase student engagement, and break down barriers for students that work remotely*.

Similarity of the TQM Paradigm to Other Educational Models

As an educational framework, the Total Quality Management paradigm can be compared and contrasted with other models, such as Learning for Mastery (LFM), pioneered by Benjamin Bloom [20, 21] and the Keller Plan, a Personalized System of Instruction (PSI) [22]. In these models, as in other initiatives in competency-based education, a student moves through the curriculum essentially at their own pace. Once they have demonstrated *mastery* of a topic, say by performance on a test or other assessment device, they are free to move on to the next topic. Students who do not demonstrate this mastery are given reinforcement, before being assessed again. The TQM method shares the idea of reinforcement, however it retains the idea of a fixed timeline. Within the timeline defined by the instructor, students are encouraged to develop competency through *volume*. The grade a student receives is directly correlated to the amount of quality work the student has completed. Work that does not meet the standard defined as exemplary is not counted, yet the more exemplary work is done, the higher the grade that is achieved. If a student feels, after several tries, that they cannot complete a particular part, they are free to choose another. In the TQM model, there is nothing to prevent a student from moving on to new material. While the reinforcement and revision aspects are shared between TQM, LFM and PPI, the power of the TQM paradigm stems from its' core principle of continuous improvement. Providing *immediate* feedback allows students the opportunity for reflection, which aids in the building of strategic thinking skills. So while we may emphasize the volume of work done, it is important to recognize the role of targeted and immediate feedback in the development of competency that is provided by TQM.

Design Approach

A five step process was utilized to conceptualize and design the ideal user-interface to ease navigation and confirm proper functionalities for AGP. These initial analysis and evaluations were critical in creating a strong foundation with which to move forward. Figure 1 below details each step and the team members involved in the task.

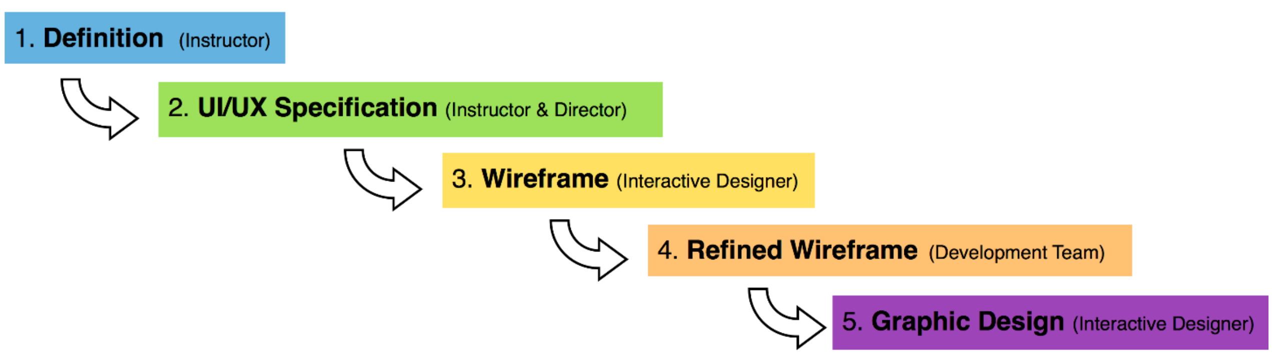

Figure 1: Workflow Steps

The accuracy of the Definition phase was significant and led the development. It was critical to ensure the full outline of each idea and to delineate all the individual components of the application in clearly segmented ways. This process took several one-on-one analysis meetings and substantial time to execute. The UI/UX Specification stage was preliminary for converting the textual information into a visual representation of the application to prepare us for a kick-off discussion with the interactive designer. After the meeting, our designer spent time reviewing and revising the Specification to construct the Wireframe for the application. During this phase, the designer was very deliberate and detailed in creating a clear, concise and user-friendly navigation flow for user ease in locating significant links, recognizing action functions, and visualizing the function relationships in the UI for increased usability. Reviewing and refining is part of the product development process and the entire development team was involved with this task. Additional functionalities were added, tweaks were discussed, approved and applied. This process took us to the final phase, graphic design for decisions on aesthetic, look and feel and logo design before beginning front-end development. A feel for the user-interface can be had by examination of Figure 2 below, which shows the log-in screen.

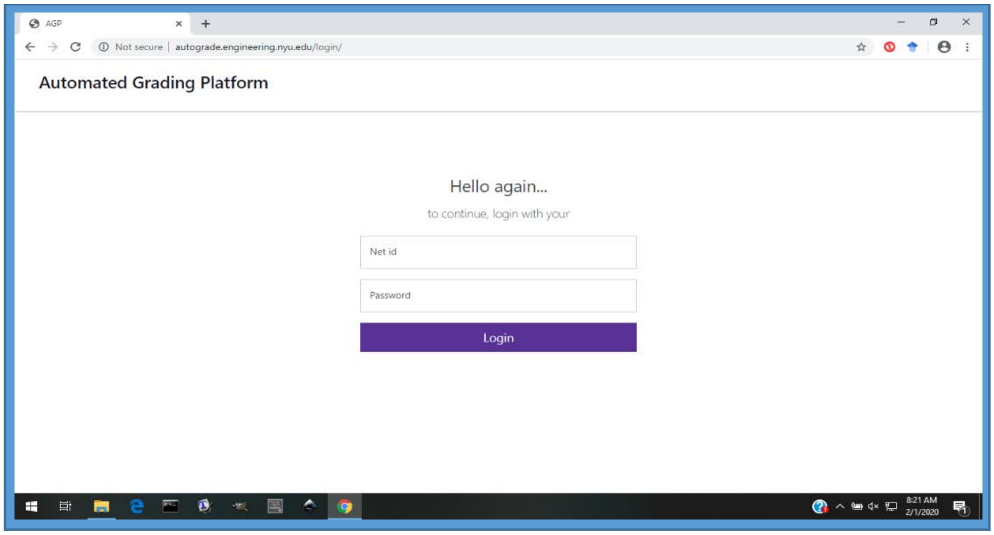

Figure 2: AGP User Interface

AGP uses basic HTML and Javascript coding for front-end page structuring and design. The main algorithm is coded in the Python programming language (Python 3.7), and uses Django as the web framework and MySQL for back-end manipulation and logical processing. A combination of all these programming tools allows for the development of different functionalities for executing AGP procedures such as uploading files, saving files, processing and displaying content. The website is hosted on university server space with a Windows 2019 server running IIS as the web server.

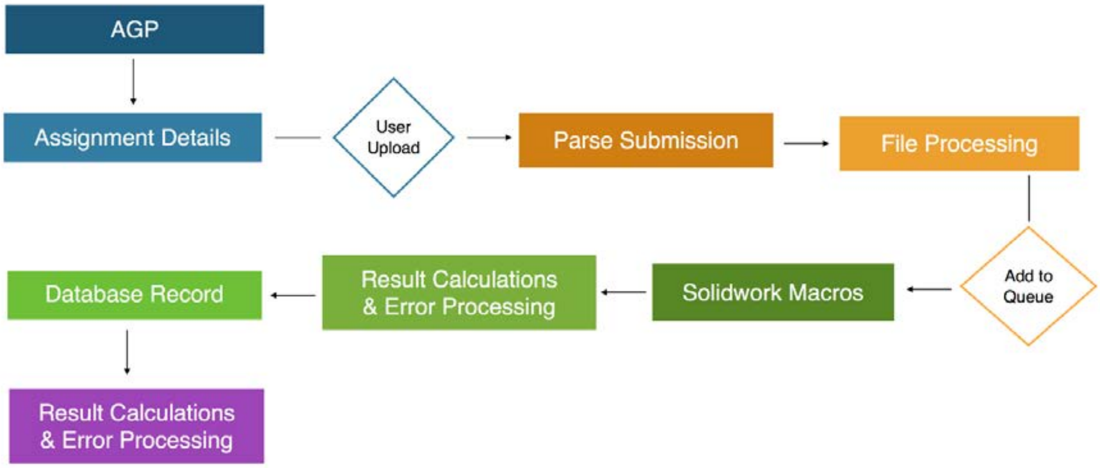

Figure 3: AGP Grading Algorithm

The most intricate component of AGP is the processing of student submitted files and the providing of immediate grading feedback to students. The combined concepts and features, which validate this platform is collecting assignments from students, auto-grading these submissions, and giving feedback to students to allow for modification and re-submission until the student gets the grade that he/she desires. The AGP grading algorithm, which compares the mass properties of a student-provided file to those of an instructor provided reference file, originates from the Solidworks Macros that have been used by the instructor for numerous years to perform the retrieval, manipulation and storing of data from submitted and instructor reference files. Figure 3 shows a flowchart that graphically depicts the how grading algorithm generically works.

According to the AGP system content algorithm, the instructor can enter a number of assignments with a set number of parts for students to select from, complete, submit and re-submit until he/she is satisfied with their grade. Hence, the student can continuously work on assignment parts to increase the number of exemplary parts submitted to earn a maximum grade in the course. Table 2 below shows an example of four parts submitted by a student for an assignment. (Note that in this particular example, the instructor has set a threshold of a 90% match to the reference part as the definition of exemplary).

Student	Assignment	Parts	% Match	Grade
student1	Assignment1	Part1	94.789931	100
student1	Assignment1	Part3	72.834293	0
student1	Assignment1	Part6	0	0
student1	Assignment1	Part8	93.3158424	100

Table 2: Representative Part Grades

The system grades the assignment by comparing the student's solution to reference files saved in the system by the instructor who set up the assignment.

Assemblies are evaluated in much the same way as individual parts. The biggest difference between the two is in the uploading procedure. All files in an assembly are required to be uploaded as a single compressed file (e.g. a zip file), which can easily be achieved in Windows OS applications by using the Pack-and-Go utility. The AGP scans the contents of the compressed file. The user is then required to manually map the filename of each uploaded file to that of a linked part in the assembly. The parts are subsequently evaluated as individual entities as described earlier, while the assembly is then evaluated in a similar fashion, i.e. by comparison to a reference file.

System Queuing

Traffic into the AGP server must be closely monitored. The AGP maintains a folder (Queue) which is the default destination for any incoming file traffic. Files are timestamped as they arrive. The queue folder maintains an ordered list of files based on their submission timestamp in a First-In-First-Out

(FIFO) order. The processing algorithm operates serially, there is only one thread (a task on the server) that is allowed to operate at any given time. As an example, consider three files being uploaded from three different users at approximately the same time. Due to subtle differences in user execution, each file will have a different timestamp. The queue will save those files in the order of their submission time (this is independent of the Solidworks creation date and time). Each of the three submissions will try to invoke the processing algorithm to start the grading process. Only the very first of those requests will be processed and the others are just simply ignored (thread processing). Once the algorithm is done processing a file (success or error, regardless), it fetches the next one from the queue folder and moves the current one to a proper destination folder post-processing. When the queue encounters an empty list, it sends a request to quit Solidworks after a period of time. The process repeats for every file submission.

Development

The programming development of AGP started in October 2019. One lead developer was assigned the task of reading all generated documentation with the assistance of two mechanical engineering graduate assistants (ME GAs) for the course being piloted. They wrote an analysis of how to proceed with development. The developer works 20 hours a week with 1-3 hours spent on meeting with the Center Director, the instructor and the ME GA. Initially, the developer concentrated on understanding all the documentation, asking questions, compiling new information and generating new files to organize the new found data. Given that we are funded by a school curriculum grant which requires the completion of this project by the end of May, the development pace for this project is very aggressive with plans to run 1-2 pilots by end of Spring 2020 semester. The instant focus for the developer was fully understanding the macros, and deciding on the appropriate server for running all needed tools to process the proposed algorithm. A significant amount of time was spent on testing the performance of the defined macros, Solidworks software, Windows Server environment and Python built-in functions and methods. Error analysis will continue to be a routine efficiency check for this project. Challenges are part of product development. The biggest challenge is needing more work hours to tackle a project of this magnitude, given we have part-time graduate student employees. How we handle this daily challenge will determine the success of the project. The top priority for us throughout the development process is AGP's performance in the web-environment and how this will scale as the number of simultaneous log-ins increase.

III. Deployment & Testing of the AGP

In the Fall, 2019 semester the TQM paradigm was deployed independently of the AGP in an undergraduate CAD class of 39 students. A working prototype of the AGP system was launched in Spring, 2020. Internal testing was conducted by the development staff, and then a soft-pilot was conducted using 20 student volunteers.

Attitudes Toward TQM

As a baseline, it was desired to ascertain the efficacy of and student attitudes toward the TQM paradigm independent of the AGP. To this end, only the TQM approach was implemented in the Fall 2019 semester. Grading was done manually using a team of graduate students. There were 4 assignments given throughout the semester, each worth 30 points. The assignment consisted of approximately 70 parts varying between one and five points in value. The window for each assignment was 30 days long. Within that time, students were free to pick and choose any combination of parts to achieve their desired total point value and had unlimited re-submissions. Grading was done on a weekly basis. At the conclusion of the semester, a survey of nine questions was administered to the class. And while the Likert scale in this initial survey was slightly skewed, future surveys will have a more balanced focus. Results for overall satisfaction with TQM are shown in Figure 4 below, while details of the remainder of the questions are presented in Appendix 1.

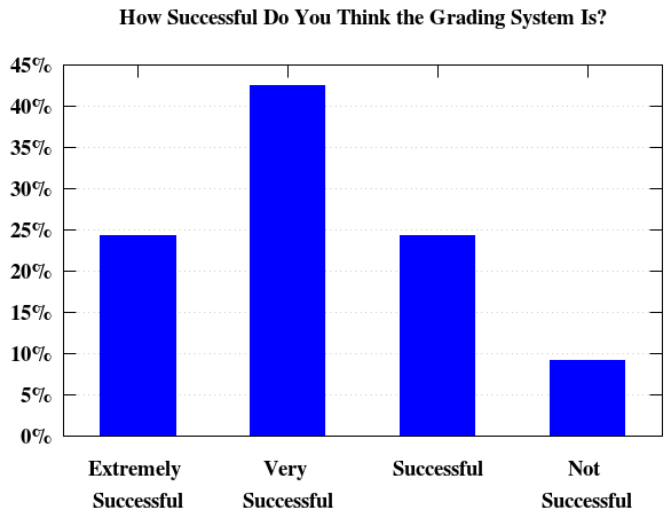

Figure 4: Representative Results of TQM Survey

Overall, student satisfaction with the total quality framework was quite high, with approximately 91% of students indicating that the system was successful in some way. The majority of results confirmed our expectations. Students generally responded well to the flexibility aspects of TQM and the ability to resubmit parts that were faulty. Approximately 49% of students felt that work being either completely acceptable or completely unacceptable had extreme value. This would suggest that these students appreciate the lack of ambiguity offered by the TQM setting. An unexpected result involved the question related to how many parts a student would submit if they received feedback faster. While 42% responded that they would submit more parts, 42% also replied that they would submit the same amount of work and 16% replied that they would submit less work. This result suggests that the TQM paradigm can support all work flows. Ostensibly, the 16% who would submit fewer parts would be working more efficiently to achieve their course goals. The power of the re-submission concept is shown by focusing on the two parts with the highest resubmission rates throughout the term. See the data in tables 3 and 4 below.

	Part A	**Part B**
Total Number of Students	102	102
Students who Resubmitted	43	31
	42%	30%

Table 3: Resubmission Rates

The highest resubmission rates (without loss of generality, the parts are identified as Part A and Part

B) were 42% and 30%. More insightful, perhaps, is the distribution in the number of times a part was

Number of Resubmitions	Part A		Part B	
1	27	63%	17	55%
2	10	23%	3	10%
3	3	7%	1	3%
> 3	3	7%	10	32%

Table 4: Resubmission Distribution

resubmitted. While the highest frequency was for a single resubmission, resubmissions of two, three and even more than three were significant. Of course, with the use of the AGP, more than three submissions would raise a flag for the instructor to reach out to the student to discover the nature of the difficulty and to devise a new plan of attack.

Internal Testing

As a testing exercise limited to the AGP developers, a random set of parts with a variety of intentional flaws were graded using the AGP and also independently via manual hand-calculation of the AGP algorithm. This was done in order to demonstrate that the software platform is bug-free and that the grades produced by the AGP are as expected. The results are shown in Figure 5 below. The agreement is excellent.

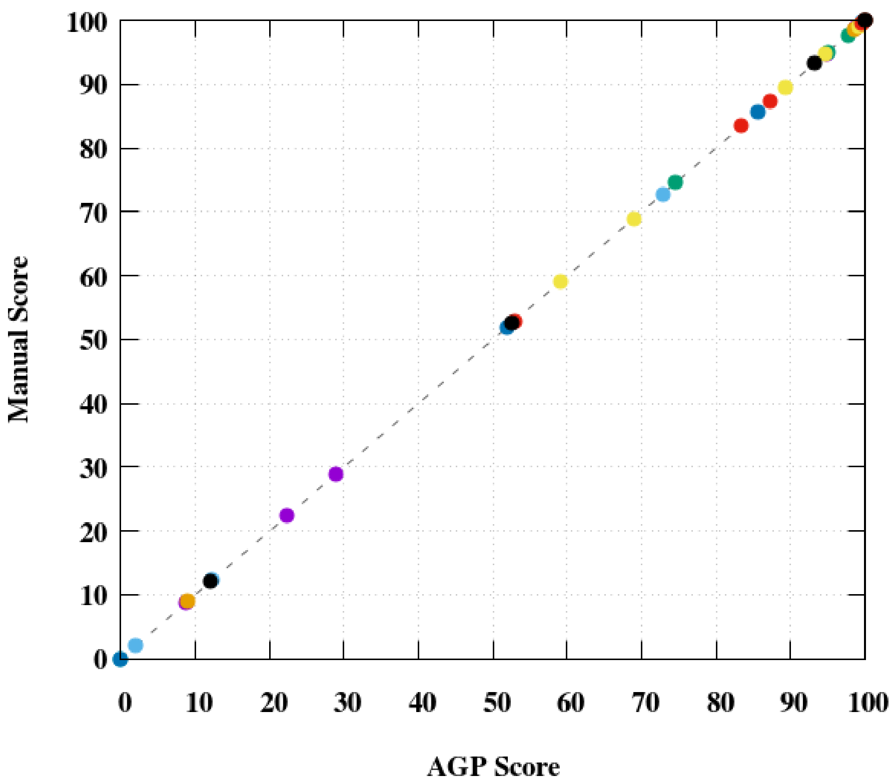

Figure 5: AGP Scoring vs. Manual Scoring

Please note that for the final deployed version of the AGP, the displayed results will be Accepted/Not

Accepted, via the TQM paradigm. However, for the initial testing done at this stage, actual part scores as determined by the AGP grading algorithm are used to ascertain system performance.

The Soft Pilot

Soft-pilot testing was conducted with 20 student volunteers, each of whom created four parts. A large pool of parts was offered, and participants chose which parts to model. All parts were of a typical complexity found in an undergraduate CAD course. A representative sample is shown in Figure 6 below:

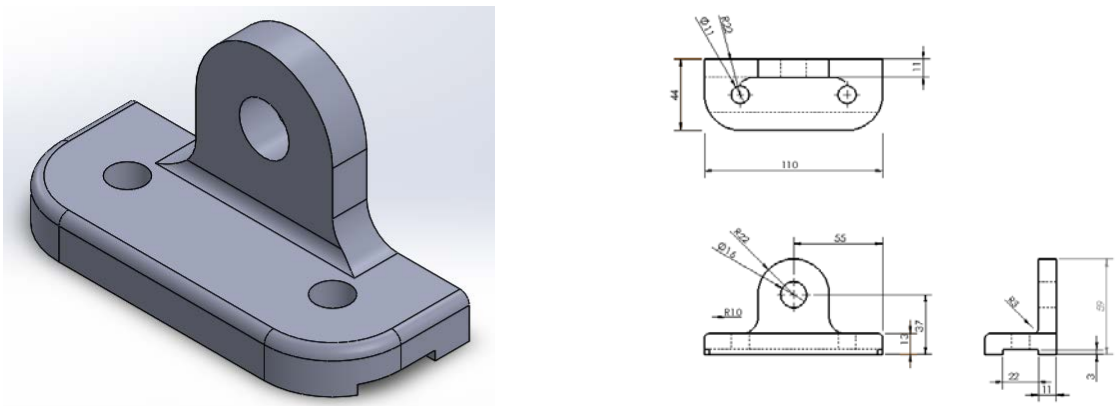

Figure 6: Typical Part Assignment

Students were encouraged to create some parts as conscientiously as possible, while creating others that intentionally introduced flaws (in a completely random manner). Flaws were to fall into one of the following categories, shown in table 5:

Incorrect units
Part uniformly scaled up
Part uniformly scaled down
Omit a required feature
Add an extraneous feature
Intentionally modify a single dimension (oversize)
Intentionally modify a single dimension (undersize)
Intentionally model the wrong part

Table 5: Fault Categories

As a representative sample of a typical students' results, refer to Figure 7 below. Due to the random nature of the flaws, scores are unevenly distributed, as expected.

Results of various parts are shown in Figure 8 below. Each of the different series corresponds to a different student. However, due to the flexibility offered by the TQM system, not every student did the same parts. Hence, the students are not identified. The significant result here is that the system can handle parts from across the entire spectrum of scores.

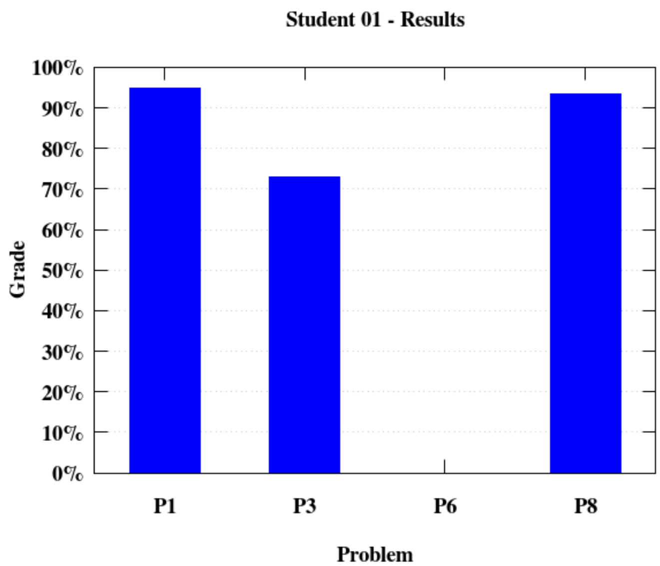

Figure 7: Typical Student Response

System Timing

No specific timing data were recorded during the soft-pilot. However, anecdotal evidence can be deduced post hoc from certain observations. The soft-pilot consisted of 20 different users uploading four files each, more or less sequentially. The overall processing time for the eighty files (41 minutes) would suggest an average processing speed of approximately 30 seconds per file. Examination of the posting of individual results, however, would suggest an approximate average processing speed of 20 seconds per file. This is raw processing time for the AGP. Overhead for data input/output and screen interface is to be expected, so this is not what the user would experience while waiting in front of the computer. It does, however, lend credence to our claim that the AGP will process files on the order of minutes, providing the user with while-you-wait feedback.

Plans for the Future

Logic has been determined to give more qualitative feedback based on metrics calculated from mass properties. These will be tested in the next pilot study, which we also anticipate will involve more student users. Another of our major goals over the next several months of the project is to include the ability to track actual features of the solid model, in addition to its mass properties. This, when combined with the logic mentioned above, will allow for a substantial amount of qualitative feedback to be provided to the user. Since our interpretation of the TQM focuses on the re-submission of parts until they are exemplary (as defined by the instructor), we intend to guide the user toward this goal with targeted feedback from the current submission. While current efforts are focused on testing and refining the AGP's ability to process both parts and assemblies, in the future we intend to expand the capabilities of the AGP to also evaluate 2-D drawings. After extensive testing to ensure a robust and secure platform, we will invite interested CAD users to log into the AGP site and take advantage of its unique capabilities.

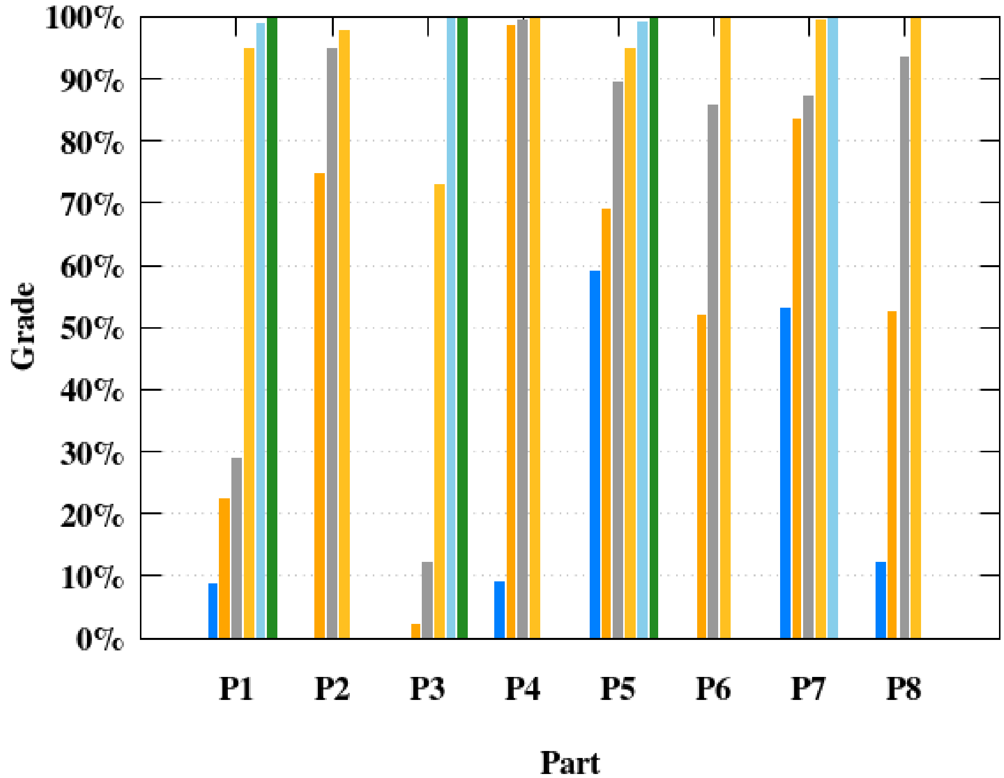

Figure 8: Grades per Student by Part

IV. Conclusions

The conception, development and initial testing of an automated, web-based platform for the grading of solid models has been presented. It grades parts and assemblies in real-time, while the student waits. Coupled with the Total Quality Management (TQM) paradigm, it is a powerful tool for students to develop proficiency in fundamental CAD skills in an efficient manner that is tailored to their specific workflow. The true value of the system can best be viewed in a broader context. The AGP allows for deeper *student engagement* in the course and encourages each student to *continuously improve* their work. The downtime typically associated with grading is effectively eliminated, and the immediate feedback essentially provides each student with the instructor's attention at any time, day or night. Students who are working remotely have the same advantages of those who are physically on campus. Test results have shown that the system is robust and accurate and can handle typical web traffic with aplomb. Further refinement will continue in the future.

Acknowledgements

The authors wish to thank the reviewers for providing valuable comments that improved the paper. Also, sincere thanks to our lead programmer, Abhishek Sharma, for fruitful discussions and for effort above and beyond the call of duty, and to Katerina Skiadis of FITL for assistance with the figures.

References

[1] G. Menary and T. Robinson, "Novel Approaches for Teaching and Assessing CAD," 2011.

[2] H. Ault and A. Fraser, "A comparison of manual vs. online grading for solid models," in *Proceedings of the ASEE 120th Annual Conference and Exposition, Atlanta, GA, USA, June 23-26, 2013.*

[3] T. Branoff, E. Wiebe and N. Hartman, "Integrating Constraint-Based CAD into an Introductory Engineering Graphics Course: Activities and Grading Strategies," in *Proceedings of the ASEE Annual Conference and Exposition, Nashville, TN, USA, June 22-25, 2003.*

[4] I. Chester, "Teaching for CAD Expertise," *Int'l Journal of Technology Design Education*, Vol. 17, pp. 23-35, 2007.

[5] Nicomachean Ethics, by Aristotle. http://classics.mit.edu/Aristotle/nicomachaen.2.ii.html, Accessed 1 February 2020.

[6] D. Kolb, *Experiential learning: Experience as the source of learning and development* (Vol. 1). Englewood Cliffs, NJ: Prentice-Hall, 1984.

[7] D. Kolb and R. Fry, *Toward an applied theory of experiential learning*. MIT Alfred P. Sloan School of Management, 1974.

[8] S. Kirstukas, "Development and evaluation of a computer program to assess student CAD models," in *Proceedings of the ASEE 123rd Annual Conference and Exposition, New Orleans, LA, USA, June 26-28, 2016.*

[9] D. Elrod and M. Stewart, "Assessing Student Work in Engineering Graphics and Visualization Course," in *Proceedings of the ASEE Annual Conference and Exposition, Salt Lake City, UT, USA, June 20-23, 2004.*

[10] E. Wiebe, T. Branoff and N. Hartman, "Dynamic Modeling with Constraint-Based CAD in Introductory Engineering Graphics," in *Proceedings of the ASEE Annual Conference and Exposition, Nashville, TN, USA, June 22-25, 2003.*

[11] A. Sanna, F. Lamberti, G. Paravati and C. Demartini, "Automatic assessment of 3D modeling exams," *IEEE Transactions on Learning Technologies*, Vol. 5, No. 1, Jan-Mar. 2012.

[12] D. Baxter and M. Guerci, "Automating an introductory computer aided design course to improve student evaluation," in *Proceedings of the 2003 ASEE Annual Conference and Exposition, Nashville, TN, USA, June 22-25, 2003*.

[13] S. Kirtukas, "A Preliminary Scheme for Automated Grading and Instantaneous Feedback of 3D Solid Models," in *Proceedings of the ASEE Engineering Graphics Division 68th Midyear Conference, Worcester, MA, USA, October 20-22, 2013.*

[14] M. Jaraiedi and D. Ritz, “Total quality management applied to engineering education,” *Quality Assurance in Education*, Vol. 2, Issue 1, pp. 32-40, 1994.

[15] S. Waks and M. Frank, Application of total quality management approach principles and the ISO 9000 standards in engineering education,” *European Journal of Engineering Education*, Vol. 24, Issue 3, pp. 249-258, 1999.

[16] P. Sakthivel, “Top management commitment and overall engineering education excellence,” The TQM Magazine, Vol. 19, No. 3, 2007.

[17] A. Rugarcia, R. Felder, D. Woods, and J. Stice, “The Future of Engineering Education I. A Vision for a New Century,” *Chemical Engineering*. *Education*, Vol. 34, No. 1, pp. 16–25, 2000.

[18] E. Glover, “Learning physics with total quality,” *The Physics Teacher*, Vol. 35, pp. 210-211, Apr. 1997.

[19] P. Connally, “CAD Software Industry Trends and Directions,” Engineering Design Graphics Journal, Vol. 63, pp. 26-33, 1999.

[20] B. Bloom, “Learning for Mastery,” *UCLA – CSEIP – Evaluation Comment*, Vol. 1, No. 2, May 1968.

[21] B. Bloom, “Human Characteristics and School Learning,” McGraw-Hill, 1976.

[22] F. Keller, “Good-Bye Teacher,” *Journal of Applied Behavioral Analysis*, Vol. 1, No. 1, pp. 79-89, 1968.

[23] F. Restrepo-Calle, J.J. Ramirez and F. Gonzalez, “Continuous Assessment in a Computer Programming Course Supported by a Software Tool,” *Computer Applications in Engineering Education*, Vol. 27, Issue 1, pp. 80-89, January 2019.

Appendix 1: Details from TQM Survey