On the effects of the fix geometric constraint in 2D profiles on the reusability of parametric 3D CAD models

Carmen González-Lluch[a]*, Pedro Company[b], Manuel Contero[c], David Pérez-López[d] and Jorge D. Camba[e]

[a]Department, of Mechanical Engineering and Construction, Universitat Jaume I, Castellón, Spain, Av. de Vicent Sos Baynat, s/n 12071, Castellón, Spain, e-mail: mlluch@uji.es ****corresponding author***

[b]Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain, e-mail: pcompany@uji.es

[c]I3B, Universitat Politècnica de València, Camino de Vera s/n, 46010 Valencia, Spain, e-mail: mcontero@upv.es

[d]I3B, Universitat Politècnica de València, Camino de Vera s/n, 46010 Valencia, Spain, e-mail: dapelo@upv.es

[e]Gerald D. Hines College of Architecture and Design, University of Houston, 4200 Elgin St, Houston, TX 77204-4000. USA. e-mail: jdorribo@uh.edu

Carmen González-Lluch, ORCID: 0000-0002-9839-5615
Pedro Company, ORCID ID: 0000-0001-6399-4717
Manuel Contero, ORCID ID: 0000-0002-6081-9988
David Pérez-López ORCID ID: 0000-0002-9293-4892
Jorge D. Camba, ORCID ID: 0000-0001-5384-3253

Abstract

In order to be reusable, history-based feature-based parametric CAD models must reliably allow for modifications while maintaining their original design intent. In this paper, we demonstrate that relations that fix the location of geometric entities relative to the reference system produce inflexible profiles that reduce model reusability. We present the results of an experiment where novice students and expert CAD users performed a series of modifications in different versions of the same 2D profile, each defined with an increasingly higher number of fix geometric constraints. Results show that the amount of fix constraints in a 2D profile correlates with the time required to complete reusability tasks, i.e., the higher the number of fix constraints in a 2D profile, the less flexible and adaptable the profile becomes to changes. In addition, a pilot software tool to automatically track this type of constraints was developed and tested. Results suggest that the detection of fix constraint overuse may result in a new metric to assess poor quality models with low reusability. The tool provides immediate feedback for preventing high semantic level quality errors, and assistance to CAD users. Finally, suggestions are introduced on how to convert fix constraints in 2D profiles into a negative metric of 3D model quality.

Keywords: Automatic feedback tool; CAD; fix constraint; model quality; reusability; 2D profile

Acknowledgements

The authors would like to thank Raquel Plumed for her support in the statistical analysis.

This work has been partially funded by grant UJI-A02017-15 (Universitat Jaume I) and DPI2017-84526-R (MINECO/AEI/FEDER, UE), project CAL-MBE.

The authors also wish to thank the editor and reviewers for their valuable comments and suggestions that helped us improve the quality of the paper.

On the effects of the fix geometric constraint in 2D profiles on the reusability of parametric 3D CAD models

Introduction

The idea that CAD is "just a tool" is surprisingly widespread (Petrina, 2003). Contrary to popular belief, we build on the idea that teaching CAD is a strategic task aimed at developing a particular modeling behavior in students, where parametric CAD models must reliably allow for modifications while maintaining their original design intent. This is a complex objective, as "when students encountered a problem they would display heuristics to satisfy problems without identifying all necessary information" (Buckley et al. 2017). The goal involves breaking natural tendencies to apply procedures, and replace them by more critical competencies of strategic CAD modeling.

With the advent of model-based design paradigms and the ever-increasing reliance of engineering activities on CAD models, the link between design reuse and CAD model reuse is becoming stronger. CAD model reuse involves modifying a master model in its native modeler so it can be applied or adapted to other design situations while guaranteeing model consistency (Hepworth et al. 2014). Along with simplification and exchange, model reuse is considered a transformation that is commonly performed to history-based feature-based parametric models.

Reusability is particularly important in the Model-Based Enterprise paradigm (Camba et al. 2017) where collaborative engineering design and analysis tasks (Red et al. 2013) require all members of a multidisciplinary team to share the same model (Briggs et al. 2015) (for example by employing new multi-user CAD tools (Stone et al. 2017)). In these scenarios, high quality models are essential (Contero et al. 2002). High quality models require identifying the right functional parameters to build the appropriate parametric structure (Bodein et al. 2014), and conveying as much design intent as possible (Camba and Contero 2015).

Models are considered reusable if they are simultaneously flexible and robust (Camba et al. 2016; Cheng and Ma 2017). A model is flexible if it facilitates design alterations, and robust if it prevents design changes from causing unexpected and undesired changes or errors.

History-based, feature-based, parametric CAD models work at three levels: 2D profiles (or "sketches"), modeling operations (or "features") and modeling sequence (also called "model tree," "history tree," or "design tree"). As stated by Company et al. (2015), "simpler features, the use of reference geometry, and the correct feature sequence improve the perception of the model during alteration (Company et al. 2015)," which means that flexibility and robustness must be achieved at all three levels.

A widely accepted prerequisite to create robust profiles is that they must be fully constrained (Company et al. 2015). Some CAD systems provide tools that can automatically constrain profiles to prevent involuntary alterations to the model while it is being edited. However, the flexibility of a profile does not depend on the amount of constraints, but the semantic level of those constraints. According to the three levels of quality for classifying CAD models defined by Contero et al. (2002), the semantic/pragmatic level determines the CAD model's ability for modification and reuse. For instance, it is commonly accepted that automatically constrained profiles are difficult to edit and do not communicate design intent, as automatic constraints are assigned without regard to the function of the model.

We consider geometric constraints as a non-dimensional relationship between elements of a geometric drawing. There are two main types of geometric constraints: those that associate geometric entities to each other (coincident, concentric, collinear, collinear parallel, perpendicular, tangent, smooth, symmetric, equal, etc.), and those that link geometric entities to the reference system, by fixing an angle (horizontal, vertical) or a location (fix constraint). The latter are also known as "ground" constraints (Ault 1999).

Therefore, the theoretical basis that supports that ground constraints, like fix relations, restrict profiles in a way that prevent reusability comes from the fact that they link one geometrical element to the reference system, instead of linking the element to their neighboring geometrical elements. In our view, constraints that link elements of a profile to the reference system—as opposed to creating links between elements—belong to a lower semantic level, as they merely locate parts of the model in the scene but hardly convey design intent. We build on the idea that a fix constraint (or fix relation, according to the terminology used by some CAD systems) is a low semantic type of geometric constraint aimed at locking the position of the entity to which it is added. As non-functional constraints, fix constraints are inefficient and, as described by González-Lluch et al. (2017b), a type of “missing design intent” error in procedural models. Therefore, a large number of profile fix constraints in a model can negatively affect editing tasks and significantly increase editing times. Surprisingly, the fact that using fixed constraints in parametric CAD modeling may reduce reusability is not “common” knowledge, yet CAD instructors should inform students about the potential problems and misuse of these constraints. We have not found previous studies that comparatively examine the negative implications of using fixed constraints versus alternative constraints that convey design intent in a more effective way.

In an effort to measure high semantic level quality aspects of CAD models, we are developing an automatic assessment tool designed to measure a relevant set of new metrics. In this paper, we use this tool to track one of such metrics: number of fix constraints in 2D profiles. Our prototype demonstrates the usefulness of the new quality metric as well as the feedback mechanism to CAD trainees, which is a first step to release instructors from routine checks.

The problem of fix constraints addressed in this paper represents a step toward the creation of parametric models of higher semantic quality. In particular, our results contribute to facilitating automatic semantic quality assessments of 3D models. From a practical standpoint, the study also yields valuable insights into how CAD tools are used, and should be used, by designers, which can then inform new methodologies and even new functionalities that can be implemented by CAD systems.

The paper is structured as follows: we initially review the state of art in the area of metrics of quality in parametric sketches to conclude that, to the best of our knowledge, the only generally accepted metric is the one that indicates whether or not a profile is fully constrained. Next, we describe an experiment to determine whether there is a correlation between the number of fix constraints in a profile and the time required to alter such profile. The analysis and results of the experiment are discussed in section “Experiment 1: Results and discussion,” where we conclude that a statistically significant relation does exist. In section “Experiment 2: Automatic detection of poorly constrained profiles,” we describe the implementation of a prototype add-in for a commercial 3D CAD system to automatically assess the use of fix constraints as a metric to determine the quality of profiles. A pilot user study is described in the following section, where we demonstrate the value of the tool both as a mechanism to automatically identify poor quality models as well as a method to provide feedback to both novice and expert users and automate repetitive assessment tasks. Finally, we conclude by describing our vision that the quality of a profile is linked to the quality of the restrictions used to define it. Additional metrics must be defined in order to get a comprehensive and accurate measurement of the overall quality of a profile.

Related work

Most CAD modelers build 3D shapes by sweeping 2D profiles. In parametric systems, it is common practice to add geometric constraints and, subsequently, dimensions to an approximate profile so the final accurate shape can be calculated. Several authors have investigated the universe of geometric constraints as well as suitable approaches to solve constrained profiles. Kramer (1991) proposed the use of degrees of freedom analysis to solve geometric constraint systems. Other authors such as Ge et al. (1999) used numerical optimization methods to solve geometric constraint problems (under- and over-constrained problems). The development of 2D constraint solvers has also been reported (Bouma et al. 1995; Fudos and Hoffmann 1997; Mata 1997; Leea and Kimb 1998; Ait-Aoudia and Foufou 2010).

Most CAD systems can automatically add inferred (or “snapped”) constraints to specific elements of a profile while the profile is being drawn based on how the user creates the geometry. This strategy is helpful to produce robust profiles quickly, but requires skilled users to avoid unnecessarily over-constrained profiles that may result from the automatic detection of undesired constraints (such as a “nearly” horizontal line that is incorrectly constrained as horizontal).

The manual insertion of geometric constraints and how to solve over- or under-constrained profiles have also received attention in the scientific literature (Joan-Arinyo et al. 2003; Ault 2004). Ault (2004) applied a method to determine the number of degrees of freedom for a set of geometric and dimensional constraints in a profile. A recent work by Dixon and Dannenhoffer III (2014) goes slightly further by reporting techniques for helping users to properly solve profile problems. The authors revisit the idea that providing users with information about profile constraint failures can help them build more efficient profiles. Ault (1999) showed how constraints may be applied to create flexible geometric models by avoiding undesirable geometry or topology changes in models. Related studies on procedural knowledge in CAD have also been reported. For instance, the work by Szewczyk (2003) focused on the difficulties derived from misunderstandings of graphic elements in the user interface. However, neither learning nor checking strategic knowledge in CAD models during instruction have received the same attention.

Modeling for reusability is not trivial. Researchers Jackson and Buxton (2007) estimated that 48% of all CAD models fail after changes. Failure means producing CAD models that contain errors or anomalies, which typically require the models to be reworked. According to Company et al. (2015), a profile is robust as long as it is fully constrained, and it is flexible if its shape, size, position, and orientation are fixed independently. In fact, a common method to determine the quality of 2D profiles is by quantifying their degrees of freedom (DOF), as this number is always positive for profiles that are not fully constrained. This metric is typically represented in many CAD systems as icons in the model tree that indicate unconstrained profiles or color coded schemes that highlight unconstrained geometric elements. A more recent study approaches 3D model reuse from the standpoint of how users learn CAD and how design intent can be used during learning Barbero et al. (2016). Attempts have also been made to define different sub-types of geometric constraints, such as the classification of "ground" constraints by Ault (1999).

A review of the scientific literature brings out the need to improve formative assessment in CAD instruction. According to Race (2001) feedback should be continuous in a formative sense (Race 2001). However, there is a lack of tools that link quality criteria, provide feedback and facilitate student assessment. After examining a representative commercial Model Quality Testing (MQT) tool, authors González-Lluch (2017a) discussed the need for mechanisms to assess higher semantic level quality aspects.

A study by Kirstukas (2016) showed that a prompt feedback is essential to improve the modeling strategies of students (Kirstukas 2016). In his work (Kirstukas 2016), students received an objective assessment via an automatic tool which compared their NX solid models against those created by instructors. The tool can automatically assess the geometry and changeability of students' solid models.

In similar studies, automatic grading systems were created to compare models against a template provided by the instructor (Ault 2013) or provide students with a list of discrepancies and images that highlight the differences between their solutions and an answer key (Hekman 2013).

In a recent work, Kwon et al. (2015) proposed evaluation metrics based on contact and coincident constraints between assembled parts. Their goal was to identify geometric elements that can be removed from a model so it can be simplified without compromising connectivity. Ault et al. (2014) used metrics to evaluate solid part model complexity and factors associated with the quality of a modeling strategy to ultimately shift CAD instruction from procedural to strategic knowledge (Chester 2007), as "the responsibility of design education goes well beyond teaching technical proficiency in the use of CAD tools" (Robertson et al. 2007). According to Hamade (2009), this approach makes the difference between "CAD users who are capable of flying on the tube and delivering in record times and those who are capable of building sophisticated models."

The detection of inappropriate uses of fix constraints is not currently supported by CAD quality testers. In a recent state of the art review by González-Lluch et al. (2017b), the authors stated that Model Quality Testing tools "are mostly aimed at homogenizing the vast amount of documents produced and shared by large OEM's," but they will only become valuable to Small and Medium Enterprises "if document homogenization ceases to be prevalent over conveying design intent." Therefore, defining and testing metrics to measure high semantic level quality aspects in CAD models is a relevant unsolved issue (Summers and Shah 2010; Kirstukas 2016), that is reflected in a lack of reliable quantitative metrics to measure the flexibility of the profiles. To the best our knowledge, the only commonly accepted metric is the one that assumes that under-constrained profiles are not robust, and dichotomically measures robustness by distinguishing between fully constrained or not.

Experiment 1: Correlation between fix constraint and profile reusability

Our vision for this first experiment is that profiles, even when fully constrained, may be poorly constrained if they are constrained by relations of low semantic level that do not convey design intent. We hypothesize that the fix constraint (a point in a profile is fixed in one location) is one of such low semantic level constraints.

Although fix contraints belong to an ancillary approach, derived from non-parametric drawing strategies, the authors are aware that naturally some exceptions exist. For example, fix constraints may be useful to lock the internal parameters that define the shape of free curves and surfaces, such as the control points of a spline. Nevertheless, this is not representative of the way they are misused by novel users. Our experience show that the common use is to carelessly introduce 'fix constraints' to fully constrain 2D profiles in 3D models, compromising the reusability of the 3D model.

The goal of this study is to check whether there is a correlation between the number of fix constraints used to control a 2D profile and the time required to modify such profile.

A typical shape used to train novice students was selected for this experiment (Figure 1). This profile is commonly used as part of the training of different groups of engineering students (Bachelor's Degree in Mechanical Engineering, Bachelor's Degree in Industrial Technology Engineering, Bachelor's Degree in Industrial Design and Product Development Engineering). The authors proposed and evaluated a large set of candidate profiles. As a result of the analysis and discussion of their resemblances and differences, we concluded that the simplicity of straight line profiles was preferred (as it avoids undesired complexities such as tangent relationships). The presence of local symmetries and oblique orientations was desired as these are items commonly related to an increase in the editing difficulties.

From the selected profile, three different sets of constraints were applied to three versions of the profile created with a 3D CAD application (SolidWorks®). Identical dimensional constraints were used for all three profiles, as well as most of the geometric constraints. However, each version of the profile was defined with an increasingly greater number of fix constraints (Figure 2 to 4). In order to decide the range of fix constrains that produce noticeable delays while editing, the authors made preliminary tests that showed that profiles with four fix constraints required a perceptible increase in editing time. We asked various engineering instructors who were not involved in the experiment to edit profiles with different number of fix relations, to gain an informal insight on the range of fix relations that they could manage (and succeed) in a limited time of 15 minutes. Thus, experimental profiles were defined as follows: Profile Type 1 contained no fix constraints (Figure 2); Profile type 2 contained four fix constraints (Figure 3), and Profile Type 3 included eight fix constraints (Figure 4). The fix constraints were manually added by replacing other constraints, which were selected by informally replicating the places where our students usually place them.

Figure 1. Reference sketch

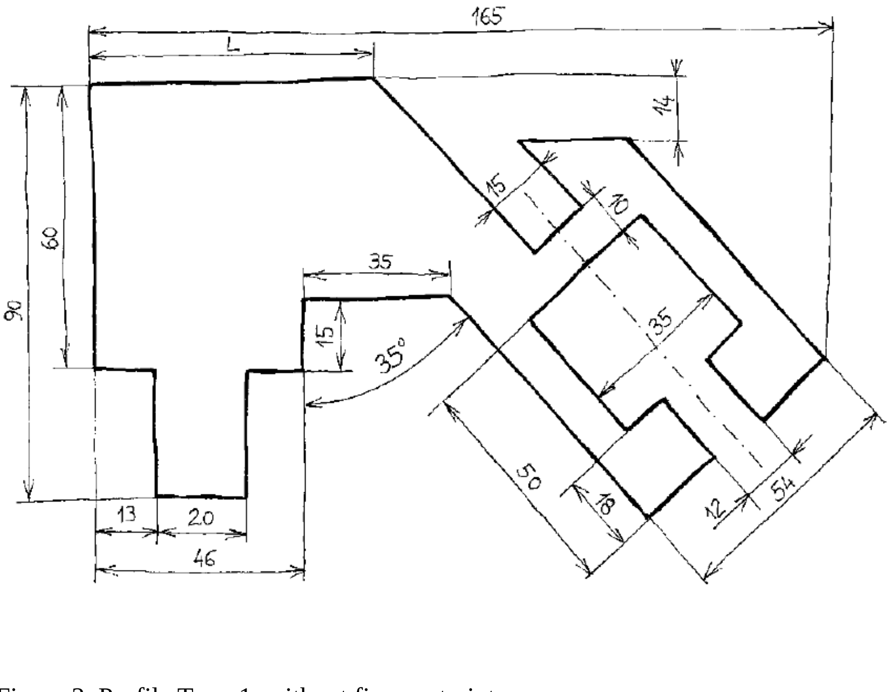


Figure 2. Profile Type 1: without fix constraints

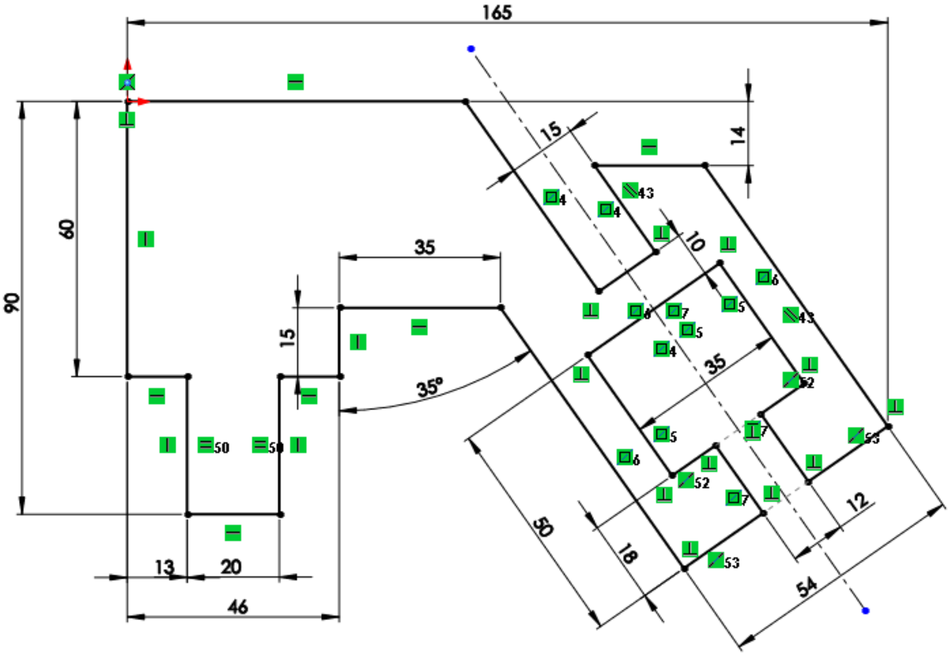


Figure 3. Profile Type 2: with four fix constraints

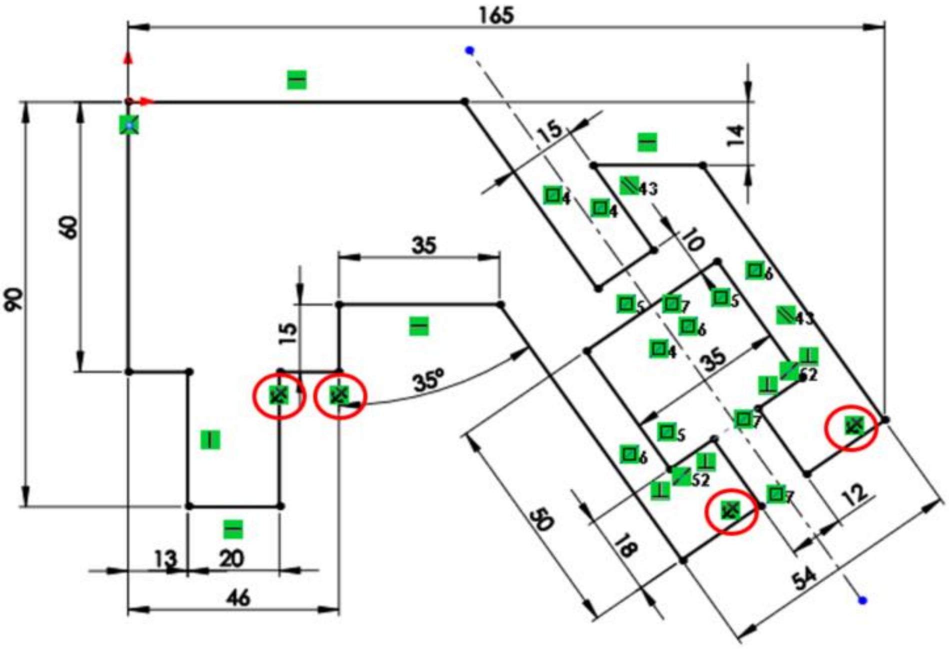
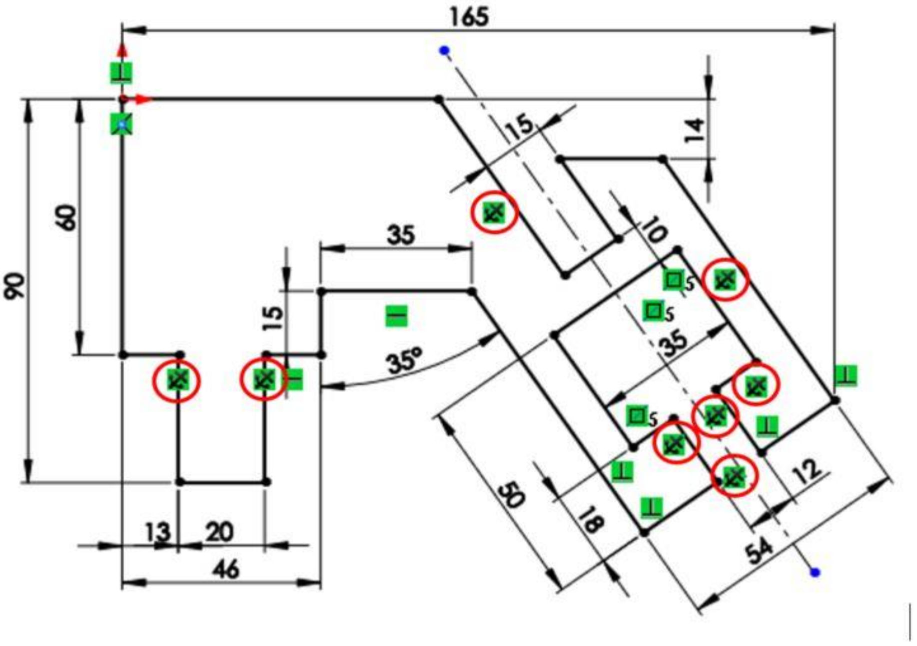
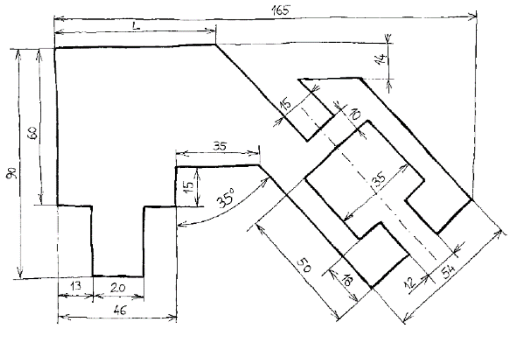


Figure 4. Profile Type 3: with eight fix constraints

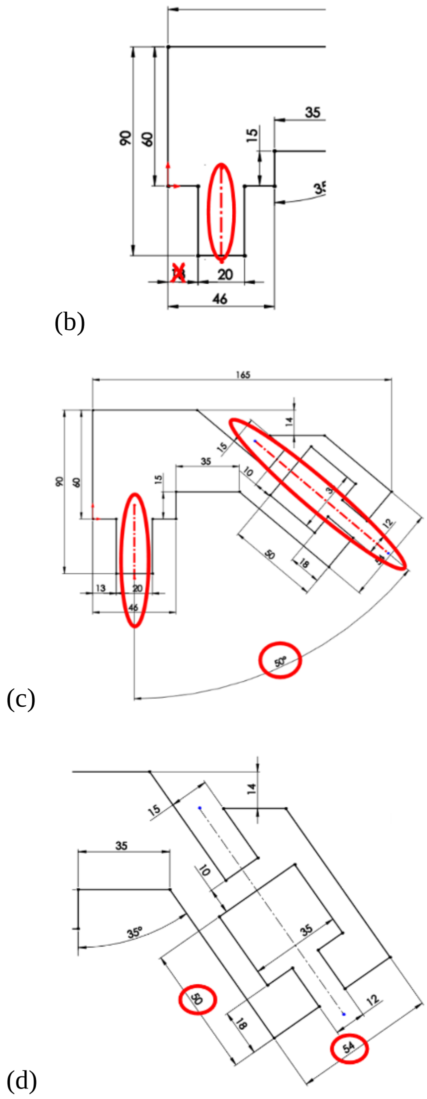


To simulate model reuse tasks, the following changes were defined: Change 1, define a local symmetry in the left part of the profile and remove the 13mm dimension (Figure 5a); Change 2, replace the 35° dimension with a 50° dimension between the two local symmetry axes (Figure 5b); Change 3, increase the 40mm dimension to 50mm and the 54mm dimension to 60mm (Figure 5c).

Figure 5. Illustrations of three Reference sketch (a) and different changes 1 (b), 2 (c) and 3 (d)



(a)



Two groups of participants (group “students” and group “instructors”) were selected. All participants in the “students” group were junior students from various engineering majors: mechanical (38 participants), industrial (18 participants) and industrial design engineering (51 participants) at Universitat Jaume I.

All participants had comparable knowledge and abilities in terms of CAD, and had to apply computer-aided three-dimensional modeling (3D CAD) for the resolution of graphic engineering problems. Throughout the course, students attended both theory and practical classes.

In the beginning of the course, students were introduced to the creation of 2D profiles based on the first chapter of Company et al. (2013). In theory classes, students were instructed on constraints and learned to distinguish between over-constrained and under-constrained profiles. During lab hours, students gained practical skills on the use of a 3D CAD application (Solid-Works®).

The “instructors” group was comprised of engineering faculty from Universitat Jaume I, all with extensive knowledge in the use of SolidWorks®.

All participants were equipped with a workstation and the appropriate CAD software. Subjects of both groups were randomly assigned one of the three types of profiles and asked to perform the alterations. A custom SolidWorks® macro was developed to measure the time taken to perform the tasks. To ensure repeatability of our study, some details regarding the development and implementation of the macro are explained. The macro ran at CAD system startup and registered the times when the file was last opened and last saved, in addition to the aggregate editing time. The information was saved as custom properties within the model file. The main limitation of the tool is that researcher has to manually extract these custom properties. We use a different solution in the experiment 2 in order to automate this task. A macro was preferred over a plug-in, as the latter would require administrative rights in the participants’ computers and this condition was not guaranteed. The macro required the addition of a non-visible form and its instantiation from the main sub. The form declares the object (i.e., the Solidworks® model file) as WithEvents, so FileSaveNotify and FileOpenNotify callbacks become available. Custom file properties were used to store time information inside the model file. These properties are created by the macro the first time the file is opened, and updated every time the file is saved. Our first attempt was to attach the macro to the model file by using the SolidWorks® Design Binder and create an equation to auto load the macro. However, the approach proved ineffective as it depended on the particular version of SolidWorks®. Additionally, repeated instantiations of the macro triggered the events multiple times which resulted in incorrect time measurements. Consequently, we decided to load the macro by running SolidWorks® with the “-m” parameter via a batch file (.bat). This action loads the macro only once (at SolidWorks® startup) but not the file, which must be manually loaded by the user.

Experiment 1: Results and discussion

A total of 76 modified profiles were collected from the “students” group, but 14 were rejected for various reasons: the file was not saved correctly (8 items), the work was not saved correctly (5 items), and malfunction of the macro (1 item). A total of 21 completed items were analyzed for Profile Type 1, 21 for Profile Type 2, and 20 for Profile Type 3. In the case of the “instructors” group, 4 valid items were collected for Profile Type 1, 5 for Profile Type 2 and 5 for Profile Type 3.

To validate our hypothesis "large numbers of profile fix constraints in a model negatively affect editing tasks and significantly increase editing time", we applied an analysis of variance (ANOVA) to verify whether statistically significant differences exist in the mean times required to edit the different samples.

We considered three different samples for each group of participants, based on the number of fix constraints in the profile (0, 4, or 8 fix constraints). The mean time required to edit each sample was the independent variable. For the collected items in the "students" group (n=61), and considering an effect size of 0.4, with α= 0.05, the statistical power of the analysis reaches 79%.

A summary of the data collected from the "students" group is shown in Table 1. Note the differences in mean times for each task: 4min 42sec (Profile 1), 9min 58sec (Profile 2) and 13min 06sec (Profile 3). The more fix constraints in the profile, the more time required to finish the task.

A single factor ANOVA (time for each type of profile) among groups (defined by the number of fix constraints in the profile) was performed. The null hypothesis (H_0) was defined as: there is no difference between the mean runtime among the three groups. Analysis of variance revealed a significant effect of the number of fix constraints on the time for each type of profile, *F*(2, 59)=14.69, *p*= $6.64 \cdot 10^{-6}$. Thus, H_0 was rejected since the probability was less than the significance level of α = 5% (which is the probability that the observed difference is the result of chance).

Table 1

Data summary for "students" group after performing the three tasks in the assigned profile

GROUP Students	Mean time (St. Dev.) (min:sec)	Fully perform Change 1 (%)	Fully perform Change 2 (%)	Fully perform Change 3 (%)	Perform fully constrained profile (%)	Perform symmetry after change 1 (%)	Profiles maintain shape after 3 changes (%)
Profile Type 1	4:42 (2:19)	66.70	71.40	100	85.70	52.40	100
Profile Type 2	9:58 (5:16)	100	71.40	85.70	81	9.50	52.40
Profile Type 3	13:06 (6:35)	95	50	60	75	35	95

The percentage of students that successfully performed Changes 2 and 3 decreased for profiles Types 2 and 3 (with more fix constraints), as shown in Table 1. Furthermore, the percentage of students who submitted a fully constrained profile was lower in the case of profile Type 3. The number of profiles that maintained their shape after 3 changes decreased in the case of profiles Type 2 and 3. These results support the idea that the use of relations of low semantic level negatively affects editing tasks.

A summary of the data collected from the "instructors" group is shown in Table 2. All available instructors with significant modeling experience in the department participated in the experiment. The mean times used to finish the tasks were: 3min 37sec (Profile 1), 6min 34sec (Profile 2), and 14min 35sec (Profile 3). The ANOVA in this case also rejects the null hypothesis *F*(2, 11)=13.13, *p*<.05, and reveals that there are significant differences, in times when instructors perform editing tasks in profiles with different number of fix constraints. The more fix constraints in the profile, the more time required to finish the task. Once again, the mean time was higher in the case of Profile Type 3.

Table 2

Data summary for the "instructors" group after performing the three tasks in the assigned profile

GROUP Instructors	Mean time (St. Dev.) (min:sec)	Fully perform Change 1 (%)	Fully perform Change 2 (%)	Fully perform Change 3 (%)	Perform fully constrained profile (%)	Perform symmetry after change 1 (%)	Profiles maintain shape after 3 changes (%)
Profile Type 1	3:37 (0:54)	75	100	100	75	50	100
Profile Type 2	6:34 (1:40)	80	80	100	80	80	80
Profile Type 3	14:35 (5:17)	100	80	100	60	20	100

Finally, the results obtained from the "students" and "instructors" groups were compared to determine whether the participants' backgrounds could affect the time needed to make the changes to the drawings. By comparing the mean times for instructors and students, and applying ANOVA, results show that there is no significant difference in the average times between both groups, *F*(1, 2)=.5, *p*>.05, which means that both groups seem to behave equally for each profile type. In addition, we observed that the success rate for the completion of the editing tasks is generally higher for the "instructors" group than it is for the "students" group. In this regard, instructors tend to go further than students when working on editing problems, despite the added difficulty of dealing with fix constraints.

Experiment 2: Automatic detection of poorly constrained profiles

One long-term goal of our research is the development of software tools to automatically evaluate poor quality CAD models.

In this context, it was argued in the introduction that models are reusable if they are simultaneously flexible and robust, where flexibility and robustness must be achieved at three levels: 2D profiles, modeling operations and modeling sequence. Therefore, profiles have poor quality if they lack robustness or flexibility. Profiles are robust if they are fully constrained, thus allowing them to be edited without causing unexpected failures. Hence, robustness can be measured as a ratio between the number of unconstrained geometrical elements and the total number of elements in the profile. Simultaneously, a profile is flexible if it allows for many changes and eases the editing process, enabling re-design. The hypothesis is that fix constraints (i.e., geometric constraints that link an element's position to the reference system) reduce flexibility. Consequently, we can measures flexibility (or lack thereof) by counting the number of fix constraints in a profile.

We designed and implemented a pilot assessment tool based on these two metrics. The tool was designed as an executable file using Microsoft Visual Basic .NET®, and the SolidWorks® Application Programming Interface (API).

Our tool loads an instance of SolidWorks® which, in turn, loads a 3D model and automatically traverses all the features in its history tree, looking for profiles (i.e., features belonging to the "Sketch" type group) to identify their fix constraints (swConstraintType_FIXED and swConstraintType_FIXEDSLOT relations). The tool also identifies and counts entities that are not auxiliary, i.e., profile vertices and edges (to calculate the ratio of constrained elements by profile).

The tool can work interactively or as a batch process. When in batch, the tool parses all the files inside a specific folder and generates a .csv file that contains the ratio of unconstrained elements to the number of fix constraints. This information is calculated for every profile in each file and tabulated for all the files in the folder. In interactive mode, the information is displayed via dialog boxes within the modeling environment. This tool can automatically process a high number of files in a short time.

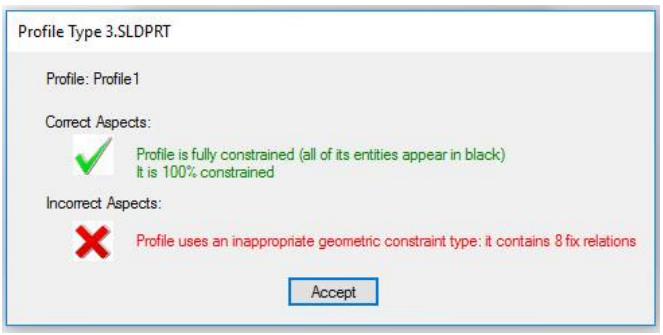
To evaluate the tool, the batch version was first used in a preliminary experiment where several instructors evaluated modeling exercises to determine the prevalence of fix constraints and identify profiles that were not fully constrained. Old exams that had been evaluated using the quality rubric described by Company et al (2015) were parsed, and the evaluation produced by the tool was compared against the corresponding criteria of the rubric (profiles consistency). The results showed a high correlation. Only minor differences appeared, mainly due to human errors, as the tool helped to detect some exams that had been incorrectly evaluated as valid when they included non-constrained profiles.

An additional experiment with students was also conducted. For this study, the interactive version of the MQT tool was used to measure the impact of the feedback provided by the tool. Students were provided with a copy of Profile Type 3 defined in Figure 4 and asked to evaluate it using the tool.

All participants in the second experiment were students from Universitat Jaume I. Participants were distributed into two sub-groups. Participants in the first sub-group (QR1) were industrial engineering students (third year). Participants in the second sub-group (QR2) were industrial engineering and design engineering students (third year). These students followed the same training than the participants in 'Experiment 1.'

The tool was tested with two sub-groups of students. The tool's interface design was guided iteratively by direct user feedback and observation, i.e., findings derived from the data collected informed the design elements next incorporated into the tool. The process was executed several times until the findings did not differ significantly. An example of a quality report shown by the custom MQT tool in interactive mode is shown in Figure 6.

Figure 6. Quality report of the profile with eight fix constraints as shown by the custom MQT tool in interactive mode



Profile Type 3.SLDPRT

Profile: Profile1

Correct Aspects:

✓ Profile is fully constrained (all of its entities appear in black)
It is 100% constrained

Incorrect Aspects:

✗ Profile uses an inappropriate geometric constraint type: it contains 8 fix relations

Both sub-groups (QR1) and (QR2) were asked to edit the profile shown in Figure5 as in Experiment 1. During the design of the experiment, the interface of the MQT tool was improved. In the case of group QR1, the experiment was performed after a two hour session of parametric modeling training. A total of 18 modified profiles were submitted, but 5 were rejected (4 files not saved correctly and 1 macro malfunction). Before obtaining an improved version of the tool (corresponding only with the interface), students in the second sub-group (QR2) were asked to edit the same profile (Figure 5), but the task was assigned after participants received an average of 16 hours of training. In this case, a total of 84 modified profiles were submitted, but 25 were rejected: 17 files not saved correctly, student submitted the wrong file (1 item), student had significantly less hours of training than the rest (2 items), or the task was incomplete (5 items). Therefore, 59 modified profiles were analyzed.

Experiment 2: Results and discussion

The batch application proved valuable as an evaluation tool. Instructors reported that they could quickly and easily detect mistakes that typically take longer to grade or pass unnoticed. The tool can release instructors from routine checks, allowing them to devote more time and effort to quality errors and modeling aspects of higher semantic level.

Some lessons learned from this automatic evaluation are described. First, we observed a low prevalence of incorrect uses of fix constraints in profiles (at least for students who have been advised against its use). After scanning hundreds of models, we estimate that approximately one out of every fifty students uses fix constraints as ordinary constraints. The recommendation to "avoid fix constraints" seems to be easy to remember and put into practice. Second, occasional "good" uses of fix constraints were also detected. Some students use fix constraints as a last resort to fix under constrained profiles when they cannot find a better alternative. In many cases, they manage to do it without compromising flexibility. For example, profiles that contain symmetry axes whose endpoints are not constrained are marked as under-defined by most CAD applications. We say these endpoints have "tolerable" degrees-of-freedom, as they do not reduce the robustness of the profile. Nevertheless, many students applied fix constraints to those endpoints so the application would not mark the profile as under constrained. Third, certain standard modeling operations provided by the CAD system use fix constraints (e.g., Solidworks' Hole Wizard®). Therefore, any metric that uses the number of fix constraints to determine the quality of profiles should consider the role of these automatically added fix constraints in the model.

The interactive version of the application did not prove useful as a self-evaluation tool for novice users (QR1). Most students were unable to take advantage of the feedback provided by the system. Students required so much advice from instructors that their self-performance was compromised. Furthermore, novice students that failed to fully constraint a profile were also unable to fully constrain it after being warned by the tool. We suspect that high level quality criteria needs to be reinforced by quality testers only after a longer training period.

The information provided in Table 3 shows a comparison between students who participated in experiment 1 (see Profile Type 3 in Table 1), and the two sub-groups of students who participated in experiment 2.

Table 3

Data summary before (Profile Type 3 in Table 1) and after the use of a Quality report (QR1 and QR2)

Type Students	Mean time (St. Dev.) (min:sec)	Fully perform Change 1 (%)	Fully perform Change 2 (%)	Fully perform Change 3 (%)	Perform fully constrained profile (%)	Perform symmetry after change 1 (%)	Profiles maintain shape after 3 changes (%)
GS (Profile Type 3, Table 1)	13:06 (6.35)	95	50	60	75	35	95
QR1	20:58 (7:45)	100	84.60	92.30	84.60	92.30	100
QR2	19:21 (10:27)	100	100	97	88	93	93

Based on the information from Table 3, we speculate that students in groups QR1 and QR2 got better success rates than those in the GS group. If so, it would be an indicator that the tool provided valuable feedback to students by showing quality errors and modeling aspects of higher semantic level. This question led us to state an additional hypothesis: “Students that use the tool solve the profile more efficiently and perform the changes more suitably than those who do not use it”.

To validate this hypothesis, we analyzed differences between samples. Examples in which the six analyzed tasks were correct were considered successful. A contingency table was applied to compare the success rate between groups. The observed (Count) and expected (Exp. Count) results for each group are shown in Table 4.

Table 4

Rate and Groups, Cross Tabulation

Rate		Groups			Total
		GS	QR1	QR2	
Success	Count	15	4	12	31
	Expected	6.70	4.40	19.90	31
Failure	Count	5	9	47	61
	Expected	13.30	8.6	39.10	61
	Total	20	13	59	92

To determine whether there is a difference in success rates between groups, we contrast the null hypothesis (H_0 “There is no difference in success rates between groups”) using a Chi-Square Test of Independence. The relationship between these variables is significant, X^2 (2, *N*=92)=20.033, *p*<.05. This test suggests that there are significant differences in the success rates of the groups. The use of the MQT tool has an impact on the tasks performed by students.

A comparison of the success rates between GS and QR1 are shown in Table 5.

Table 5

Comparison of the success rates between GS and QR1

Rate		Groups		Total
		GS	QR1	
Success	Count	15	4	19
	Expected	11.50	7.75	19
Failure	Count	5	9	14
	Expected	8.50	5.50	14
	Total	20	13	33

The relationship between the success rates and groups GS and QR1 is significant, X^2 (1, *N*=33)=6.31, *p*<.05. We conclude that there are significant differences between the success rates of GS and QR1.

A comparison of the success rates between GS and QR2 are shown in Table 6.

Table 6

Comparison of the success rates between GS and QR2

Rate		Groups		Total
		GS	QR2	
Success	Count	15	12	27
	Expected	6.80	20.20	27
Failure	Count	5	47	52
	Expected	13.20	38.80	52
	Total	20	59	79

The relationship between the success rates and groups GS and QR2 is significant, X^2 (1, *N*=79)=19.838, *p*<.05. We conclude that there are significant differences between the success rates of GS and QR2.

Finally, a comparison of the success rates between QR1 and QR2 are shown in Table 7.

Table 7

Comparison of the success rates between QR1 and QR2

Rate		Groups		Total
		QR1	QR2	
Success	Count	4	12	16
	Expected	2.90	13.10	16
Failure	Count	9	47	56
	Expected	10.10	45.90	56
	Total	13	59	72

The relationship between the success rates and groups QR1 and QR2 was analyzed by a Pearson chi-square test (with Yates continuity correction). Results led us to conclude that there are no significant differences between the success rates obtained in group QR1 and QR2, X^2 (1, *N*=72)=.203, *p*>.05.

The results (showed in Table 3) are not surprising since all participants have a basic training using the MQT tool. Despite the different interface tool the QR1 group used an earlier version of the MQT tool with a more rudimentary interface) and the amount of training received (the QR1 group received less training than those in QR2) our study shows that there are no significant differences between both groups, and students seemed to apply similar strategies to locate fixed constraints when designing a profile.

Further discussion and future developments

Our first goal was to demonstrate that fix constraints make the reuse of profiles of CAD models difficult. Our second goal was to demonstrate that software mechanisms that measure the quality of CAD models based on quality metrics are both viable and useful. Both goals are clearly related to each other, but we strongly believe that two separate experiments were required. Otherwise, it could have been argued that an untested tool was biasing the behavior of the subjects, the collection of the information, or its analysis. Similarly, it could also been argued that a non-validated metric was being used to automatically measure quality of CAD models. Therefore, we have independently demonstrated (a) the negative effect of using fix constraints on the re-usability of CAD models, and (b) the usefulness of an automatic assessment tool based on quantitative metrics of CAD models quality.

However, sub-dividing the participants of the second experiment in different groups while the tool was being improved was not a desirable strategy. Only after the experiment was completed, did we realize that the influence of this strategy had not affected the validation of the hypothesis. In this regard, a single combined experiment could have been used as an alternative more efficient approach, which would have also resulted in a shorter explanation. That said, for the sake of repeatability, we have tried to clearly explain our actual experimental procedure.

Regarding the tools, it is to be noted that some similar testing tools can be found in the literature. For example, Cheng et al. (2018) developed and tested a tool to analyze three models by extracting its history tree feature by feature and then calculating various centrality parameters. Similarly, Kirstukas (2016) developed a tool that can automatically assess the geometry and changeability of student solid models..

The main strength of the macro developed for the experiment 1 is its ability to measure time accurately without adding manual triggers. We consider the fact that the researcher has to manually extract the information stored in the custom properties as its main limitation.

Alternatively, the tool developed for experiment 2 automatically writes a .csv file containing the results for all the parsed documents, which eliminates human error during file processing.

But, what we see as the main strength of our approach is that it is based on metrics directly derived from quality concepts that have been validated independently from each other.

Conclusions

Current commercial history-based parametric CAD applications provide mechanisms (or at least information) to effectively work with both under-constrained and over-constrained sketch profiles. However, there is no support to manage fully constrained profiles that use low semantic constraints. This type of profiles is simply accepted by the system without any feedback or warning to the user.

In this paper, we have demonstrated that low semantic level constraints such as the fix constraint, may result in poorly constrained profiles that compromise the flexibility of the 3D model. The inappropriate use of fix constraints may hinder reusability, as these constraints have been shown to negatively impact editing tasks and significantly increase editing times.

Since CAD models typically contain many profiles, checking whether or not those profiles are poorly constrained can quickly become a tedious and time-consuming task. In an effort towards the development of a system that can automatically detect (and potentially correct) bad constraining practices, we have implemented a tool to track fix constraints in parametric profiles. The tool was tested in a pilot experiment where instructors evaluated students' work and students self-evaluated CAD models before editing them. This experience sheds lights on the idea of implementing mechanisms to quantitatively assess the quality of parametric CAD profiles.

Certainly, stating that fix constraints are poor quality constraints may derive in useful lessons for future trainees. However, the main goal of our research is to find suitable metrics to automatically evaluate the global quality of CAD models by helping detect low quality master models that may compromise efficiency in a Model Based Enterprise paradigm.

As future work, we envision the development of quick and efficient mechanisms to perform quality assessments of parametric profiles by analyzing low semantic constraints (for example, by determining the percentage of profile elements that get locked when a fixed constraint is applied). We hypothesize that profiles that are over-constrained with redundant but compatible relations are even more difficult to edit than those that are fully constrained but have low semantic constraints. The validation of this hypothesis is a natural next step. We will then measure both redundant and low semantic constraints to combine their metrics and measure the unnecessary complexity of profiles that may prevent them from being flexible and reusable.

References

Ait-Aoudia, S., & Foufou, S. (2010). A 2D geometric constraint solver using a graph reduction method. *Advances in Engineering Software, 41(10),* 1187-1194. https://doi.org/10.1016/j.advengsoft.2010.07.008.

Ault, H. K. (1999). Using geometric constraints to capture design intent. *Journal for Geometry and Graphics, 3(1),* 39-45.

Ault, H. K. (2004). *Over-Constrained, Under-Constrained or Just Right? Goldilocks Evaluates DOF of Sketched Profiles*. Paper presented at American Society for Engineering Education, 59th Annual Midyear Meeting Past, Present & Future?, Williamsburg, November 21-23.

Ault, H. K., Bu L., & Liu K. (2014). *Solid modeling strategies-analyzing student choices*. Paper presented at Proceedings of the 121st ASEE annual conference and exposition, Indianapolis, June 15-18.

Ault, H. K., & Fraser A. (2013). *A Comparison of Manual vs. Online Grading for Solid Models*. Paper presented at 120th ASEE Annual Conference & Exposition, Atlanta, GA, June 23–26, 2013, Paper ID #7233.

Barbero, B. R., Pedrosa, C. M. & Samperio, R. Z. (2016). Learning CAD at university through summaries of the rules of design intent. *International Journal of Technology and Design Education*, 1-18. https://doi.org/10.1007/s10798-016-9358-z.

Bodein, Y., Bertrand, R., & Caillaud, E. (2014). Explicit reference modeling methodology in parametric CAD system. *Computers in Industry, 65(1)*, 136-147. https://doi.org/10.1016/j.compind.2013.08.004.

Bouma, W., Fudos, I., Hoffmann, C., Cai, J., & Paige, R. (1995). Geometric constraint solver. *Computer-Aided Design*, *27(6)*, 487-501. https://doi.org/10.1016/0010-4485(94)00013-4.

Briggs, J. C., Hepworth, A. I., Stone, B. R., Cobum, J. Q., Jensen C. G., & Red E. (2015). Integrated, Synchronous Multi-User Design and Analysis. *Journal of Computing and Information Science in Engineering, 15(3),* 031002. https://doi.org/10.1115/1.4029801.

Buckley, J., Seery, N. & Canty, D. (2017). Heuristics and CAD modelling: an examination of student behaviour during problem solving episodes within CAD modelling activities. *International Journal of Technology and Design Education*, 1-18. https://doi.org/10.1007/s10798-017-9423-2.

Camba J. D., & Contero, M. (2015). Assessing the impact of geometric design intent annotations on parametric model alteration activities. *Computers in Industry*, *71*, 35-45. https://doi.org/10.1016/j.compind.2015.03.006.

Camba, J. D., Contero M., & Company, P. (2016). Parametric CAD Modeling: An Analysis of Strategies for Design Reusability. *Computer-Aided Design*, *74*, 18-31. https://doi.org/10.1016/j.cad.2016.01.003.

Camba, J. D., Contero, M. & Company, P. (2017). *CAD reusability and the role of modeling information in the MBE context*. Model-Based Enterprise Summit 2017. National Institute of Standards and Technology (NIST), Gaithersburg, MD, April 3-7. https://www.nist.gov/file/361581 MBE17-020.

Company, P., Contero, M., Otey, J. & Plumed. R. (2015). Approach for developing coordinated rubrics to convey quality criteria in CAD training. *Computer-Aided Design, 63,* 101-117. https://doi.org/10.1016/j.cad.2014.10.00.

Contero, M., Company, P., Vila, C., & Aleixos, N. (2002). Product data quality and collaborative engineering. *IEEE Computer Graphics Applications, 22(3),* 32–42. https://doi.org/10.1109/MCG.2002.999786.

Cheng, Z., & Ma, Y. (2017). A functional feature modeling method. *Advanced Engineering Informatics, 33,* 1–15. https://doi.org/10.1016/j.aei.2017.04.003.

Cheng, Z., Xie, Y., & Ma, Y. (2018). Graph centrality analysis of feature dependencies to unveil modeling intents. *Computer-Aided Design and Applications*, 1-13. https://doi.org/10.1080/16864360.2018.1441236

Company, P., & González-Lluch, C. (2013). *CAD 3D con SolidWorks® Tomo I: Diseño básico*. España: Castellón.

Chester, I. (2007). Teaching for CAD expertise. *International Journal of Technology and Design Education, 17*, 23–35. https://doi.org/10.1007/s10798-006-9015-z.

Dixon, B. M., Dannenhoffer III, J. F. (2014). Geometric Sketch Constraint Solving with User Feedback. *Journal of Aerospace Information Systems, 11(5),* 316-325. https://doi.org/10.2514/1.I010110.

Fudos, I., & Hoffmann, C. M. (1997). A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics, 16(2),* 179-216. https://doi.org/10.1145/248210.248223.

Ge, J. X., Chou, S. C. & Gao, X. S. (1999). Geometric constraint satisfaction using optimization methods. *Computer-Aided Design*, *31(14)*, 867-879. https://doi.org/10.1016/S0010-4485(99)00074-3.

González-Lluch, C., Company, P., Contero, M., Camba, J. D., & Colom, J. (2017a). A Case Study on the Use of Model Quality Testing Tools for the Assessment of MCAD Models and Drawings. *International Journal of Engineering Education*, *33(5)*, 1643–1653.

González-Lluch, C., Company, P., Contero, M., Camba, J. D., & Plumed, R. (2017b). A survey on 3D CAD model quality assurance and testing tools. *Computer-Aided Design, 83,* 64-79. https://doi.org/10.1016/j.cad.2016.10.003.

Hamade, R. F. (2009). Profiling the Desirable CAD Trainee: Technical Background, Personality Attributes, and Learning Preferences. *Journal of Mechanical Design, 131(12),* 121009-121019. https://doi.org/10.1115/1.4000455.

Hekman, K. A., & Gordon, M. T. (2013). *Automated Grading of First Year Student CAD Work*. Paper presented at the 120th ASEE Annual Conference & Exposition 2013, Atlanta, GA, June 23–26. Paper ID #6379.

Hepworth, A., Tew, K., Trent, M., Ricks, R., Jensen, C. G., & Red, E. R. (2014). Model consistency and conflict resolution with data preservation in multi-user computer aided design. *Journal of Computing and Information Science in Engineering*, *14(2)*, 021008. https://doi.org/10.1115/1.4026553.

Jackson, C., & Buxton, M. (2007). *The design reuse benchmark report: seizing the opportunity to shorten product development*. Aberdeen Group, Boston.

Joan-Arinyo, R., Soto-Riera, A., Vila-Marta, S. & Vilaplana-Pastó, J. (2003). *Transforming an under-constrained geometric constraint problem into a well-constrained one*. Paper presented at Proceedings of ACM SM03, Seatle, June 16-20.

Kirstukas, S. J. (2016). *Development and Evaluation of a Computer Program to Assess Student CAD Models*. Paper presented at ASEE Annual Conference & Exposition, New Orleans, June 26.

Kramer, G. (1991). *Using degrees of freedom analysis to solve geometric constraint systems*. Paper presented at Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications 1991, Austin, June 05-07.

Kwon, S., Kim, B. C., Mun D., & Han, S. (2015). Graph-based simplification of feature-based three-dimensional computer-aided design models for preserving connectivity. *Journal of Computing and Information Science in Engineering*, *15(3)*, 031010. https://doi.org/10.1115/1.4030748.

Leea, J. Y., & Kimb, K. (1998). A 2-D geometric constraint solver using DOF-based graph reduction. *Computer-Aided Design*, *30(11)*, 883-896. https://doi.org/10.1016/S0010-4485(98)00045-1.

Mata Burgarolas, N. (1997). Solving incidence and tangency constraints in 2D. Technical Report LSI-97-3R, Departament LiSI, Universitat Politècnica de Catalunya.

Race, P. (2001) The lecturers Toolkit—A practical guide to learning, teaching and assessment. Glasgow, Great Britain.

Petrina, S. (2003). Two Cultures of Technical Courses and Discourses: The Case of Computer Aided Design. *International Journal of Technology and Design Education*, *13*, 47–73.

Red, E., French, D., Jensen, G., Walker, S. S., & Madsen, P. 2013. Emerging design methods and tools in collaborative product development. *Journal of Computing and Information Science in Engineering*, *13(3)*, 031001. https://doi.org/10.1115/1.4023917.

Robertson, B. F., Walther, J., & Radcliffe, D. (2007). Creativity and the Use of CAD Tools: Lessons for Engineering Design Education From Industry. *Journal of Mechanical Design*, *129(7)*, 753-760. http://doi.org/10.1115/1.2722329.

Stone, B., Salmon, J., Eves, K., Killian, M., Wright, L., Oldroyd, J., Gorrell, S., & Richey, M.C. (2017). A Multi-User Computer-Aided Design Competition: Experimental Findings and Analysis of Team-Member Dynamics. *Journal of Computing and Information Science in Engineering*, *17(3)*: 031003. https://doi.org/10.1115/1.4035674.

Summers, J. D., & Shah, J. J. (2010). Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability. *Journal of Mechanical Design*, *132(2)*, 21004-21015. https://doi.org/10.1115/1.4000759.

Szewczyk, J. (2003). Difficulties with the novices' comprehension of the computer-aided design (CAD) interface: Understanding visual representations of CAD tools. *Journal of Engineering Design*, *14(2)*, 169-185. http://doi.org/10.1080/0954482031000091491

Acknowledgements