International Journal of Engineering Education Vol. 33, No. 5, pp. 1643–1653, 2017
Printed in Great Britain

0949-149X/91 $3.00+0.00
© 2017 TEMPUS Publications.

A Case Study on the Use of Model Quality Testing Tools for the Assessment of MCAD Models and Drawings*

CARMEN GONZÁLEZ-LLUCH[1], PEDRO COMPANY[2], MANUEL CONTERO[3],
JORGE D. CAMBA[4] and JESÚS COLOM[1]

[1] Dept. of Mechanical Engineering and Construction, Universitat Jaume I, Av. de Vicent Sos Baynat, s/n 12071 Castellón, Spain.
E-mail: mlluch@uji.es

[2] Institute of New Imaging Technologies, Universitat Jaume I, Av. de Vicent Sos Baynat, s/n 12071 Castellón, Spain.
E-mail: pcompany@uji.es

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

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

In this paper, we build on the idea that specialized instruction improves the overall quality of CAD documents by guiding students into selecting the most suitable modeling strategies and approaches. To this end, automatic assessment tools can be used to detect errors and provide feedback, thus relieving instructors from routine checks and allowing them to address quality errors and modeling aspects of higher semantic level. A representative commercial Model Quality Testing (MQT) solution was selected as a case study to determine whether these tools may become automated assistants for student evaluation and feedback. As a result, a new taxonomy of modeling aspects that can be automatically checked is proposed. We claim that current MQT tools can supplement the learning of quality concepts, but require significant tuning and only provide limited testing and tutoring capabilities. Extending the capabilities of these tools (through macros or dedicated API's), or even developing entirely new MQT tools specifically aimed at instruction purposes, is an essential requirement to develop automated teaching-assistants based on MQT techniques.

Keywords: CAD assessment; CAD model quality; model quality testing

1. Introduction

CAD data quality is a topic of great industrial interest. The standard ISO/PAS 26183:2006 defines product data quality as "a measure of the accuracy and appropriateness of product data, combined with the timeliness with which those data are provided to all the people who need them." In this context, product data includes not only CAD data but also CAM, CAE and other data types managed throughout the product lifecycle. ISO /PAS 26183:2006 and similar standards such as the German version VDA 4955 provide a set of data quality criteria mainly centered on CAD. There are several stand-alone product data quality (PDQ) checkers in the market used for the assessment of the quality criteria specified by PDQ standards. Some commercial CAD systems also provide dedicated modules to apply in-house design standards, and then check CAD documents against them.

To create high quality CAD models, all the stakeholders involved in the product development process must be knowledgeable in CAD quality and aware of the applied criteria. Although current PDQ standards mainly focus on the mathematical and topological correctness of CAD models, aspects of higher conceptual level (e.g., CAD model reusability) are starting to be considered [1]. However, only a limited number of experiences in the engineering education field have connected CAD teaching methodologies to the PDQ world and taken advantage of PDQ tools for the assessment of CAD data created by students during their training process. In terms of the procedural aspects of CAD model creation, Leith et al. [2] showed that students tend to select the easiest modeling methods based on how they visualize the part, ignoring alternative strategies. From an educational standpoint, researchers agree that strategic knowledge should be a fundamental component of solid modeling instruction [3, 4]. In this regard, some authors have focused on developing coordinated and concise rubrics to enforce effective modeling practices during the CAD training of novice product designers [5]. Research shows that prompt feedback is essential for students to improve their modeling strategies [6]. Furthermore, feedback should be continuous in a formative sense [7].

When analyzing work by engineering and technical graphics educators published in recent years, three main strategies can be identified that improve CAD teaching and incorporate specific PDQ concepts in the learning activities:

1. Development of rubrics for the assessment of CAD models [5, 8–10].

* Accepted 22 April 2017.

2. Creation of activities or exercises to increase students' awareness of the methodological aspect of CAD model construction [11– 14].
3. Use of automated electronic tools to homogenize and improve CAD grading [6, 15–17].

Regarding the first approach, various authors have introduced metrics to support the assessment of trainees [5, 8–10]. For example, Ault et al. [8] proposed metrics for evaluating solid models and comparing modeling strategies. In their work, the authors suggest that students' training should include strategic discussions regarding the various uses of part models and alternative modeling methods to raise students' awareness about the importance of model planning. Other researchers have developed computer-assisted rubrics to convey CAD quality concepts and provide feedback to students [9, 10]. The authors concluded that computer-based rubrics offer advantages over their paper-based counterparts, as they facilitate the creation of adaptable and adaptive e-learning systems.

The second strategy is based on the preparation of educational materials aimed at improving modeling skills. Branoff et al. [11] presented three courses where students reverse-engineered existing designs, modeled standard parts from catalogs, and solved numerous design problems. Feedback from instructors was provided as written comments. In a subsequent study, a variety of exercises was introduced, ranging from the uses of constraint-based geometry to the development of proper solid modeling strategies to support downstream applications [12]. Students received feedback throughout the course and a rubric, which included grading criteria for sketches, solid models, drawings and assemblies, was used for assessing their final projects. In a more recent study [13], a methodology was developed to support student's learning. Students evaluated their own work by measuring distances and the surface area of the models they created. Dimensions from master models created by the instructors were used as baseline. Barbero et al. [14] approached 3D model reuse by examining how CAD is learnt and also considering the convenience of introducing design intent through proper modeling strategies from the moment students start to learn CAD. The authors presented practical exercises that emphasized the importance of proper design intent communication (e.g. describing the thinking process involved in modeling a part, or introducing new concepts and rules through selected exercises).

Additional studies have focused on automated tools to provide immediate feedback to students, which can save time [15] and provide an objective assessment [6]. Ault et al. [16] created an automatic grading system, where a student's model was compared to a template provided by the instructor. This template checked procedural and strategic knowledge on part models, which reduced grading time. Automatic grading was used to assess placement of features, feature order, and use of constraints to capture design intent. In a similar system developed by Hekman and Gordon [17], students receive a list of discrepancies and pictures contrasting their solutions with the answer key. An analogous computer program was created by Kirstukas [6] to compare NX solid models created by the instructor with the students' models, provide a score, and offer feedback to students.

A review of the available literature reveals a growing concern about how to improve formative assessment and introduce quality concepts in CAD instruction. Particularly, there is a need for automated tools that link quality criteria, provide feedback, and facilitate student assessment. In academic research contexts, only a few ad-hoc applications have been developed for such purposes.

The work presented in this paper contributes to the improvement of CAD instruction by analyzing off-the-shelf software tools to support formative assessment and CAD quality concepts in engineering education arenas. As a first step, a new mapping of quality criteria is presented to ensure high quality models for parts, assemblies, and drawings. A representative commercial Model Quality Testing application—SolidWorks Design Checker® (SWDC)—is examined to determine whether these types of tools offer automated solutions to evaluate students' work, provide feedback, and ensure high quality modeling practices during the CAD training process.

The paper is organized as follows: In section 2, model quality testing tools are described. Next, the particular tool analyzed in this study is introduced (section 3). In section 4, testable properties are classified and the capability of the SWDC to automatically detect quality errors is validated through examples and explanations. A discussion is presented in section 5. Finally, section 6 provides conclusions and a brief reference to future works.

2. Model quality testing (MQT) tools

Product data includes a variety of digital documents that are managed during the product lifecycle. One of the most important elements in a digital product development process is the 3D geometrical representation of the product or system (i.e. the CAD model), which, in the Model-Based Enterprise paradigm, is considered the primary view that feeds secondary views linked to downstream CAx applications. The tools used to analyze CAD models are often referred to as Model Quality Testing (MQT) tools. These tools (1) are usually interactive, (2) require complex tuning, and (3) work mainly with low semantic level errors [18].

CAD MQT is an activity that involves identifying "dirty clean-up problems" in a master CAD model [18]. Although digital 3D part models are generally the main type of document produced by history-based parametric mechanical CAD applications (digital 3D assemblies and 2D drawings are other typical outputs), the term *model* is sometimes used as a generic word that encompasses all types of documents produced by these CAD applications. All three types of documents have related quality issues.

Model quality technology enables designers to identify, locate, and even resolve model integrity problems before the file leaves the CAD system [19]. Some MQTs simply detect failures, while others also repair the document. Some MQT tools are embedded or linked to particular CAD systems, while others are independent. This results in different market segments for MQT tools.

In this work, a commercial MQT solution is examined. SolidWorks Design Checker® (SWDC) is an add-in distributed with the professional and premium versions of SolidWorks®, which verifies and repairs drawings, models, and assemblies [20]. This tool was selected as a convenient representative since it supports three types of documents (models, assemblies and drawings) and belongs to the most affordable segment of MQT tools, as described by [18]. Higher-end MQT tools such as CADfix® or 3DTransVidia® are not embedded or linked, and support a larger variety of CAD formats. However, both their price per license and annual maintenance may increase by a factor of ten, thus making them unaffordable for many Small and Medium Enterprises and for teaching higher level CAD quality concepts [18].

The goal of the study is to determine if this representative MQT tool can automate the detection of CAD model quality errors, and thus become an automated assistant for evaluation and feedback that can monitor and guide students in selecting high quality modeling strategies. The experimental work was aimed at defining a new taxonomy of modeling aspects that can be automatically checked. To this end, the quality criteria described by Company et al. [5] was compared against the parameters implemented in SWDC.

According to Company et al. [5], the first three dimensions of quality (validity, completeness and consistency) are dichotomist, and thus should be easily implementable by MQT tools. Our mapping demonstrates that this has been only partially accomplished. Furthermore, we argue that the quantitative evaluation can sometimes be improved by a complementary qualitative evaluation. For instance, if we assume that the model is the primary view (while the drawing is a secondary or derived view), then exporting dimensions from the model to the drawing is good practice, while manually adding dimensions to the drawing is bad practice. Therefore, we can use the testable properties of drawings to determine whether or not all their dimensions were *imported* from the model. The model would be classified as good quality if all dimensions were imported, and the quality metric would linearly decrease to zero if no dimensions were imported. To a certain extent, this metric is subjective: some non-imported dimensions may be good practice (e.g. they result from cosmetic changes that attempt to improve the readability of the drawing). Alternatively, a single non-imported dimension may also imply a catastrophic dimension transfer that can drastically modify the original design intent of the model.

In general, qualitative evaluations are desirable as a complement to quantitative and automatic assessments. To facilitate communication, the information should be displayed upon request and in a manner that clearly distinguishes between presumably good and bad practices. For instance, a color scheme can be used so imported and manually added dimensions are displayed in different colors. This simple strategy allows the instructor to do a quick visual inspection and complement the automatic-evaluation with a qualitative assessment on how critical the non-imported dimensions really are.

3. Mapping tested properties to quality dimensions

SWDC verifies design elements such as dimensioning standards, fonts, materials, and sketches to ensure that SolidWorks® documents meet predefined design criteria [20]. SWDC integrates the following four modules [20]: *Build Checks, Check Active Document*, *Check Against Existing File*, and *Learn Checks Wizard*.

The requirements for evaluation can be set via the *Build Checks* module. *Check Against Existing File* is used to validate the active document against design checks created from existing files. The *Learn Checks Wizard* is used to retrieve design checks from an existing SolidWorks® part, assembly or drawing document based on specific attributes. To validate a document, the interactive tool *Check Active Document* [20] can be used, where items that fail to pass the check can be handled and corrected individually or as a group.

Prior to testing any document, the *requirements*

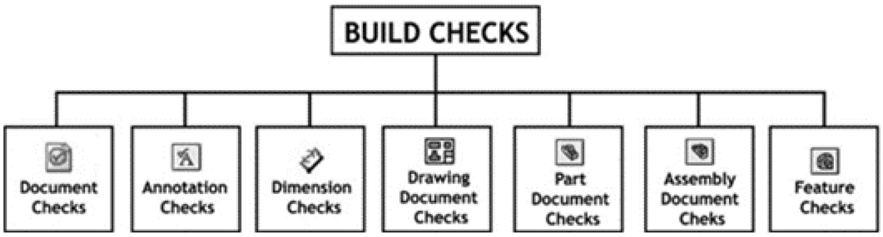


Fig. 1. Seven categories of requirements that *Build Check* can verify.

(verifications that are predefined in the application) must be included in a checklist called *Standards File*. *Build Checks* is an interactive module to define lists of requirements based on specific verifications or checks, which are classified in seven different categories: document, annotations, dimension, drawing document, part document, assembly document and feature (Fig. 1).

Although the seven categories of requirements are not directly connected to the three types of documents (models, assemblies and drawings), our re-mapping of the requirements shows that they are at least aligned with them (Fig. 2). Validations that affect models are separated in two groups: *Part Document Checks,* aimed at guaranteeing the correct definition of the material; and *Feature Checks,*

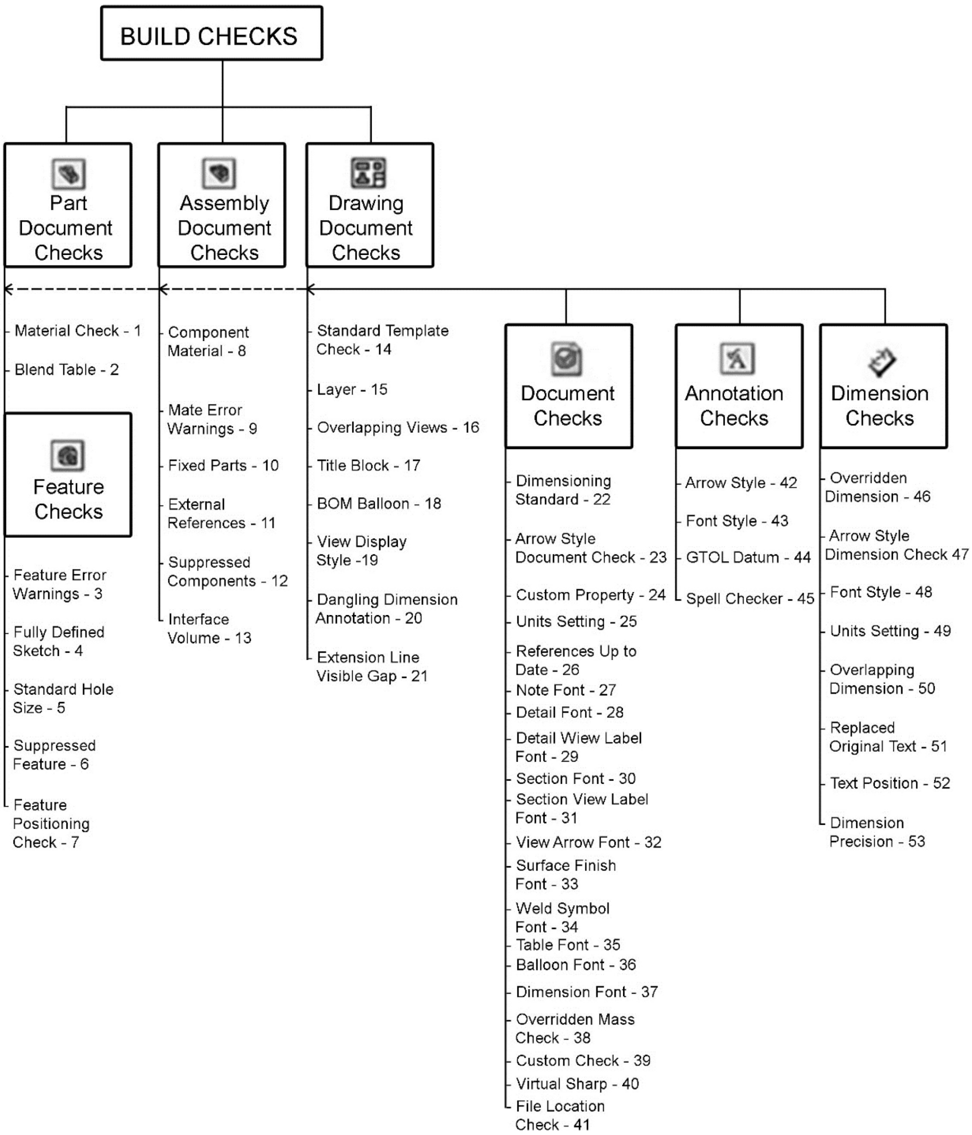


Fig. 2. Build Check requirements mapped according to the type of document.

which handles the history-based modeling process. Document, annotation, and dimension categories are mostly related to drawings due to the fact that CAD practices still favor these documents, as standards for the representation of dimensions in models and assemblies (as defined in Digital Product Definition Data Practices like [21] or [22] are not yet widespread. Nevertheless, in SWDC the categories are transversal to all three document types, although some exceptions such as criteria 35 and 36 only apply to drawings.

An existing file may be used as a template to create a check-list "on the fly" and set the evaluation requirements. The *Check Against Existing File* module compares the settings of a given template with those of the active document. In addition, the *Learn Checks Wizard* module can be used to configure the suitable requirements as a new check-list by retrieving design checks based on attributes from an existing part, assembly, or drawing document.

The fact that two different modules are provided to set requirements from templates suggests that the MQT tool is designed to maintain document consistency within large enterprises. In other words, the tool is designed to ensure that all documents in a company follow the same general morphology and syntax. However, MCAD model quality also involves semantic correctness, as emphasized by Pratt [23]. By further developing this idea, authors Contero et al. [1] defined three levels of quality to classify CAD models: morphologic, which relates to the geometric and topological correctness of the CAD model; syntactic, which assesses the proper use of modeling conventions, and semantic/pragmatic, which focuses on the CAD model's ability for modification and reuse. Although semantic/pragmatic level quality aspects seem to be absent from the SWDC tool (see Fig. 2), a refined classification was developed to better determine the extent up to which the tool can manage the overall quality of a CAD document.

4. Mapping requirements to quality dimensions

To determine the suitability of MQT tools to automatically assess the quality of MCAD documents, the six quality dimensions defined by Company et al. [5] were compared against the 53 Build Check requirements of the SWDC, as listed in Fig. 2. Since no direct mapping is possible between the six main criteria (valid, complete, consistent, concise, simple, and conveys design intent) and the full list of SWDC requirements, the expanded version of the criteria was used. The original Level of Detail (LoD) was provided to introduce quality concepts in a *bottom-up* approach (the most abstract concepts must be presented only after the detailed quality issues are fully understood). To this end, we present three tables that illustrate the mappings between quality criteria and SWDC requirements.

The same approach is applied to models, assemblies, and drawings. The first column of each table shows the expanded list of quality criteria for models derived from the one defined by Company et al. [5]. The second column lists the numbers of the corresponding SWDC requirements. For mappings considered dichotomist, no details are provided. Further explanations are given for mappings considered unclear or incomplete.

4.1 Mapping of quality criteria for models

Our proposal for the mapping of the expanded list of quality criteria for models (derived from the one defined by Company et al. [5]) and the Build Check requirements (Fig. 2) is shown in Table 1.

Criterion 1.3 is covered by SWDC, specifically by requirements 3 and 9. However, SWDC does not change the model automatically. A warning message is the only feedback. Moreover, SWDC is to some extent redundant, since some of the errors can also be identified in the model tree (e.g., error messages and warnings attached to the corresponding modeling operations). In all cases, the errors must be corrected manually by the user.

Criteria 3.1 and 4.1 are mapped to requirement 4. This detection is also redundant, since it is possible to detect whether a profile is fully constrained as it is being created by observing its line color, which can be configured from the program's configuration menu. This information is also available after the profile has been closed, in the form of a minus sign (–) preceding the profile name when the sketch is not fully constrained.

The mapping between criterion 4.1 and the Build Check requirement 4 is unclear, as the behavior of the SWDC is different for repetitive and fragmented constraints. For instance, if a profile contains repetitive numeric constraints (dimensions), the user is warned by the program as soon as the constraint is added. At this point, the user is forced to solve the error by making the dimensions driven, driving, or simply removing them. SWDC cannot autocorrect the problem or launch its interactive dialog, which forces the user to manually edit the profile.

Both the CAD application and SWDC fail to detect constraints that are repetitive but not incompatible. For example, the profile shown in Fig. 3 has repetitive geometric constraints (horizontal and parallel constraints are simultaneously applied to the horizontal sides of the rectangle). However, these repetitive constraints are not detected by the CAD application or the SWDC. An example of

Table 1. Quality criteria proposal for models

Quality criteria	**Build Check requirement**
1. The model is valid	
1.1 The file of the model can be located and opens in neutral state	
1.2 Model is compatible with the CAD system	
1.3 Model tree is free from error messages	3, 9
2. The model is complete	
2.1 The model replicates the shape and the size of the part	
3. The model is consistent	
3.1 Profiles are free from duplicated and segmented lines, and are fully constrained	4
3.2 The model is aligned and oriented relative to global reference system (Its main views align with Front, Top and Side Planes)	
3.3 Model uses suitable datums (that define a scaffold that helps build and edit the model)	
3.4 The model tree is free from unnecessary dependencies	
4. The model is concise	
4.1 Profiles are concise (free from repetitive or fragmented constraints)	4
4.2 The model is free from repetitive or fragmented modeling operations	
4.3 The model is free from repetitive, fragmented or unused datums	
4.4 Replication operations (translate-and-repeat, rotate-and-repeat and symmetry) are used whenever possible	
5. The model is clear	
5.1 Modeling operations are labeled in the modeling tree to emphasize their function, instead of how they were built	
5.2 Related modeling operations are grouped in the model tree to emphasize parent-child relationships	
5.3 The most compatible modeling operations are always used	
5.4 The most standard modeling operations are always used	5
5.5 Dimensions used in profiles are readable and easily editable (do not overlap, have a suitable size, etc.)	
5.6 Properties of dimensions and leader lines from the profiles are suitable, according to ISO, UNE standards (primary precision, size dimensions, color lines, etc.)	
6. The model conveys design intent	
6.1 The model tree is like a "script" that describes the elements that constitute the part and their functionality	
6.2 The modeling sequence moves from primary to secondary elements	
6.3 The model was created in a manner that prevents the loss of design dimensions (there are no dimension transfers or conversion of dimensions into geometrical constraints)	
6.4 The model was created in a manner that prevents the loss of symmetries and replication patterns	
6.5 User defined constraints allow the model to be both flexible (allows many design changes) and robust (prevents catastrophic changes)	

fragmented constraints not detected by SWDC is shown in Fig. 4, where a perpendicular constraint coexists with vertical and horizontal constraints.

Finally, the mapping between criterion 5.4 and the Build Check requirement 5 is also unclear. SWDC only identifies standard holes and other standard modeling operations such as Fillet and Rib are not covered. Furthermore, SWDC does not autocorrect or allows the user to interactively repair the selected entities.

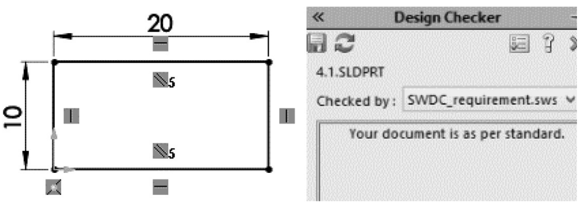
We were unable to determine a valid mapping for criterion 5.6, although requirement 23 (Arrow style Document Check) certainly allows the selection of different arrow styles based on size, shape, etc. However, passing the validation does not necessarily mean that the properties listed meet the standard. It only indicates that the document criteria match the validation.

In general, limited support is provided by the Build Check requirements for the proposed quality criteria (1–5). Specifically, only 4 out of the 23 quality criteria listed in Table 1 are covered by SWDC. Some Quality Criteria such as design intent are not available and other criteria cannot be checked while the model is being created.

4.2 Mapping of quality criteria for assemblies

Our proposal for the mapping of the expanded list of quality criteria for assemblies (derived from the one defined by Company et al. [5]) and the Build Check requirements (Fig. 2) is shown in Table 2.

Build Check requirement 9 is mapped to Quality



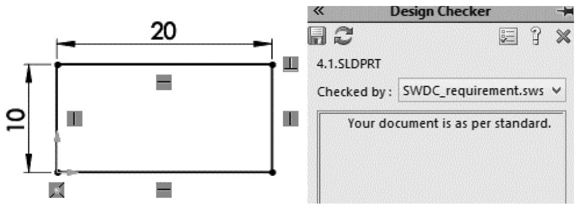
Design Checker

4.1.SLDPRT

Checked by: SWDC_requirement.sws

Your document is as per standard.

Fig. 3. Sketch with repetitive geometric constraints.



Design Checker

4.1.SLDPRT

Checked by: SWDC_requirement.sws

Your document is as per standard.

Fig. 4. Warning displayed when a sketch contains fragmented constrains.

criterion 1.2, as it identifies mates that have rebuild errors or warnings. Criterion 2.1 is currently unreachable for SWDC, as it does not depend on neutral and generic settings. This is an example of a quantitative evaluation that is only possible when (1) we can feed the MQT tool with a finished model that can be used as "ground truth", and (2) the MQT tool is able to compare the degree of correspondence between the final shape and size of the models. This criterion must be, at present, qualitatively evaluated by an expert.

The mapping between criterion 2.4 and the Build Check requirement 13 is implemented by SWDC via the interference volume of assembly components. A comparison operator is used to select the desired interferences. Valid operators include "range of values," "list of values," or "exclusion" (which acts like the "not" operator). It is also possible to calculate interferences between components from the CAD application as the assembly is being created.

The mapping between criteria 3.2 and 6.1, and the Build Check requirement 10 is unclear, since the requirement only determines the number of fixed components that belong to the assembly. Furthermore, criterion 6.1 is only partially checked.

Finally, the built-in capabilities of the CAD

Table 2. Quality criteria proposal for assemblies

Quality criteria	**Build Check requirement**
1. The assembly is valid	
1.1 The assembly file can be located and opens in neutral state	
1.2 The assembly can be used. The file is free of errors	9
1.3 All components linked to the assembly can be accessed (including parts, sub-assemblies and library parts), even when libraries are not available, or when software compatibility issues exist between versions	
2. The assembly is complete	
2.1 The assembly includes all and only the necessary parts and sub-assemblies	
2.2 Standard library parts are included when required, which are suitably instantiated from the library	
2.3 Relative locations among components (parts, sub-assemblies or library parts) match their functional positions	
2.4 Components (parts, sub-assemblies or library parts) are free of unwanted interferences	13
3. The assembly is consistent	
3.1 The base component is correctly assigned and linked to the global reference system	
3.2 All components are suitably assembled by way of mate conditions (assembly allows valid movement and prevents undesired movement)	10
4. The assembly is concise	
4.1 Replication operations (translate-and-repeat, rotate-and-repeat and symmetry) are used whenever possible	
4.2 The parent/child relations in the assembly tree are free of unnecessary dependencies	
5. The assembly is clear	
5.1 All components, sub-assemblies, and mate constraints are properly labeled and organized in groups	
5.2 The assembly uses compatible and standard mates	
6. The assembly conveys design intent	
6.1 The assembly tree replicates the assembly/disassembly sequence	10
6.2 Sub-assemblies encapsulate clearly perceived functions	
6.3 Mate constraints in sub-assemblies allow proper motion (they have been unfrozen)	
6.4 Mating features provided to ease assembly (if any) are mostly used for mating	
6.5 Parts that belong to modular families (if any) can be easily and safely replaced	

application allow users to review the assembly tree and obtain information that is similar to that provided by SWDC. In particular, two symbols are automatically added to the part name in the assembly tree when a component is fixed (f) or not fully constrained (–). Overall, only 4 out of 18 criteria are covered by Build Check requirement, although some are also available in the CAD application during the assembly modeling process.

4.3 Mapping of quality criteria for drawings

Our proposal for the mapping of the expanded list of quality criteria for drawings (derived from the one defined by Company et al. [5]) and the Build Check requirements (Fig. 2) is shown in Table 3.

Criterion 1.3 is mapped to Build Check requirement 3. However, only errors at the part level are identified (i.e., SWDC will not identify specific errors in the assembly tree, but errors within the model trees of individual components in the assembly).

Criteria 2.4 and 2.5 are mapped to Build Check requirements 18 and 20. The Bill of Materials (BOM) balloon requirement (18) identifies the item numbers in the BOM that have missing balloons. Requirement 20 checks for dimensions or annotations that no longer have a reference. Criterion 2.6 is mapped to Build Check requirement number 16, where drawing views with overlapping boundaries are identified. Although SWDC does

Table 3. Quality criteria proposal for drawings

Quality criteria	Build Check requirement
1. The drawing is valid	
1.1 The file has the expected name and is in the expected place (folder or web page)	
1.2 The file is free of errors	
1.3 Drawing tree is free from error messages	3
2. The drawing is complete	
2.1 The views show all the external elements of the object	
2.2 The section views show all the internal elements of the object	
2.3 All dimensions are shown	
2.4 Auxiliary lines (axes of symmetry, etc..) and annotations complement the object representation	18, 20
2.5 The drawing is free of redundant information in views, dimensions, and annotations	18, 20
2.6 Arrangement of views and dimensions facilitates drawing readability	16
2.7 The drawing is free of views, dimensions, and annotations that obstruct visibility	
3. The drawing is consistent	
3.1 The views, sections and dimensions are imported from the model	
3.2 The drawing is free of unnecessary "cosmetic items"	
4. The drawing is presented correctly	
4.1 The views, annotations, and parts lists follow ISO or UNE standards	22
4.2 All document properties are defined (dimensions, arrows, lines format, etc. . .)	23, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 42, 43, 47, 48, 49, 53
4.3 Standard sheet sizes and format are used, when appropriate	14
4.4 Dimensions are properly placed and clearly arranged within the views (dimension length, dimension lines, overlay, etc.)	21, 50, 52
4.5 Borders and title block follow standard guidelines	17
4.6 The drawing scale is appropriate to facilitate viewing	
4.7 The author and the owner are clearly identified	17
4.8 The representation system is clearly indicated	
4.9 Scales are clearly stated and correspond to the drawing	
4.10 The drawing units are clearly indicated	
4.11 Mark arrangement facilitates information search	
5. The drawing conveys design intent	
5.1 The object orientation facilitates the understanding of its functionality	
5.2 The layout of views and dimensions help highlight symmetries and patterns	
5.3 The drawing sequence prevents the loss of design dimensions (there are no transfers of dimensions)	
5.4 The drawing has been created in a manner that prevents the loss of symmetries and patterns	
5.5 The order of the marks suggests a realistic assembly sequence	

not distinguish between views, sections, and dimensions (imported or added manually), a color scheme can be set in the CAD application so driven dimensions (those added by the user) are displayed in a different color.

The mapping between criterion 4.1 and the Build Check requirement 22 is unclear, as dimensions in the graphics area are ignored by SWDC. The tool only evaluates whether the dimensioning scheme follows the standard specified in the document settings.

For the remaining mappings, general criterion 4 (the drawing is presented correctly) is significantly covered by the SWDC, particularly for sub-criterion 4.2, as shown in Table 3.

Criterion 4.3 can be validated in SWDC through a standard template (configured with the desired projection type) selected by the user.

5. Discussion

The initial tuning required to use the SWDC tool is typically done with the assistance of a specific module (*Build Checks*), which facilitates the archival and long term usage of the settings and tests. Identified errors are reported as a list that can be used for individual inspection, comparative analysis (e.g. reporting the incidence of the different types of errors), and archival purposes (such as error tracking in large distributed enterprises where strict communication protocols are enforced to prevent misunderstandings that can cause catastrophic failures). In general, the structure of the application is designed for intensive use and to maintain consistency across vast amounts of documents. Most criteria implemented in SWDC are aimed at verifying settings, and thus intended to ensure the semantic correctness of the CAD model.

The mapping of Build Check requirement against the CAD quality criteria (see Tables 1 to 3) revealed a poor coverage of high semantic quality criteria, which translates into unsuitability as an instructional tool. Although certainly important, the criteria covered by the tool only represent a small fraction of the overall quality dimension of consistency. Furthermore, a significant portion of the quality failures that can be tested by SWDC can also be tested (and sometimes corrected) interactively by the CAD application during the modeling stage.

In addition, a common perception among the authors while preparing the experiments was that the tuning procedure is time consuming and the testing and tutoring capabilities, limited, which can be discouraging for many instructors. In fact, many may find it more productive to manually grade exercises than to fine tune a variety of templates that can reasonably accommodate the content that is typically included in a CAD course. Consequently, we suggest that future alternatives for MQT should include: (1) extended capabilities in the form of suitable macros or dedicated API's, or (2) new MQT tools specifically aimed at instruction and tutoring. Furthermore, the functionality of these MQT tools should go beyond automatically repairing errors by offering mechanisms to automatically annotate models and provide an informative output about the modeling process that contributes to students' learning.

Alternative MQT tools may also be useful to Small and Medium Enterprises (SME) interested in improving the quality of their CAD documents. In this case, low cost (both in terms of acquisition and maintenance), ease of use, and customization are a must [18].

Finally, our vision is that CAD quality is not limited to MQT tools, as it depends on three actions: (1) maximize the quality of CAD models while modeling is in progress, (2) use MQT tools to analyze the models and repair quality failures, and (3) convey high-level CAD quality information through intelligent annotations embedded in the models. To this end, we intend to convert e-rubric platforms (such as the one described in [10]) into more comprehensive educational resources by enriching them with interactive quality-oriented CAD training materials. Additionally, as model complexity and volume of information increase, more efficient and user-friendly methods to interrogate annotations are needed. Further studies are required to: (1) define a reference frame to delimit the meaning and reach of the annotations (aimed at standardizing them), (2) define the expected behavior of textual annotations under queries, and (3) allow the robust exchange of annotations between CAD applications.

It is in this context where we consider that MQT remains an open practical problem, as new quantitative metrics are required to measure higher semantic level quality aspects. We propose to further define and test metrics to monitor the presence or absence of high semantic-level quality criteria in CAD models. For instance, it has been stated [5] that profiles must be robust (changes do not produce unexpected failures) and flexible (allow for many changes). Robustness is usually measured by the amount of geometrical and dimensional constraints, and over-constrained profiles are automatically detected by most CAD applications. But a clear metric for flexibility is still missing. If we can experimentally validate the hypothesis that the flexibility of the profiles depends not on the amount but on the semantic level of the constraints, valid metrics for flexibility may follow. For instance, in our vision, constraints that link each element of a drawing to the reference system belong to a lower semantic level than those that create links between drawing elements. Thus, detecting excessive use of poor "fix" relations that lock point coordinates is an example of the type of high semantic quality tests that are not supported by current CAD quality testers.

6. Conclusions

In this paper, SolidWorks Design Checker® (SWDC) has been studied as a representative case study of current commercial Model Quality Testing tools (MQT). SWDC can identify and sometimes repair data errors that could affect the simplification, interoperability, and reusability of CAD models.

The primary goal of this work was to determine the usefulness of this MQT tool as an assessment mechanism both for instructors and for self-evaluation. By mapping the Build Check requirement of SWDC against the CAD quality criteria available in the literature, two main conclusions can be drawn: (1) SWDC only covers lowest semantic level quality criteria, and (2) SWDC is designed for intensive use to maintain consistency across documents.

Two additional observations are included: (1) SWDC will repair certain errors, but others (not all) will only be identified, and (2) SWDC partially overlaps with the built-in checking capabilities of the CAD application, which can sometimes perform better that the MQT (from the point of view of an educational mechanism aimed at improving CAD modeling quality).

From an educational standpoint, current MQT tools like SWDC are unsuitable to teach CAD quality concepts, as they require significant tuning to provide, at best, limited testing and tutoring capabilities. This paper sheds light on the idea that, although MQT is considered solved by a number of scholars, it remains an open practical problem, as new quantitative metrics require the design of new application programming interfaces that transform current MQT tools into mechanisms to assess higher semantic level quality aspects.

Acknowledgements—This work was partially funded by the Spanish "Ministerio de Economía y Competitividad; Proyectos I+D+I Convocatoria RETOS 2013," Project TIN2013-46036-C3-1-R (ANNOTA: la aplicación de las anotaciones 3D en el contexto industrial de la empresa basada en modelos (model based enterprise) y la formación técnica).

References

1. M. Contero, P. Company, C. Vila and N. Aleixos, Product Data Quality and Collaborative Engineering, *IEEE Computer Graphics and Applications*, **22**(3), 2002, pp. 32–42.
2. J. Leith and H. K. Ault, Using Think-Aloud Exercises to Reveal Students' Solid Modeling Strategies, *Proc. Midyear Conf. Eng. Design Graphics Division of the Am. Soc. for Eng. Education*, Worcester, October 20–22, 2013, pp. 28–33.
3. G. Menary and T. Robinson, Novel approaches for teaching and assessing CAD. *International Conference on Engineering Education*, Belfast, N. Ireland, August 21–26, 2011, pp. 21–26.
4. A. Rynne and W. Gaughran, Cognitive Modeling Strategies for Optimum Design Intent in Parametric Modeling, *Computers in Education Journal*, **18**(1), 2015, pp. 55–68.
5. P. Company, M. Contero, J. Otey and R. Plumed, Approach for developing coordinated rubrics to convey quality criteria in CAD training, *Computer-Aided Design*, **63**, 2015, pp. 101–117.
6. S. J. Kirstukas, Development and Evaluation of a Computer Program to Assess Student CAD Models, *ASEE Annual Conference* & Exposition, New Orleans, Louisiana, June 26–29, 2016, Paper ID #15834.
7. P. Race, *The lecturers Toolkit—A practical guide to learning, teaching and assessment*, Routledge-Falmer, Glasgow, Great Britain, 2001, pp. 1–279.
8. H. K. Ault, L. Bru and K. Liu. Solid Modeling Strategies—Analyzing Student Choices, *Proceedings of the 121st ASEE Annual Conference and Exposition*. Indianapolis, June 15–18, 2014, Paper ID #9242.
9. P. Company, J. Otey, M. Contero, M. J. Agost and A. Almiñana, Implementation of Adaptable Rubrics for CAD Model Quality Formative Assessment, *International Journal of Engineering Education*, **32**(2 A), 2016, pp. 749–761.
10. P. Company, M. Contero, J. Otey, J. D. Camba, M. J. Agost and D. Pérez-López, Web-based system for adaptable rubrics: case study on CAD assessment, *Educational Technology & Society*, **20**(3), 2017, pp. 24–41.
11. T. J. Branoff, E. N. Wiebe and N. W. Hartman, Integrating constraint-based CAD into an introductory engineering graphics course: Activities and grading strategies, *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*, 2003, Session 1338, pp. 1–10.
12. T. J. Branoff. Constraint-based modeling in the engineering graphics curriculum: laboratory activities and evaluation strategies, *Proc. Midyear Conf. Eng. Design Graphics Division of the Am. Soc. for Eng. Education*, Williamsburg, VA, 2004, pp. 132–138.
13. K. L. Devine and M. A. Laingen, Assessing Design Intent in an Introductory-Level Engineering Graphics Course, *68th Mid-Year Conference. ASEE Engineering Design Graphics Division*, Worcester, October 20–22, 2013, pp. 59–63.
14. B. R. Barbero, C. M. Pedrosa and R. Z. Samperio, Learning CAD at university through summaries of the rules of design intent, *International Journal of Technology and Design Education*, 2016, pp. 1–18.
15. A. E. Tshibalo, The Potential impact of computer-aided assessment technology in higher education. *South African Journal of Higher. Education*, **22**(6), 2007, pp. 686–695.
16. H. K. Ault and A. Fraser, A Comparison of Manual vs. Online Grading for Solid Models, *120th ASEE Annual Conference & Exposition*, Atlanta, GA, June 23–26, 2013, Paper ID #7233.
17. K. A. Hekman and M. T. Gordon, Automated Grading of First Year Student CAD Work, *120th ASEE Annual Conference & Exposition*, Atlanta, GA, June 23–26, 2013, Paper ID #6379.
18. C. González-Lluch, P. Company, M. Contero, J. D. Camba and R. Plumed, A Survey on 3D CAD Model Quality Assurance and Testing Tools, *Computer-Aided Design*, **83**, 2017, pp. 64–79.
19. D. McKenney, Model Quality. The Key to CAD/CAM/CAE Interoperability, *1998 Americas Users'* Conference, 1998, pp. 1–12.
20. Dassault Systemes. Solidworks Help. Welcome to Solidworks Design Checker, http://help.solidworks.com/2015/english/solidworks/solidworks_design_checker/c_welcome_design_checker.htm, Acceded 11 October 2016.
21. American Society of Mechanical Engineers. ASME Y14.41-2012, *Digital Product Definition Data Practices*, 2012.

22. International Standard Organization. ISO 16792-2015, *Technical product documentation—Digital product definition data practices*, 2015.

23. M. J. Pratt, Geometric Modelling: Lessons learned from the 'Step' Standard. IFIP, *The International Federation for Information Processing*, **75**, 2001, pp. 130–146.

Carmen González-Lluch is an associate professor in the area of Graphical Expression in Engineering (Department of Mechanical Engineering and Construction) at Universitat Jaume I (UJI), Castellón, Spain, where she teaches CAD I and is responsible for the CAD II course in the Industrial Design and Product Development program. She also teaches Engineering Graphics in the Mechanical, Electrical, Industrial Technologies and Chemical programs. She is a member of the research groups *Computer-Aided Design* and *Biomechanics and Ergonomics*. Her main research interests focus on CAD quality modeling and the development of methods and tools for CAD quality assurance.

Pedro Company earned his PhD in Mechanical Engineering from Polytechnic University of Valencia (UPV) in 1989. He joined Jaume I University (UJI) in 1994 and since 1996 he is a Full professor of Engineering Graphics. Dr. Company teaches Engineering Graphics mainly in Industrial Engineering, but also in Mechanical Engineering and Industrial Design. His research interests are centered on Computer Aided Design, Computer Graphics and Sketch-Based Modeling. Currently, he is contributing to the development and applicability of a new sketch-based modeling interface (http://www.regeo.uji.es/), as well as the development of methods and tools for CAD quality assurance.

Manuel Contero is a full professor of Engineering Graphics and CAD in the Graphic Engineering Department at the Universidad Politécnica de Valencia, Spain (UPV). He earned an MS degree in Electrical Engineering in 1990, and a PhD in Industrial Engineering in 1995, both from UPV. In 1993, he joined Universitat Jaume I of Castellón, Spain (UJI) as an assistant professor and was promoted to associate professor in 1997. He returned to UPV in 2000 and was appointed full professor in 2008. His research interests focus on sketch-based modeling, collaborative engineering, human computer interaction, development of spatial abilities, and technology-enhanced learning.

Jorge D. Camba is an Assistant Professor of Industrial Design in the Gerald D. Hines College of Architecture and Design at the University of Houston. Prior to joining UH, he taught in the College of Engineering at Texas A&M University for eight years. Dr. Camba comes from a multidisciplinary background with degrees in Computer Science (BS, MS), Digital Media (MS) and Design and Manufacturing (PhD). His research interests include Model-Based Engineering, CAD quality and complexity, computer graphics, and human-computer interaction.

Jesús Colom was a senior undergraduate student in the Industrial Engineering program at Universidad Jaume I, UJI by the time the analysis of the MQT tool was completed.