

Development and Evaluation of a Computer Program to Assess Student CAD Models

Dr. Steven Joseph Kirstukas, Central Connecticut State University

Steve Kirstukas is an Associate Professor at CCSU, where he teaches courses in solid modeling, MATLAB programming, and engineering mechanics. He is exploring the use of computer-aided assessment of CAD files to give consistent, accurate, and quick feedback to students. He has degrees in civil and mechanical engineering, with a Ph.D. from the University of Minnesota. Steve has worked in industry as a civil engineer, software developer, biomechanics researcher, and mechanical design engineer.

Development and Evaluation of a Computer Program to Assess Student CAD Models

Abstract

This paper presents a computer program that can automatically assess the geometry and changeability of student solid models. The program is launched from within the CAD software package and a graphical user interface is presented so that the user has the opportunity to set values for twelve different deduction categories. The program compares one or more student model files to a single “gold standard” model. The program allows for the assessment to be thorough, repeatable, non-subjective, and very fast. It can process 20 or more models per minute and generates output very similar to what would've been generated by hand. The program has been used to re-grade exams from the Spring 2015 semester; when computer-graded scores were compared to instructor-graded scores, a strong correlation was observed between the two scoring systems.

Introduction

At Central Connecticut State University, all engineering and engineering technology students learn 3D solid modeling in a single course that covers part-making, formation of parts into assemblies, and creation of layout drawings. We approach part model creation as an opportunity to participate in open-ended engineering design: it's one of the few times in the engineering curriculum in which traditional mathematical problem-solving plays a limited role in engineering design. Various different strategies can create acceptable solid models. For example, something as simple as a washer can be modeled by 1) extruding two concentric circles, or 2) revolving a rectangle about an axis.

When the modeled part is complex and utilizes multiple constrained sketches, extrudes, revolves, holes, mirrored features, patterned features, etc., many of which may depend on one another, evaluating the quality of the model becomes a time-consuming task¹ and one that can be affected by fatigue and influenced by subjectivity. It is important that the model not just reproduce the required geometrical shape and size but that it be changeable in the future in a manner that honors the design intent². Because the creation of changeable models requires some planning², modelers are faced with the seemingly opposing concepts of producing a given geometry quickly and producing a high-quality changeable model³. High-quality models are easy to change and require little time to revise⁴. The development of quality solid models should be a principal goal from the start of instruction⁴. Kirstukas⁵ proposed that design intent can be inferred by careful inspection of the provided part drawing. If the drawing contains the necessary and sufficient information to produce a unique solid model, then the solid model should utilize all drawing dimensions, leave out no dimensions, and repeat no dimensions. Thus, the set of dimensions in the model can be compared to those in the original layout drawing to form part of a quantitative assessment of model quality.

Computer grading of model files has been a goal for a long time, in order to reduce the total time needed by the instructor to assess many student models, to reduce the time between model

submittal and feedback delivered to the student, and to quantitatively assess model quality. Baxter and Guerci⁶ described the existence of a computer program written to automate the grading of SolidWorks files. The program compared key data from the student file to that of the correct file. However, details on the grading algorithm and results were not presented. Ault and Fraser¹ developed a program to automate the grading of Creo files. The program compared information from the student file to that of the instructor file, such as model volume, the presence or absence of critical dimensions, and the total number of extrude, revolve, hole, and pattern features. The program did not attempt to evaluate model changeability.

Both of these efforts read solid model files and compare information in one model to that of another model. Because file formats are usually proprietary and typically change after major software releases, a vendor-supplied software library called an Application Programming Interface (API) provides tools to query the solid model.

In this paper, a new computer-graded scheme is described. The previously described algorithms are expanded to evaluate shape and size, orientation, modeling practices, and changeability. The computer program is written specifically to compare Siemens NX solid model files, although much of the algorithm is relevant to other CAD software packages that have well-developed APIs.

The File Comparison Program

The computer program presented here is written in the VB.NET language (a form of Visual Basic, an object-oriented computer language that is relatively easy to learn) and is compiled in Visual Studio to produce an executable file. The program uses the Siemens-provided NX Open API to extract information from NX model files. There are many sample NX Open based programs included with the NX CAD package, and the internet has a wealth of resources for learning more about the potential of user-built NX Open based programs. The websites www.nxjournaling.com and www.eng-tips.com provide additional sample programs, example code, and a forum from which clues, hints, or complete solutions to specific issues can be obtained.

The computer program compares one NX solid model file to another. When one model is the instructor's "perfect" model and the other model is the student's model, the output of the comparison can be used to assess the student model.

The file comparison program is launched from within NX and a graphical user interface (GUI) is presented. The graphical user interface (Figure 1) is divided into three main groups — Deductions, File Selector, and Options — for which the user may alter default inputs. Below these three groups, there is an area for which output text may be displayed. The text output box is stretchable and can be enlarged to fill the screen. In addition, there are two buttons, one for clearing the text output box and one for initiating the file comparison.

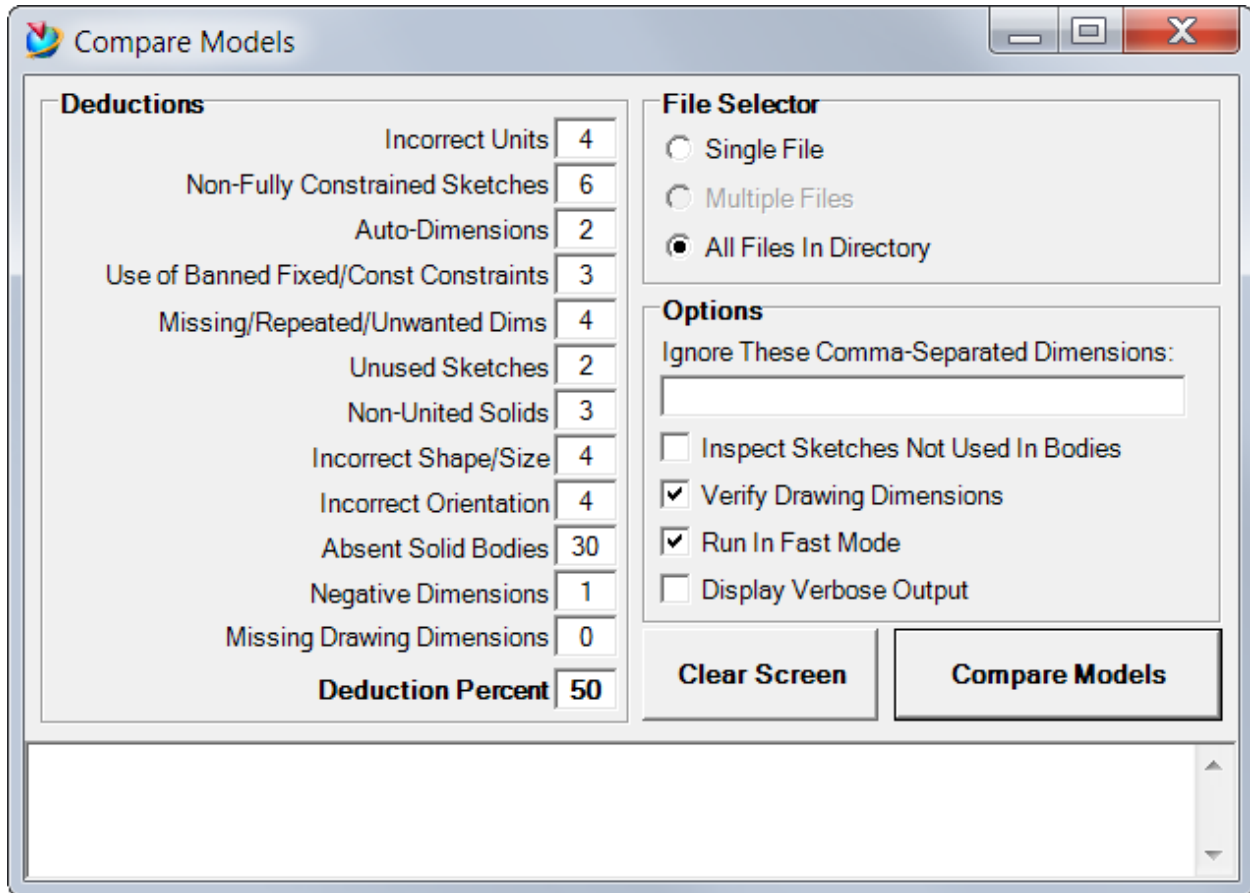


Figure 1. The user interface of the computer program.

A Deductions group lists 12 categories of issues that are common in beginning student models. For each category, point deductions may be individually set. These 12 deduction categories are:

Units: The first decision the modeler must make is often the unit convention. In our CAD software, length units are typically in either millimeters or inches. The correct choice is important, as it is not easy to convert a model from one system of units to the other.

Unconstrained Sketches: Every sketch is expected to be fully and properly constrained with no auto dimensions.

Auto Dimensions: A non-fully constrained sketch will generally contain one or more auto dimensions. Auto dimensions are not locked down and indicate that the sketch can be deformed or broken by dragging a vertex. This deduction applies an additional penalty on non-constrained sketches.

Banned Constraints: Students are encouraged to make generous use of the geometric constraints of horizontal, vertical, coincident, concentric, equal length, equal radius, tangent, point-on-curve, midpoint, and a few others. These all lead to changeable models that incorporate desired design intent. However, some beginning modelers, in an effort to eliminate auto dimensions, set certain points as “fixed”, or line segments at “constant length” or “constant angle”. These three

constraints make the model unchangeable and are not to be used. Any sketch containing one or more banned constraints will additionally be classified as a non-constrained sketch.

Missing/Repeated/Unwanted Dimensions: The modeler's task is to transfer the design intent as expressed in the layout drawing into the model. If the layout drawing contains the minimum information to fully understand the part, there will be a one-to-one correlation between dimensions in the drawing and dimensions in the model. If the drawing contains extra dimensional data, then the burden is on the modeler to decide which dimensions are to be left behind, or set as reference dimensions.

Unused Sketches: Beginning modelers sometimes create extraneous sketches that are not used in any other features. These may contain some model dimensions and may be partially or fully constrained. If these sketches were never used in features such as extrudes, revolves, or holes, they don't contribute to the solid model and the modeler should have deleted these before submitting the model.

Non-United Bodies: In our CAD software, if a part is composed of non-united bodies then operations such as placing holes through adjacent bodies, or placing edge blends on an edge shared by the bodies, do not go as expected. Adjacent bodies should be united into a single body.

Shape and Size: The student model in its submitted unaltered form (static configuration) should have the shape and size specified on the layout drawings.

Orientation: Models must be constructed with the goal of being used to create a layout drawing. The placement of a model upon the world coordinate system should result in the desired isometric orientation.

Absent Solids: Outside of the first couple of weeks, students construct solid models. If a solid is expected but not produced, then models would lose these points, as well as deductions based on shape and size, and orientation.

Negative Dimensions: The typical extrusion is created by sweeping a selected region in a direction perpendicular to the region. An equivalent solid could be produced by extruding material in the opposite direction by a negative distance. This is confusing and can be avoided. In our CAD software, there is a "reverse direction" button so that there is never a need to use negative numbers.

Missing Drawing Dimensions: While the program is primarily developed for evaluating sketches and solid models, it can assist with the tedious evaluation of rather complex layout drawings. It can verify whether the critical dimensions of the original drawing have been incorporated into the final layout drawing.

Deduction Percent: Early in the semester when models depend on a single sketch and contain just a few numbers, the values associated with the deduction categories would typically be higher. Later in the semester when models are more complex, the deductions may be scaled down. To reduce or eliminate the need for changing 12 deduction values, a 13th item has been

introduced. When deduction percentage value is 100, then a 2-point deduction would decrease the total score by two points. If the deduction percentage value is reduced to 50, then the same 2-point deduction would decrease the total score by $0.50 \times 2 = 1$, or just one point.

The remaining two input groups are the File Selector group and the Options group. The File Selector radio button group allows for the comparison of a gold standard model to a single student model, or to all student files in a directory (class assessment mode). Single file comparison is used primarily for debugging and testing. It is envisioned that at some point, students will have access to the program and can use the single model option to compare their model to the gold standard model. Finally, there is an inactive (not yet implemented) option to permit the comparison of any number of selected files to the gold standard model.

The Options group allows the instructor to customize operation of the program for the models being compared. A text field input is available to accept a list of comma-separated numerical values that are optional for both the model and the drawing. For instance, if a value was on the original drawing merely to enforce symmetry, the student may have been encouraged to use a midpoint constraint instead. So that a model with or without the centering value would be graded the same, that value could be input as an excluded value. Another common excluded value is 360, commonly seen in revolved parts or in rotational patterns. In the example of the washer stated previously, one modeler chose to revolve a rectangle about an axis through 360° . But 360 doesn't have to be in the model and it would not be in the model if a second modeler chose to extrude two concentric circles. Similarly, if there was a 45° chamfer of size 2, another modeler may select an equivalent symmetric 2-millimeter chamfer and the 45 would never appear in the model. In these examples, 45 and 360 are angular dimension values that could be specified as "excluded values". There is no penalty assessed if the model or drawing contains an excluded value. It exists to avoid deductions for otherwise optional values.

If the student model layout drawing is not present or not of interest, the instructor would turn off *Verify Drawing Dimensions*, causing review of the student's drawing to be skipped. Additionally, it sets the value of the deduction for *Missing Drawing Dimensions* to zero. This has no effect on the scoring but is done to clarify that the category is inactive. Turning this option on causes the value of the deduction for *Missing Drawing Dimensions* to be reset to the default value.

If the option *Run In Fast Mode* is turned off, the program will show each processed model from an isometric perspective, display its name, and pause until the user wants to resume. This allows the user to get a qualitative sense of the quality of models that are being analyzed.

The last option *Display Verbose Output* allows for displaying a lot of information during model evaluation to the text output window. This is useful mainly during debugging.

Program Initiation and Execution

When the instructor presses the *Compare Models* button, the program obtains the name of the gold standard file and either the name of the one student file or the location of many student files, as set by the File Selector group.

The program analyzes the gold standard model, evaluating all sketches and features, and stores the key numerical values. Geometrical information such as the bounding box height, width, and length is obtained. Additionally, the program extracts surface area, volume, and moment of inertia about each of three axes.

The program then does a similar analysis of one or more student models. Every sketch in the model is examined, whether external or internal. If a sketch was created but never used by any other feature, it is a sign of sloppy modeling. An unused sketch should have been deleted by the modeler. In these cases, the program counts the number of unused sketches, but does not further penalize the model if those sketches are not fully constrained, etc. Sketches that are used by other features are checked that they are fully constrained with no auto dimensions. If a sketch has auto dimensions the program maintains a count of the total number of auto dimensions.

There are several geometrical constraints that students are taught to avoid because those constraints limit the easy changeability of the model. These “banned” constraints (fixed, constant length, and constant angle) are never needed and should not be used. The number of banned constraints is counted.

If the part file also contains a layout drawing, then the list of drawing dimensions is stored. The orientation of each dimension is also examined, as students are asked to always present dimensions with horizontal orientation.

The set of excluded values entered in the GUI’s input text box are subtracted from the gold standard model dimensions to produce a critical list of model dimensions. The list of student model dimensions is compared to the list of critical model dimensions. The student model dimensions that are not in the critical model list are categorized as *unwanted dimensions*. Items from the critical dimension list that are not in the student model are categorized as *missing dimensions*. Finally, the program examines the list of dimensions in the student model and looks for repetitions (*repeated dimensions*). Unless a particular number was on the original drawing more than once, it should not appear in the model more than once. There should be a one-to-one correspondence between numbers in the original drawing and the numbers in the model.

Surface area and volume values are compared to see if the student part is the same shape and size as the gold standard part, regardless of orientation. If the student part has correct size and shape, then its orientation is verified by comparing the moment of inertia components and the bounding box dimensions. However, moment of inertia components are unit-dependent. If the student used incorrect units (for example, inches instead of millimeters), then direct comparison of inertia components is invalid. Bounding box dimensions are independent of units and can be used to crudely verify orientation provided that the model has dimensions where bounding box height, width, and length are significantly different from each other.

However, if the part has a significant missing or incorrect feature but has otherwise correct orientation, use of the moment of inertia components or bounding box dimensions may lead us to an incorrect conclusion regarding orientation.

Extraction of Model Dimensions

A significant portion of the score depends on the total number of missing dimensions, repeated dimensions, and unwanted dimensions. The dimensions of the model must be extracted from the total list of expressions in a model. An expression in its simplest form could be something like “ $p1=5$ ”. The left side of the expression is a parameter name, in this case “ $p1$ ”. The right-hand side of the expression is a text string that may appear to be simply real numbers or they can be rather complex mathematical expressions that can reference other defined parameters, for instance “ $p7 = 17 - (p2 + 23)$ ”. In this case, the program needs to extract all of the unique numerical values on the right hand side so the $p2$ is disregarded and the 17 and 23 are stored. So it’s not a trivial task to extract numerical values from an expression. The program must be able to decipher algebraic equations involving numbers, parameter names, parentheses, and mathematical symbols.

In the right-hand side text string, all mathematical operators are replaced with commas. Each individual text string in the comma-separated text string is analyzed. If an individual text string starts with “ p ” the program assumes that it has identified a reference to another parameter name and disregards it. If an individual text string is a “2” or “0.5” and there is more than one individual string, then that numerical value is disregarded. An expression such as “ $p7=50/2$ ” is commonly seen in various situations. In symmetrical components, to achieve a final width of 50, it can be useful to perform a symmetrical extrude of “ $50/2$ ” (rather than 25) in each direction. The component will have a final width of 50, and the 50 is incorporated in the model by allowing the CAD program do the math. The program would process the right hand side of “ $50/2$ ” by retaining the value 50, and disregarding the 2. Another situation where a divisor of 2 is seen is in a parametric equation for centering. While a midpoint geometric constraint would be preferred, a parametric equation is considered acceptable provided that it does not introduce any extra dimensions into the model. Finally, while it would always be preferable to see circle diameters specified directly with diameter dimensions, it would be acceptable to specify a radius as a diameter value divided by 2. Either way, the critical diameter dimension is in the model and can be quickly located and easily modified.

All dimensions in sketches are stored. All non-zero dimensions in other features are stored. Zeroes in sketches indicate the need for more geometrical constraints. The number of negative values in the student model is counted.

Calculations of part volume, surface area, and moment of inertia pertain to a single body. If the model is composed of various non-united bodies, these bodies are temporarily united for purposes of evaluating shape, size, and orientation.

Comparisons of Computer Grading to Instructor Grading

In Spring 2015, a final exam based on a “back-of-envelope” sketch (Figure 2) was given to a section of 19 students. The students were asked to construct a model and make a layout drawing.

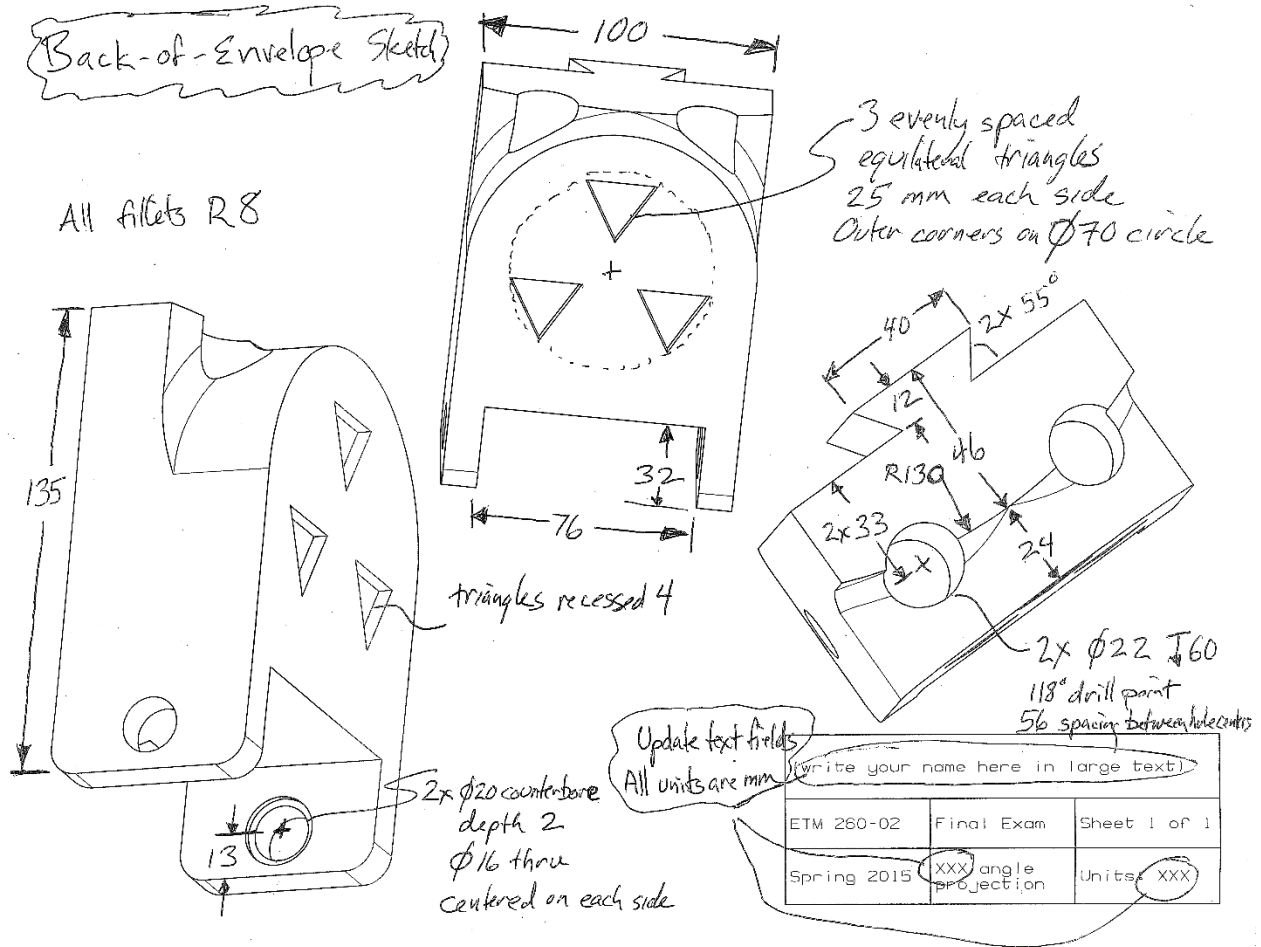


Figure 2. The part to be modeled during the Spring 2015 final exam.

This model requires three sketches and three extrudes to form the main geometry. It is not immediately apparent which sketches and features must be created before others. It is up to the student to study the drawing, understand dependencies, and formulate a modeling plan. Additional sketches are required to place the holes and to create the pattern of recessed triangles. Careful selection of the order of events can lead to a simpler model. For example, if the edge blend operation along the curved edge is done before the two top holes, then it is a simple one-click operation. But if done after the holes, then the curved edge has been broken into three segments and creating the edge blend becomes a clumsier operation.

The final exam models were hand-graded by the instructor at the end of the Spring 2015 semester, before the first line of computer grading code had been written. The instructor reviewed each model by opening it, turning off all features, then sequentially turning the features on in chronological order and recording the number of occurrences of specific issues in a spreadsheet-based scoring system (Table 1). The right-most column shows how a specific student model was evaluated.

Exam 3 Deductions	deduction value	max number of deductions	number of observations in a student model
Model missing dimensions from original drawing			1
Model uses numbers not from original drawing			4
Model has redundant dimensions			4
Total number of dimensional issues in the model	2.5	12	9
Unused/unneeded sketches or datum planes	1	3	0
Issues in part tree (e.g., exclamation points)	3	3	0
Model is unnecessarily complex	3	1	1
Sketch not fully constrained	6	3	0
Incorrect feature or dimension	3	6	1
Other comments:			missing dimensions: 76 repeated dimensions: 55, 8, 46, 24 unwanted dimensions: 30, 22, 260, 12 incorrect radius on front face
Exam 3 Model Score			72

Table 1. The grading spreadsheet showing the evaluation of one student model.

For a given item, there was a limit on the number of deductions. Using this information, the spreadsheet computed a final score:

$$score = 100 - \sum_{i=1}^6 itemDeduction_i \quad \text{Eq. 1}$$

where,

$$itemDeduction_i = deductionValue_i \times \min(nMax_i, nOccurrences_i)$$

For the student model shown in Table 1, the score is thus computed as,

$$score = 100 - 9(2.5) - 1(3) - 1(3) = 71.5$$

Computer Assessment of Student Work

The same deduction values displayed in the user interface in Figure 1 were used, with two changes. First, as we are only interested in examining the model and not the layout drawing, the option *Verify Drawing Dimensions* was turned off and thus the value of the *Missing Drawing Dimensions* deduction was set at zero. Because orientation of the part was not pre-specified and because more than one view of this symmetric part makes an acceptable isometric orientation, the deduction value for *Incorrect Orientation* was also set to zero. The same student model that was evaluated manually in Table 1 was re-evaluated by the computer program (Table 2).

Your model has the correct shape and size.
Your model is missing 2 dimensions from the original drawing: 40, 76.
Your model has 10 repeated dimensions: 8, 12, 20, 22, 22, 22, 24, 46, 55, 130.
Your model has 2 dimensions that were not from the original drawing: 30, 50.
Use fewer numbers and more geometric constraints!!!

Model Score: 72

Table 2. Output of the computer program after analyzing the student model.

The two schemes produced virtually the same score. This is partly because there is some similarity in the algorithms that produce the scores. The computer program employs a score

calculation similar to Eq. 1, except that there are now 12 deduction categories and *Deduction Percent* is incorporated. But this exact correlation is not expected. Note that in the manual grading scheme, some dimension issues were missed due to sloppiness of the instructor. Others were categorized differently (22 is an unwanted dimension in the manual evaluation and multiple instances of 22 are categorized as repeated dimensions in the computer evaluation). Miscategorization of dimensions does not affect scoring, as missing/repeated/unwanted dimensions all carry the same deduction values.

A high-quality model will respond to changes in its dimensions in ways that produce expected outcomes, without surprises. Changeability can be assessed by attempting to change a few pre-determined dimensions in a model². For many semesters, the instructor tested the changeability of each student model by changing a few dimensions and observing what happened. It is still done occasionally, but only for the benefit of the students to demonstrate that models that lack the one-to-one correspondence of original drawing dimensions to model dimensions will have issues when the model needs to change. In the Spring 2015 model, we could modify the height, width, and depth of the part by changing three dimensions: $135 \rightarrow 160$, $100 \rightarrow 120$, and $24 \rightarrow 30$. To change these dimensions in the model, we just locate the first chronological instance of the desired number and update it. Because the model was designed with a set of unique dimensions, if we can locate the desired number, we know it is the number to change.

As noted in the computer output of Table 2, the student model has the exact static geometry as desired. However, due to the lack of one-to-one correspondence of original drawing dimensions to model dimensions, it does not handle change very well. In Figure 3, we see the instructor's model on the left (which was identical in appearance to the as-submitted student model). In the center, we see the instructor's model after the three dimensional changes. It honored the design intent in the original drawing, and just got larger. On the right, we see that the student model developed some issues during resizing. The edge blend was lost, the pair of top holes became un-centered, the radius of the curved surface increased, the gap between the interior side walls increased, the dovetail width increased, and some extra geometry appeared in the back of the part.

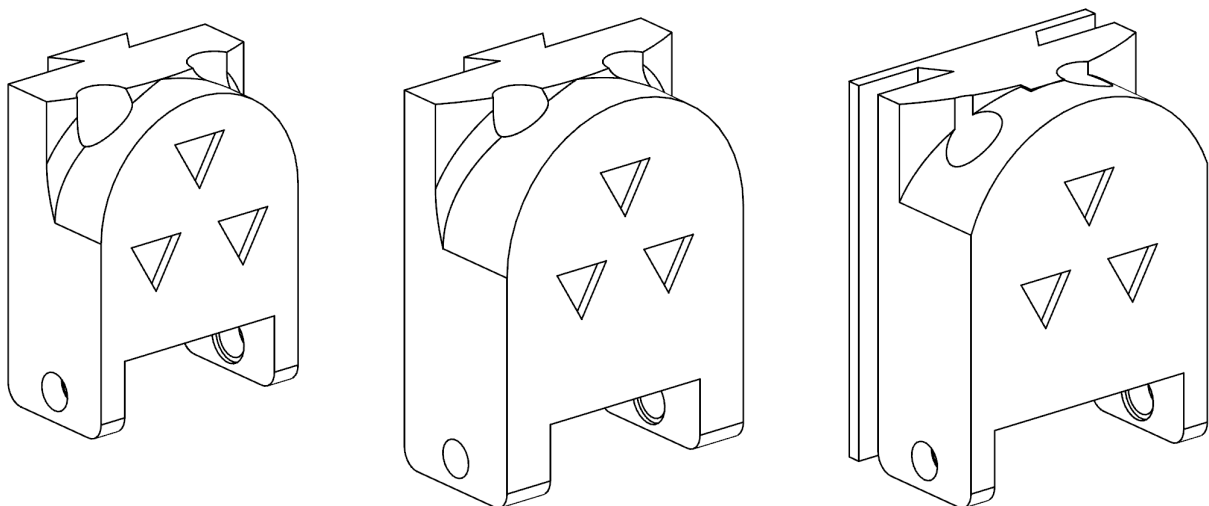


Figure 3. The desired initial model, the instructor's model after three dimensional changes, and the student model after the same dimensional changes.

Computer Evaluation of an Entire Course Section

The computer program was used to evaluate the models of an entire section of 19 students from the Spring 2015 final exam. Results were compared to the previous instructor evaluations of these same models (Figure 4). Data points that lie on the diagonal line indicate perfect agreement between instructor and computer. Data points significantly off the diagonal trendline could indicate a need for tweaking the deduction values or otherwise improving the algorithm. If a data point was significantly above the diagonal trendline, then it could indicate favoritism. If a data point was significantly below the trendline, then it could indicate a negative attitude toward that student.

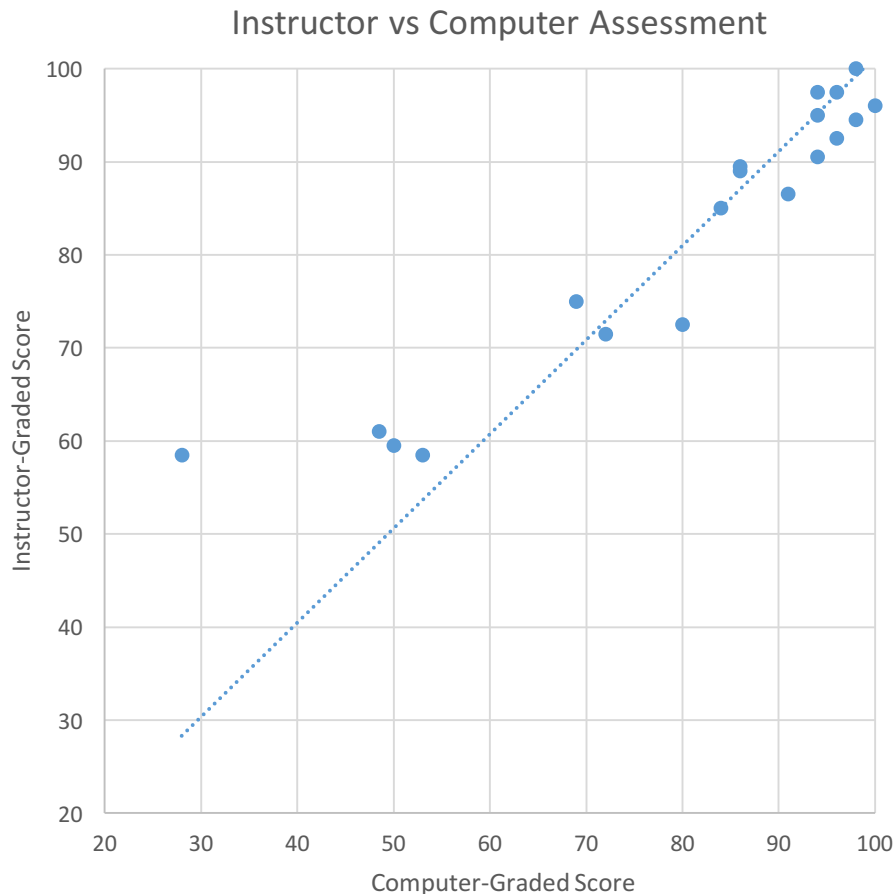


Figure 4. The scoring systems compared to one another.

The coefficient of determination (R^2) can provide some quantifiable information on the correlation between computer-graded models and instructor-graded models. When all 19 student models are evaluated together, $R^2 = 0.65$.

The method is best at differentiating fair and good models from excellent models. Student models that have major issues can actually score lower than an empty model if they have enough non-constrained sketches, missing/repeated/unwanted dimensions, etc. When an empty model (nothing but a coordinate frame) is compared to the Spring 2015 gold standard, it scores 38. The lowest scoring student in Figure 4 scored 28 in the computer-based assessment. Even though the

student demonstrated some capability in various modeling strategies and captured some aspects of the correct geometry, the program has trouble determining this. If we leave out the one outlier student score, $R^2 = 0.86$, demonstrating a strong correlation between the scoring systems.

Program Limitations and Future Work

Outside of volume and surface area comparisons, very dissimilar models can appear similar when reduced to a list of dimensions. For instance, a cylinder with diameter 50 and length 200 will have the same critical dimensions as a square plate that is 200 on each side and 50 thick. The deductions for shape and size as currently set are relatively small, so an inappropriate model with the expected dimensions could score well.

The program does not currently monitor the number of instances of certain non-pattern based features. For example, if an R4 edge blend was specified at three places, but it was used only once (perhaps possibly even on the wrong edge), then the “4” would be extracted from the student model and compared to the “4” in the gold standard model. The lack of the other two edge blends would be ignored, as well as the possible incorrect edge election. Comparisons for shape and size may detect the error, but the default deductions are relatively small.

The deduction values could be adjusted to increase the correlation between the quality of the model and its score. No systematic attempt was made at finding the set of deduction values that would minimize the difference between the scores of the human grader and the computer.

The current version of the program has processed many different models of various complexity, with features including revolves, extrudes, edge blends, chamfers, various hole types, mirrored and/or patterned features, and internal and external threading. Furthermore, versions of each of these models have been produced by 40 or more different students. Although students sometimes use non-taught and unexpected features of the CAD program, the program has been expanded to handle these cases. The computer program now successfully handles all models that it has encountered. Every large set of new models has the potential to reveal some weakness in the program or room for improvement.

Conclusion

Hundreds of hours have been invested in the creation of the program in an effort to save hundreds of hours of grading time in the future. We're already reaping benefits, as the computer program allows for the grading to be thorough, accurate, repeatable, non-subjective, and very fast. It can process models in seconds and generates output very similar to what would've been generated by hand. We feel 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. We find that the computer, unlike the instructor, never suffers from fatigue, does not miss anything that it was programmed to look for, and is not subjective in the assessment of students. It treats all students identically whether they personally have a good relationship or a not-so-good relationship with the instructor.

While the computer code that has been written and evaluated is specific to one particular CAD software and API (Siemens NX), much of the algorithm can be transferred to other CAD software packages that have well-developed APIs.

Bibliography

1. Ault, H. K., & Fraser, A. (2013), *A Comparison of Manual vs. Online Grading for Solid Models* Paper presented at 2013 ASEE Annual Conference, Atlanta, Georgia. <https://peer.asee.org/19045>
2. Ault, H.K. & Giolas, D.T. (2005), *An investigation of solid modeling practices in industry*, Engineering Design Graphics Journal, Vol 69, No.1, pp. 34-43.
3. Ault, H. K., & Bu, L., & Liu, K. (2014), *Solid Modeling Strategies – Analyzing Student Choices*, Paper presented at 2014 ASEE Annual Conference, Indianapolis, Indiana. <https://peer.asee.org/23026>
4. Otey, J. M., & Company, P., & Contero, M., & Camba, J. D. (2014), *A Review of the Design Intent Concept in the Context of CAD Model Quality Metrics*, Paper presented at 2014 ASEE Annual Conference, Indianapolis, Indiana. <https://peer.asee.org/19992>
5. Kirstukas, S. (2013). *A Preliminary Scheme for Automated Grading and Instantaneous Feedback of 3D Solid Models*, Proceedings of the Midyear Conference of the Engineering Design Graphics Division of ASEE, pp. 53-58.
6. Baxter, D., & Guerci, M. (2003). *Automating an Introductory Computer Aided Design Course to Improve Student Evaluation*, Paper presented at 2003 Annual Conference, Nashville, Tennessee. <https://peer.asee.org/11479>