Improving the Learning Experience within MCAD Education: A Tool for Students to Assist in Self-Assessment during Modeling Exercises

Ferruccio Mandorli[1] and Harald E. Otto[2]

[1]Polytechnic University of Marche, f.mandorli@univpm.it
[2]Polytechnic University of Marche, h.e.otto@univpm.it

Corresponding author: Ferruccio Mandorli, f.mandorli@univpm.it

Abstract. Automation of CAD model grading is obviously capable of considerably reducing the time required for analyzing and assessing CAD models created by students, though the type and complexity of CAD model that can be analyzed, and the quality of the feedback that is generated, are still quite limited. Recent trends to allow students to use the same software tools as teachers use for grading CAD models suffer, from the methodological and conceptual approach, because grading provides feedback that is based on a finalized result, and thus always contains one assessment criterion that is related to the completeness of the solution. Being structured in this manner, it cannot be a direct part of the process and learning experience during the performance itself, that is the design, creation, and alteration of a CAD model. This paper reports on efforts and results from the first step of a two-step project to improve this situation as follows. Firstly, a learning-experience-oriented approach is presented, aimed at improving skill and expertise development in regard to robust and alterable parametric feature-based solid model creation. Secondly, a novel concept and a key metric for CAD model assessment and core behavior evaluation are introduced. Thirdly, a newly developed software tool is presented. This tool is dedicated to supporting students in putting into practice this learning-experience-oriented approach.

Keywords: reflection on performance and outcome, dormant deficiency, formative feedback, skill and competency development, CAD model alterability and associativity.
DOI: https://doi.org/10.14733/cadaps.2022.534-560

1 INTRODUCTION

Current developments and recent work in educational research have been aimed at creating awareness of and addressing the most prominent shortcomings and failures of current CAD education, in particular at institutions of higher education. Such efforts have provided new insights and recommendations, although the work is still limited and the results sometimes contradictory (see also discussions in [9,14,41]). Obviously, there is demand for a change of focus in traditional CAD education from the declarative knowledge relating to geometric algorithms and commands required for operating a CAD system, in the literature referred to as *command knowledge*, toward knowledge and expertise which can transcend a particular CAD system. This demand highlights the need for higher level thinking relating to what is commonly known as *strategic knowledge*, i.e. knowledge of the different methods of achieving a specific task (goal) and knowing how to choose among those methods. This requires, among other factors, a high-quality learning experience during frequent educational exercises in the CAD laboratory, providing opportunities for students to experience both the design and creation of their own CAD models and the re-design and alteration of them. This also includes promoting good design practice by relating CAD model attributes and parameters to part functionality and design intent, which, in turn, depends on the restructuring of curricula. Current efforts are aimed at designing alternative teaching approaches and integrating suitable elements of those into CAD education so that it is transformed into a more student-centered, learning-oriented and practice-oriented system. It needs to be better structured, so that it efficiently and effectively matches actual student learning outcomes with skills and competencies related to, among other attributes, spatial ability and mental visualization, cognitive model composition, and meta-cognitive processes including planning, predicting, revision, and, most importantly, self-assessment and self-regulation (cf. [2,49,51]).

However, translating the potential and benefit of those encouraging ideas and plans into educational practice also requires better structured and more frequent assessment and feedback than can be achieved with traditionally employed assessment and feedback techniques. This, in turn, requires the provision of a means for students to actively engage in having more frequent feedback to support self-assessment, thus giving them more ownership and responsibility regarding the learning process and the actual outcomes achieved. Unfortunately, within CAD education, dedicated techniques and student tools are not yet available to support the implementation of enhanced learning experiences that support the learner's efforts at self-assessment through more frequent and timely feedback, and, in particular, within the context outlined earlier, that assist students in both the assessment of the CAD models they have produced and the recognition of and learning from mistakes committed.

2 BACKGROUND, SCOPE, AND OBJECTIVES

2.1 Background and Related Work

The basic goal and purpose of any learning experience are seated in acquiring the skills, knowledge, and competency to change and improve an existing behavior or create a new one. Those changes in behavior should have measurable impacts that relate to key metrics indicating success in achieving the desired learning outcome. Within the context of CAD education for mechanical engineering (MCAD), those metrics are, in most cases, directly linked to the assessment criteria for CAD models created by students and submitted to teachers for grading. This approach has several shortcomings. Firstly, feedback is usually delayed due to the complexity of CAD model assessment and the rapidly increasing number of students in CAD courses at institutions of higher education. Secondly, the structure and quality of feedback based exclusively on CAD model grading is usually not sufficient to support learning from errors and developing metacognitive skills related to self-assessment, and subsequently self-improvement. Thirdly, feedback based on grading is applicable only to the final outcome, that is, to CAD models that are created and remodeled until they reach a finished version that is deemed by students to be ready for grading by teachers. Hereby, the process of learning from errors, reflecting on mistakes committed, and trying out alternative solutions based on newly gained understanding and improved skills, is somewhat short-circuited, and thus considerably reduced in its potential for the learner.

Recent approaches to the automation of CAD model grading are obviously capable of considerably reducing the time required for analyzing and assessing CAD models created by students, though the type and complexity of CAD model that can be analyzed, as well as the quality of the feedback that is generated, are still quite limited. Efforts to allow students to use the same software tools as teachers use for grading CAD models suffer not only from the limitations outlined, but also, from the methodological and conceptual approach, because grading, as traditionally practiced within the educational context, provides feedback that is based on a finalized result, and thus always contains one assessment criterion that is related to the completeness of a solution. Being structured in this manner, it cannot be a direct part of the process and learning experience during the performance itself, that is the design, creation, and alteration of a CAD model. Examples of recent approaches for technical drawings and 2D CAD files can be found in [8,19,21]. Examples of recent approaches for 3D CAD models and related empirical studies are reported in [2,3,14,15,27]. An interesting approach to providing visual feedback for automated CAD model grading using heat maps is reported in [24]. Further discussions on the subject of automated CAD model grading, including a summary of the literature and pointers to gaps in research, can be found in [12].

2.2 Scope and Objectives

One of the key characteristics of a parametric feature-based CAD system is the possibility of generating variants of a CAD model by altering the features of which it is composed. The features can easily be suppressed and changed by modifying their related dimensions. However, the correct and consistent generation of the new CAD model variants depends explicitly on how the features and feature dependencies have been created during the modeling process. In practice, it happens quite often that changes in some dimensions of a CAD model that initially appears to be correct and sound, will actually lead during model regeneration to a flawed and error-ridden variant. In other words, a CAD model without any apparent flaws and errors will degenerate into a deficient variant after it is regenerated according to dimension-oriented parameter changes. This phenomenon can be explained through what the authors define as *dormant deficiencies*. This novel concept is also used as a key metric not only for CAD model assessment in the context outlined, but also for students to assess success regarding their CAD exercise learning experience.

The presence of dormant deficiencies in a CAD model can drastically limit the model usability, and, as a consequence, quite often force the user to re-model from scratch in cases where a CAD model variant is required during the design process. From a pedagogical point of view, teaching novices how to recognize and avoid critical modeling situations that may introduce dormant deficiencies, represents an important issue that is quite challenging. Firstly, identification of the kinds of modeling mistakes that introduce dormant deficiencies is not trivial, because, as was mentioned earlier, the actual presence and impact of this kind of deficiency become evident only after the CAD model is altered and regenerated. Secondly, an efficient strategy to trigger and correct possible errors related to dormant deficiencies will require testing of the model during its development and not just at the end of the modeling process.

These, among other reasons, led the authors to initiate a two-step project. The objective of the first step is the design, development, and preliminary testing of a software tool aimed at supporting the identification of dormant deficiencies and critical situations, based on the collection and automatic change of values of all relevant CAD model dimensions, and reporting of all deficient situations resulting from the alteration of these dimensions. The objective of the second step of the two-step project is to provide the tool to all the students of the current CAD course, in order to improve their learning experience and to assist in self-assessment during modeling exercises. The aim of the present paper is to present important elements of the first step of the two-step project, as follows. Firstly, this paper presents an approach aimed at improving the development of skills and expertise in regard to the creation of robust and alterable parametric feature-based solid models. It aims to achieve this by systematically analyzing and enhancing the learning experience during exercises related to CAD laboratory and course assignments. Secondly, the paper reports on the concrete development of a novel concept, namely dormant deficiency. This concept is used for CAD model assessment and it is the key metric that identifies success, but it can also identify core behaviors that prevent the metric from being attained. Thirdly, the paper presents and discusses the development and implementation of a software tool designed for students and learners, to enable and support the putting into practice of certain components of this learning experience.

3 APPROACH, DESIGN, AND DEVELOPMENT

3.1 Current Gaps between the Learner and Desired Learning Outcomes

Within the recently restructured MCAD course, various modeling exercises are provided employing a novel teaching approach that systematically utilizes negative knowledge in addition to traditional lectures and tutorials (cf. [39,40]). Each of these modeling exercises addresses a particular learning goal. Outcomes of the exercises, in the form of CAD models created by students, are collected and analyzed, to identify shortcomings and errors which usually remain hidden from students due to their limited domain knowledge and expertise. Results are then used as input for formative assessment and feedback. Currently, a series of design and modeling exercises is being administered, where each exercise corresponds to the domain subject being taught within individual course units associated with it. The exercise assignments are designed to begin with a less complex design object, and gradually increase in complexity of modeling task and object shape according to progress made in the course and the domain subject being taught.

The newly developed educational framework is now in its fourth year of successful operation within the department where the authors operate. Therefore, various forms of student feedback embodied in observational records, CAD models, final exam material submissions, and questionnaires have been obtained and systematically filed for the purposes of post-processing and analysis (see also [32]). As expected, this has revealed several gaps that still exist between the performance, skills and competency development of the students and the learning outcomes. In particular, it was a major concern that learning outcomes were not achieved during particular CAD modeling exercises which were focused on domain concept understanding and modeling skill development related to robustness and alterability of parametric feature-based CAD models. A closer look at this situation showed that one shortcoming was the response time and frequency of formative feedback from teacher to student. Obviously, one straightforward remedy for this imperfection is to have more frequent formative feedback (see also discussions in [18,22]). However, this can only be achieved by increasing the engagement of students through enabling and supporting them to practice self-assessment. This has led to the re-design of the learning experience for particular CAD modeling exercises, and that, in turn, has required the development of a novel CAD model assessment metric that can also be used as a key metric to evaluate core student behavior. A software tool has also been designed specifically for the needs of students, to engage in and support self-assessment of CAD models in regard to robustness and alterability.

3.2 Toward an Improved Student-Centered Learning Experience

Efforts were made to reduce the gap between actual student learning as achieved and learning goals as pre-assigned within the CAD course which is currently a part of the curriculum for the Laurea degree in mechanical engineering at the institution represented by the authors. These efforts resulted in a systematic approach being adopted in order to enhance the learning experience, in particular during exercise performance related to CAD laboratory and course assignments. This approach is structured according to the elements of learning and user experience design, and it also includes some backward-working segments between individual stages such as analysis and design (cf. [6,13,48]). Those main elements and the stages used within the approach taken can be outlined as follows.

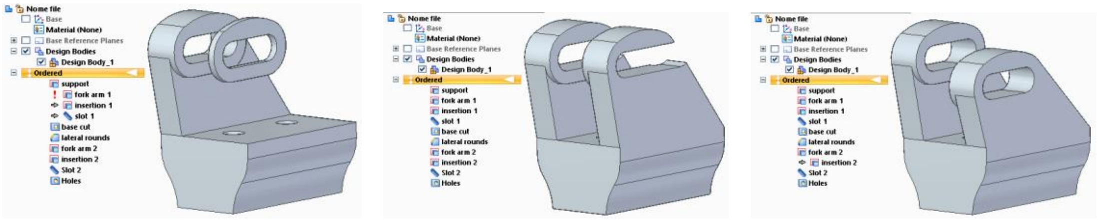
3.2.1 Key metric

At this point, we should recall the main objective of the learning experience subject to design and development, which is to create robust and alterable parametric feature-based CAD models. For this purpose, a key metric, in the form of so-called *dormant deficiencies*, has been formulated and developed. This key metric has been designed not only to represent a measure of success, but also as a supporting concept to aid learning and to assist in understanding the central ideas and domain concepts of CAD model alterability and associativity. Dormant deficiencies can be conceptualized as errors in associativity, which were introduced during the modeling process by making mistakes in the specification of dependencies between geometric entities and features. However, the effect and impact of these mistakes on the CAD model remain dormant until an actual CAD model regeneration is triggered and executed through an alteration. In this context, the outcome in regard to deficiencies is related to which of three different error situations occurs. Accordingly, dormant deficiencies are classified as type I, type II, or type III, as follows.

A type I dormant deficiency leads to deficiencies in features. The regenerated CAD model contains features, which are labeled with a status of warning or failed. This will result in a shape that is incomplete and / or incoherent, as shown in Figure 1(a).

A type II dormant deficiency leads to deficiencies in geometry. The regenerated CAD model does not contain any features labeled with a status of warning or failed, but its shape is incoherent. For example, it may contain a slot that has an open gap in its wall, as shown in Figure 1(b).

A type III dormant deficiency leads to deficiencies in the design rationale. The regenerated CAD model does not contain any features labeled with a status of warning or failed, nor is its shape incoherent, but the shape does not meet the design requirements (cf. [38,44]). For example, it may violate a symmetry condition as shown in Figure 1(c).



(a) (b) (c)

Figure 1: Examples of dormant deficiency types. From left to right: (a) type I dormant deficiency related to deficiencies in features, (b) type II dormant deficiency related to deficiencies in geometry, (c) type III dormant deficiency related to deficiencies in the design rationale.

Note that, in order to understand and avoid a type III dormant deficiency, engineering design knowledge and a full understanding of the design rationale of the component is required (see also discussions in [4,38,44]). However, within the context of the CAD course as outlined elsewhere in this paper, where students can be considered novices, the focus is mostly on type I and type II dormant deficiencies.

The nature and category of dependencies between geometric entities and features are determined, among other characteristics, by their range in regard to features and the class of features they are associated with, that is profile-based features and non-profile-based features. This results in intra-feature dependencies and inter-feature dependencies. Here, the former represent dependencies between geometric entities within one and the same feature, while the latter refer to dependencies between geometric entities of more than one feature. From this viewpoint, particular modeling situations can be grouped as listed in Table 1, where dormant deficiencies are introduced most likely by novices, and thus turn into critical situations (cf. [39]).

	Intra-Feature	***Inter-Feature***
Profile-Based	profile creation • definition of dimensions • definition of constraints	reference plane selection profile creation • definition of dimensions • definition of constraints extrusion options setting
Not Profile-Based	N/A	entity selection

Table 1: Modeling situations in respect to feature class and range that become critical if dormant deficiencies are introduced.

Note that, in the case of profile-based features, deficiencies within intra-feature dependency are most likely to be introduced during profile creation, when associativity is created between the 2D geometric entities of the profile. Here, in cases where the profile is comprised of a basic non-complex outline, the CAD system usually creates rudimental geometric constraints automatically. However, in the case of complex profiles, the user is required to explicitly define all the constraints required. Currently, most commercially available CAD systems provide functionality for testing the condition of those profiles, that is, whether they are fully constrained or not.

From a more theoretical and formal point of view, the key metric as introduced above can be defined as an operator, which takes a CAD model as input and returns dormant deficiencies. This can be expressed by a mapping denoted by *M*, the set of CAD models denoted by *CADMODEL*, the set of natural numbers, excluding zero, denoted by **N**, and the set of dormant deficiencies denoted by *DD* as shown in Equation 3.1.

$$M\colon CADMODEL \rightarrow (DD \cup \varnothing) \times (\mathbf{N} \cup \{0\}) \tag{3.1}$$

Here the set *DD* consists of the union of disjoint subsets DD_k of dormant deficiencies for each respective type, as formulated in Equation 3.2. Note that the index $k \in \{I,II,III\}$ is used for explicitly indicating the dormant deficiency type by using the same roman numerals as in the dormant deficiency concept introduction and example discussions elsewhere in the paper.

$$DD = \bigcup_{k=I}^{III} DD_k \tag{3.2}$$

Note that results of cases in which the software-tool-assisted CAD model assessment cannot reveal any dormant deficiencies — the CAD model has no dormant deficiencies under the metric *M* — that is, the set-valued correspondences between *DD* and *CMD* are empty, are represented accordingly by both an empty dormant deficiency set and a deficiency counter value of zero (cf. Equation 3.1). With the set of CAD model deficiencies denoted by *CMD*, the relationship between individual dormant deficiencies $dd \in DD$ and actual CAD model deficiencies $cmd \in CMD$ can be formally expressed as a set-valued correspondence (cf. [1,43]) of deficiencies denoted by *SCD* with $SCD \subseteq DD \times CMD$. Taking into account that this correspondence needs to structurally encapsulate the three types of dormant deficiencies (see also Figure 1, Table 1, and Equation 3.2), this set-valued correspondence can be defined as shown in Equation 3.3. For more details on how individual sets and correspondences are computed with the software tool and displayed in regard to the properties and characteristics of their actual entities within the visual simulation display see the Appendix, Figure A1, and Figure A2.

$$SCD = \bigcup_{k=I}^{III} \{DD_k \times CMD\} \tag{3.3}$$

Given a vertex set denoted by *V* and an edge set denoted by *E*, a heterogeneous relation defined as $R: V \rightarrow E$, can be characterized by a graph that is a pair $G = (V,E)$ of sets such that $E \subseteq V \times V$, with elements of *E* being 2-element subsets of *V* and $V \cap E = \emptyset$. We may speak of a bipartioned graph $G^2 = (V,E)$, also referred to as a bipartite graph (cf. [7,10,16]), if *V* admits a partitioning into 2 subset classes, interpreting *V* as a set of vertices on the left and the right, and *E* as a set of edges such that every edge has its ends in a different set. Now, the set-valued correspondence of deficiencies *SCD* from *DD* to *CMD* can also be characterized by its bipartite graph G^2, that is the subset of the product space $DD \times CMD$ as defined in Equation 3.4.

$$G^2(SCD) = \{(dd,cmd) \in DD \times CMD \mid cmd \in SCD(dd)\} \tag{3.4}$$

Here we shall say that $SCD(dd)$ is the image or value of *SCD* at *dd*, that is, the set of CAD model deficiencies that correspond to a dormant deficiency. The image of *SCD* denoted by *IMG* is the union of all the images $SCD(dd)$, when *dd* ranges over *DD* as shown in Equation 3.5.

$$IMG(SCD) = \bigcup_{dd \in DD} SCD(dd) \tag{3.5}$$

In regard to the actual system implementation, these images are the actual CAD model deficiencies that are the result of dormant deficiencies that were activated during CAD model regeneration through a simulated alteration of CAD model parameters. However, it is beyond the scope of this paper to advance the formal framework of the key metric and the concept of dormant deficiencies, as outlined, into a more detailed formal analysis through relational mathematics and graph theory in regard to multi-valued mappings and sets.

As outlined earlier, most software tools supporting automated grading and assessment of CAD models that are currently provided to students in CAD courses at institutions of higher education are limited by their metrics and assessment approach. In particular, the metrics they use are of a rather static and exclusive nature, relying heavily on the final outcome, that is the completed CAD model, which then has its data structure compared to that of a fixed reference solution. Such approaches are not structured suitably to assess CAD model quality in regard to robustness and alterability, due to their static and exclusive nature, which usually leads them to discount CAD model re-creation processes and their impact after alteration. They are also not sufficiently structured to explicitly support formative self-assessment carried out by students during individual steps of the modeling process, as part of their exercise work. This problem arises because the software tools used are unable to assess partially created CAD models, since they appear to be incomplete according to the metrics and rubrics provided in relation to the exercise specification and the fixed reference solution associated with it. The newly developed metric of dormant deficiencies is a dynamic and more inclusive measure that takes into account the impact that alteration and the CAD model re-creation process have on the original modeling process and its outcome. It is, therefore, a more process-oriented and simulation-based assessment approach, and so those limits outlined above can be overcome in a straightforward manner, as described in detail in the following subsections.

3.2.2 Core behaviors and supporting resources

Based on this key metric, and working backwards, the core behaviors that prevent the metric from being reached have been determined as follows. Students need to be able to recognize critical situations during modeling that might result in dormant deficiencies, need to know what not to do in order to avoid dormant deficiencies being introduced into the CAD model, and need to know how to properly define profiles and sketches, along with their related dimensions, in a manner consistent with the design intent and functionality of the part subject to modeling in the CAD environment.

Next, the necessary knowledge, skills, and resources required to enable and support those core behaviors need to be determined. These include, for example, defining effective associations, which is a skill built upon the knowledge of what makes certain dependencies and constraints effective, and which ones are most likely to result in dormant deficiencies and thus are better avoided. Knowing how to accurately create associativity is a knowledge-based task requiring a certain amount of practice. Here, within the context outlined earlier, students need to experience what it means and looks like to actually be able to create an alterable feature-based CAD model. This requires resources that allow students to systematically engage in self-assessment regarding the quality of the outcomes achieved during exercises, that is the CAD models created and the understanding and skills developed and improved. This requires some assistance. The students need to experience some phenomena, which bring important concepts to life, and they need a means to help assess their CAD models in regard to the key metric.

Based on those requirements, a novel software tool has been developed that enables students to assess a parametric feature-based CAD model in regard to dormant deficiencies. This software tool, which represents an important resource central to implementing the assessment approach outlined earlier, also provides experience for students in regard to both critical situations and dormant deficiencies. The tool shows visually, and in real time, the change of modeling parameters and the effects that dormant deficiencies can have on a CAD model. Details of the software tool, its functionality, its technical components, and the information architecture of the user interface, are described in the next subsection.

3.3 Software Tool Development

3.3.1 Requirements and approach

Requirements discussed in detail below are of two kinds: those that are more closely related to the goals of the learning experience design, and those that are more closely related to the supporting resources, that is, more specifically, to the software tool design.

Firstly, one central requirement is to provide an experience that brings some phenomena of important domain concepts to life. Within the given context, this relates to the concepts of robustness and alterability of parametric feature-based CAD models and, subsequently, to the concept of dormant deficiencies. Students need to experience first-hand during practical exercise work what it means to make mistakes during the creation of dependencies and thus introduce dormant deficiencies, resulting in CAD models that are neither robust nor alterable. They also need to see with their own eyes how those dormant deficiencies that they introduced during the modeling process can impact the CAD model they created in down-stream processes and model reuse requiring redesign, alteration, and subsequently CAD model re-creation. Of particular interest here are the structural and visual phenomena related to invalid features and invalid geometry that are produced by the CAD system during the re-creation of the CAD model after a parameter, such as a dimension value in a profile or feature, has been altered and has thus activated dormant deficiencies.

Secondly, another central requirement is to provide a means for supporting CAD model assessment in regard to the key metric, that is, dormant deficiencies. Students should be able to run automated tests under various assessment scenarios to see if their models contain dormant deficiencies or not. Those scenarios should allow for including / excluding fully constrained profiles and finite feature extrusion dimensions. Students should also be allowed to set breakpoints during the automated assessment process, to be able to investigate and study interim test results, particularly errors and deficiencies, stepwise and in detail.

Thirdly, because the software tool is aimed at supporting formative self-assessment during exercise-based learning experiences, it is also important that CAD model assessment using the software tool can be conducted at any stage during the modeling / remodeling process. This is in stark contrast to automated grading tools generally used by teachers that can be applied only to final outcomes, that is, to finished CAD models, to support a kind of summative assessment.

Fourthly, a means is required to provide students with assistance in locating and analyzing mistakes committed earlier during the creation of dependencies. This is important to support reflection on, as well as learning from, errors. These are two important elements in the development of skills and expertise in regard to both knowing what not to do in certain situations, in order to avoid mistakes, and knowing what to do instead (cf. [32,39]).

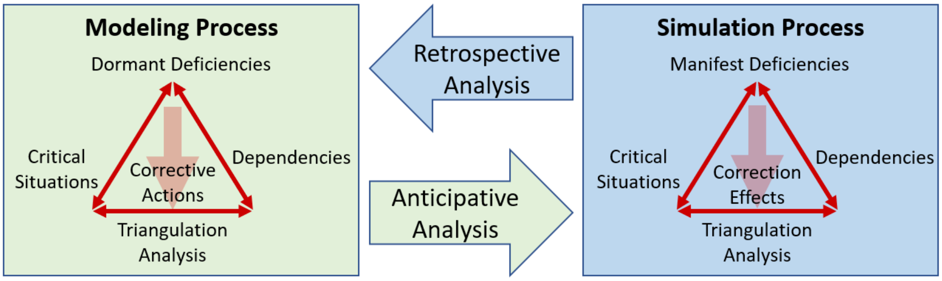


Figure 2: Integrated triangulation analysis across processes.

These requirements have been approached within the framework design and system architecture development (details are provided in the next subsection) by considering a simulation-based CAD model assessment in regard to dormant deficiencies with a real-time oriented graphical student user interface and a cross-linked view between the CAD modeling environment and the simulation environment that supports dedicated user interactions such as brushing (cf. [37,47]). Simulation of the CAD model alteration process and the assessment of the CAD model in regard to the metric are approached in a straightforward manner by linking the simulation module and assessment algorithms to the CAD system. Various scenarios are then executed by the CAD system regarding test and breakpoint settings that are specified through the user interface and controlled by the software tool. These include the CAD model re-creations in regard to parameters that were altered. As both the CAD modeling environment and the simulation-based assessment environment are cross-linked, students can see and experience in real-time the effects that dormant deficiencies, if present, can have on CAD model re-creation after model alterations have taken place.

Within this approach, dormant deficiencies are not only activated and detected, but also recorded and analyzed. This is done in regard to the critical alteration situation and the dependencies context they appeared within and the number and kind of deficiencies they caused. This structured feedback, together with the user interactions and cross-linked view provided, as outlined earlier, offers students basic information and functionality to assist in locating and analyzing mistakes committed earlier during the modeling process, which introduced dormant deficiencies. A schematic overview on how this approach is translated into concrete actions is given in Figure 2. First, through triangulation of information from the data sources in regard to critical situations, deficiencies, and dependencies across the simulation process and the modeling process, integrated with retrospective analysis, possible locations of errors in dependencies can be determined in the form of features and their associated geometry that was rendered invalid during

CAD model re-creation. This provides a base for more detailed analysis of the dependencies within the modeling process triangulation. After error correction attempts, another round of simulation-based assessment can be executed. If the error correction attempts were successful, the deficiencies will have been eliminated, which then removes the data source in regard to deficiencies within the triangular analysis (see again Figure 2). This, in turn, eventually leads to the collapse of the triangulated data sources into just situations and dependencies, where anticipative analysis finally confirms that a CAD model is robust and alterable according to the key metric. Note that with this simulation-based and process-oriented approach, CAD model assessment requires neither a finished modeling outcome nor a reference solution in the form of an ideal CAD model for comparison. Assessment can be performed with the software tool at any stage during the modeling process, and thus it fully supports students in formative self-assessment during exercise-based learning experiences and skill development. Details of how those requirements and the approach as described above are translated into software tool design and implementation are presented in the next subsection.

3.3.2 Design and implementation

The software tool design is based on the approach of an alteration simulator operating on the parametric structure of feature-based solid models. The software tool design is conceptually integrated with the current CAD course structure and resources in regard to CAD model assessment and feedback. The student user can control the simulation process through an interactive graphical user interface. Once the user starts the simulation and analysis process, the software tool automatically collects and changes, one-by-one, all the dimensions that are related to sketches, profiles of features, and extensions of finite extrusions. Note that, due to the cross-linked structure between the software tool and the CAD system, each dimensional change is immediately reflected in the shape of the CAD model being regenerated. The user can define the proportions of the CAD model alterations that are used during the simulation process. The current system default is 10%. After each change in a dimension value, the number and type of features that have entered a failed or warning status are recorded within the report-based feedback section of the user interface. The user can also set breakpoints in order to pause the simulation and analysis process. Breakpoints can be set either after each change of a single dimension value or after a dimension has been both increased and decreased. Following the activation of breakpoints, the simulation and analysis process can be terminated, or it can be resumed with either the current breakpoint settings or an uninterrupted execution until the end of the process is reached. The graphical user interface has been developed and implemented within the Excel VBA framework and is provided through the dashboard of the software tool. This interactive graphical user interface is a part of the cross-link structure and brushing supporting the visual simulation display (VSD). The VSD, and in particular the user interface within the dashboard, have been designed not only to mediate communication and user interaction through computer screens, but also to inform and to provide feedback in an effective and efficient manner in regard to the target user group, that is, the students. The design is guided by functional / data requirements — outlined elsewhere in this paper — and is grounded in principles of visual usability and the application of various insights and groundwork of cognitive and perceptual psychology. Here, in particular, the latter is aimed at improving the intuitive aspect of the digital interface and system, and subsequently experiences in regard to how human users perceive them and interact with them. A selection of the functional and visual design characteristics considered central to the user interface within the dashboard and also significant for the VSD were translated and developed as follows.

We should recall that the role of the software tool is that of a central resource aimed at supporting students in formative self-assessment during exercise-based learning experiences. Therefore, from the beginning, the principles of familiarity, and, in relation to it, consistency, most strongly affected basic decisions regarding the interface design, and subsequently the system development, as outlined next. Reasons for that were related to various factors. External consistency, and in particular familiarity, allow users to transfer expectations and experiences they have acquired from systems and interfaces that were similar, thus allowing for an almost immediate interaction without first having to overcome a tedious learning curve. Underlying factors for that are attributed to the tendency for users to draw on their cumulative experience from other digital systems and experiences that support them in their understanding of how things work and what to expect when encountering something new (see also Jacob's Law and consistency in visual design and usability, for example in [42,50]).

Now, leveraging those existing mental models and related knowledge-effective experiences with the software tool can help students to focus quickly on their tasks — CAD model assessment and learning from errors — since they spend less mental effort and energy on learning and getting used to a new interface and system. Since the interface design systematically supports this user ability of transferring parts of mental models and related knowledge between interfaces and systems appearing similar, it not only helps to reduce the cognitive load, but allows the students to be more dedicated to achieving their primary objectives. This is a very important factor for educators, taking into account the current situation, where students in the CAD course not only have to learn domain-specifics and how to master a CAD system, but are now being confronted with a newly developed CAD model assessment tool which they have never used previously. This, among other factors, led to a decision to base the design and development of both the software tool and the functional and visual characteristics of the VSD on applications the students can be considered familiar with in the given context, namely the CAD modeling system that is used during exercises and MS Excel from the Microsoft Office suite. This early strategic decision had an impact on the design of the software tool's system architecture and implementation, as described elsewhere in the paper. However, despite this impact, many visual usability principles could be successfully applied during the design, as demonstrated below.

Based on the strategic decision outlined above, it is known which interfaces and user interactions the student users can perceive as related. This, in turn, allows us to anticipate to some extent their expectations, which helps us to base decisions on functional and visual design characteristics aimed at establishing consistency. For example, to identify data related to dormant deficiencies in all forms of information and feedback the software tool provides, that is the dashboard, the assessment reports, etc., a label in the form of a colored symbol is used, which closely resembles the one that is used by the CAD system to indicate errors and deficiencies detected in a CAD model. In general, to support consistency in the design, most basic interface conventions were defined in regard to user expectations in as much as those can be anticipated in the given context.

The visual hierarchy of the VSD as shown in Figure 3 has been structured based on a combination of position and contrast in size to relate the relative importance of elements to their visual prominence based on contrast. Here, as excellently formulated in [46, p.33], "contrast is about creating visible differences between elements and is the essential ingredient of visual hierarchy". For the cross-linked view pertaining to the CAD modeling environment, placement and proximity to frame (cf. discussions on Gestalt psychology principles related to visual interface design in [26,47]) have been used to affect contrast and perception of hierarchy regarding the rendered CAD model shape and its subsequent model re-creations during the simulation process. For the interface of the dashboard, placement and proximity of elements and eye behavior have been used to improve perception of hierarchy. For example, in the case of the former, elements such as initial conditions, test results, and simulation breakpoint settings were grouped together and isolated from others by having white space surrounding them. This space creates contrast, thus enhancing visual dominance. In case of the latter, important information and central controls have been positioned together in the upper left quadrant of the user interface, as elements in the top left position are most frequently viewed and are interpreted as more important (see also discussions on F-shaped reading patterns and the inverted pyramid principle in [31,36]). An additional reason for this grouping and positioning in regard to the first interface quadrant was to enable the design of a visual resting space, that is white space for the eye to rest on in a screen (cf. [42]). Note that this visual resting space can also temporarily be used during assessment to literally park the floating dialog window that lets users control and resume the process at breakpoints, thus reducing visual clutter (see again Figure 3 and also Figure 4) without limiting information view and control. For users that have a rather small screen at their disposal, this visual hierarchy design also allows them to enlarge the dashboard display and scroll to the left without losing the overall view and main control of the interface. Contrast in size has been used for the large group of information regarding critical simulation situations, deficiencies, dimension values, etc., to increase the visual weight, and thus prominence of, this information set during the simulation / assessment process. This characteristic of the visual hierarchy design is aimed at helping to guide the student user's attention and focus on the screen locations within the VSD that are central during the visual and simulation-based assessment process.

Within the visual interface design presented, color has been used strategically to highlight and draw attention. This policy is consistent throughout the CAD model assessment feedback in the data tables such as the initial conditions and the test results, where color is used to indicate and draw attention to dormant deficiencies detected and to features that were identified as not fully constrained, or otherwise found to be defective. Color is also used during brushing within the cross-linked view to highlight a dimension within the CAD modeling environment after its counterpart within the list-based data set in the dashboard has been selected. The color scheme used for the visual interface design has been customized so that it can be accessible to both the color impaired and the viewer with full color vision (cf. [17,37]). The legibility of the currently used color scheme design has been verified with a complementary software tool, namely *ColorOracle2* [25], an open-source simulator of color-impaired vision.

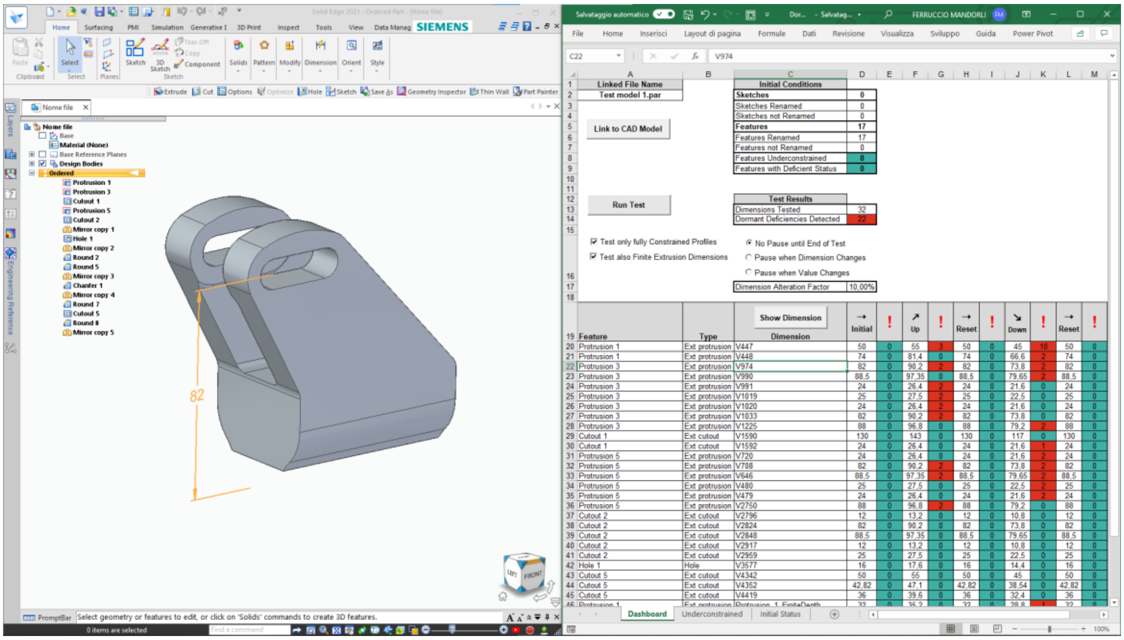


Figure 3: Example of the visual simulation display enabling a cross-linked view in real time during simulation and visual analysis.

Finally, some consideration has been given to the overall appeal of the VSD design as it affects perception of use due to the aesthetic-usability effect (cf. [28,45]). This is because users often perceive interfaces with aesthetically pleasing designs to be more practical. Hence, an aesthetically and visually pleasing design can actually influence usability, extend credibility, and get users to be more tolerant of minor usability issues [50]. In particular, the interface has been kept as a minimal design to allow students to focus on the task at hand and the feedback generated in real-time during simulation-based CAD model assessment. For example, this includes switching off the grid in the background and reducing the ribbon (menus and toolbars) to a minimum as system default. In addition, the entire wording within the dashboard has been designed to appeal to novice users by avoiding expert language and difficult domain-related terms and symbols. Instead of referring to 'brushing' or 'breakpoint scenario setting', phrases in plain English are employed in the interface, such as 'show dimension' and 'pause when value changes'. These additional design efforts are aimed at contributing to the overall appeal of the VSD design by providing a personality for the interface that fits best with the characteristics of the persona of novices and student users and their expectations about what the software tool does and for whom it is meant. The feedback generated by the software tool is provided in two modes. One mode is interactive, with feedback provided in real time, visually displaying the CAD model and its shape being modified and re-created within the CAD modeling environment during the simulated CAD model alteration. The other mode is static, with feedback provided in the form of a structured report that contains general information such as the number of features with profiles not fully constrained, and information in the form of linked lists regarding critical situations that were encountered during the simulation. The latter are structured to enable the student to systematically reflect on mistakes that led to deficiencies in the CAD model. In particular, these lists provide information that supports the development of understanding and insight on what went wrong during model alteration, where undesired changes and errors in the form of invalid features and flawed / incorrect model geometry occurred as a result of dormant deficiencies, and their possible causes.

To provide an overview of the appearance of the visual simulation display and the manner in which student user interactions pertaining to assisted CAD model assessment are supported by this prototype implementation, part of an actual analysis session is presented. This was executed as a segment of the CAD model analysis example in connection with the material shown in Figure 8, which is a part of the presentation and discussion of the software tool test and evaluation in the next section. Detailed below are some selected individual user interactions and the operational steps involved in regard to the related interface components and cross-linked views of the software tool environment, as well as the CAD modeling environment. These are as shown in Figure 4.

After the CAD modeling environment has been cross-linked with the software tool environment, automated analysis of the linked CAD model in regard to basic assessment criteria is initiated. This includes checks to determine the condition of the CAD model in regard to the sketches and features it contains and whether those are not fully constrained or have any other deficiency that can be checked through the basic functionality of the CAD system. Generally, the first student user interactions involve creating a cross-link between the two system environments and the setting of basic tool parameters and process breakpoints. Also, some additional parameter settings can be applied to adjust the simulation-based analysis. These relate to the processing of particular kinds of profiles and feature dimensions, whenever deemed necessary. Those initial interactions can be performed using the dashboard of the software tool environment, as shown in the upper left-hand side in Figure 4.

Next, the simulation-based analysis process is started, according to the breakpoint scenario defined by the student user. This is done by activating the "Run Test" button, for example with a mouse click. The button turns red during the entire analysis process. If no breakpoints have been set, the process terminates when all CAD model alterations as defined have been executed and analyzed. The results of the simulated parameter alterations are shown in real time within the visual simulation display. In particular, the CAD model shape, which is regenerated after each alteration, is shown within the cross-linked CAD environment, while details of individual alteration situations and dormant deficiencies, if detected, are listed in the cross-linked software tool environment. If breakpoints have been set, the simulation-based analysis process pauses accordingly, as shown on the upper right-hand side in Figure 4. The student user can then manage further breakpoints through the breakpoint interaction window, which can be moved temporarily to the white space within the user interface. This reduces screen clutter, as described elsewhere in the paper.

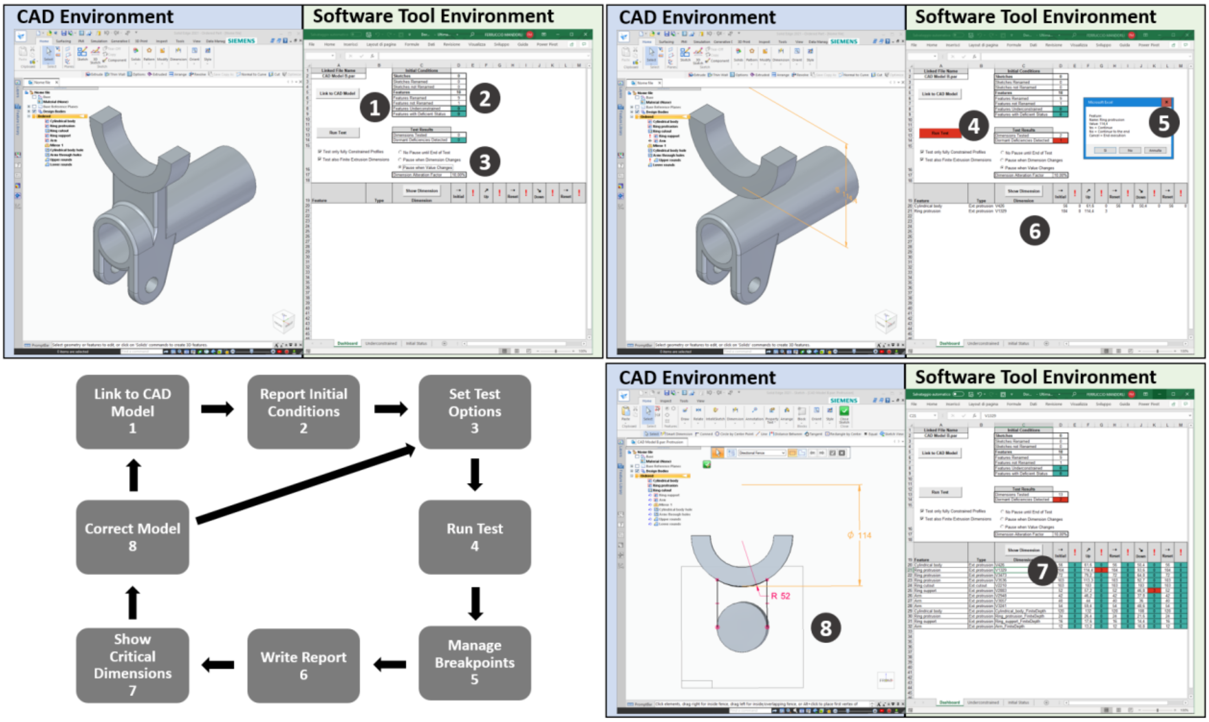


Figure 4: Example of student user interactions and related interface components during software tool assisted CAD model assessment.

If dormant deficiencies have been detected during the application of the software tool, further tool-assisted CAD model analysis can be conducted. This includes triangulation of information provided to the students from within the visual simulation display and the tool-generated reports, to locate and identify errors in associativity, usually caused by flawed dependencies in regard to features and their associated geometry that became invalid during CAD model regeneration. A snapshot of a concrete example of tool-assisted analysis of dormant deficiencies is shown on the lower right-hand side in Figure 4. In this example, the regenerated CAD model contains failed features and geometric deficiencies in the form of an incomplete shape. The critical dimension has been detected by the tool and its value is highlighted.

Note that the size and spatial position of individual windows of both the software tool environment and the CAD modeling environment can be freely arranged by the user. The layout of the interface components and cross-linked views, as shown in Figure 4, is just one example chosen to accommodate, as well as to spatially optimize, the figure layout as shown.

The newly developed software tool features a technical architecture that leverages API-based functionality provided by commercially available CAD systems to support a modular and highly cohesive system architecture, as shown in Figure 5. Within the current implementation, the modeling environment deploys a commercially available parametric feature-based solid modeling system, namely *SolidEdge* from Siemens AG. At present, the simulator module and the assessment criteria used for the CAD model deficiency analysis are implemented within the CAD modeling environment as procedures and functions based on Visual Basic for Applications (VBA). The computation and interface structures of the assessment report and the dashboard of the simulator module are implemented within Microsoft *Excel* using VBA functions and procedures. Usability and functionality tests have been successfully conducted with prototype and finalized versions of the implemented software tool. During the software tool test, simulations were carried out on a wide selection of parametric feature-based CAD models that had been created and submitted by students during exercises related to CAD laboratory and course assignments in the last academic year. Details of the software tool test and the formative usability study conducted are presented in the next section.

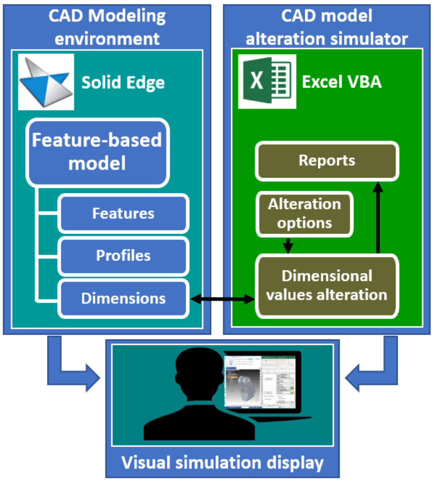


Figure 5: Overview of technical architecture of the software tool.

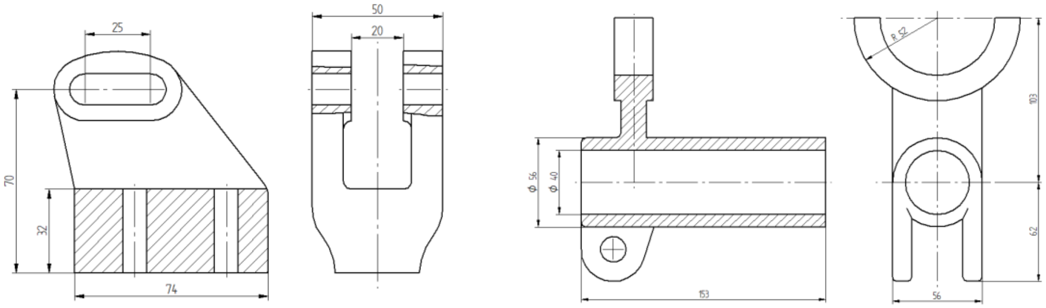
4 TEST AND EVALUATION

4.1 Outline

As described in the literature, several interdependent main aspects of usage need to be considered for design and evaluation of technology and devices, and the services and experiences they provide for a particular group of users within a particular context (cf. [5,11,23,33,35]). Within the educational context as described earlier, this relates to the software tool that represents a supporting resource central to the learning experience design, as described elsewhere in this paper. According to [34] these aspects should be reflected in functionality, usability, and user experience. Here, the last-mentioned, in the context given, relates to the experience of using the software tool, and thus concerns the wider relationship between the students and the tool. This aspect will be evaluated in more detail in a summative assessment during the second step of the project. Evaluation of functionality, which is a technical aspect related to the software tool, needs to answer questions and take into account issues such as whether the tool does what it is designed for (detecting dormant deficiencies and supporting students to locate, understand, and learn from the errors they committed), whether it is useful to the target user (students of the CAD course), and whether its performance is efficient and reliable. Evaluation of usability, which relates to characteristics of the interaction between the user and the software tool, is an important issue, especially during design and development. Here, in regard to [23], perhaps usability is best assessed as quality in use based on "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (see also discussion in [5]). Efforts to address those issues and results achieved within the first step of the project described in this paper are presented in the next subsections.

4.2 Software Tool Functionality and Reliability

Testing and assessment of the software tool in regard to its functionality, performance, and reliability during development have been conducted as follows. Evaluation of how well and how thoroughly the software tool can assess parametric feature-based CAD models in regard to dormant deficiencies was conducted by using CAD models that had been created by students for CAD course exercise assignments in the previous academic year.



(a) (b)

Figure 6: Examples of actual CAD course exercise assignments. From left to right: (a) outline and overall dimensions of CAD Model A, (b) outline and overall dimensions of CAD Model B.

To keep the presentation and discussions concise, transparent, and clear, the examples reported in this paper are limited to a selection of examples of two CAD models, referred to as CAD Model A and CAD Model B. The shape and some overall dimensions of these CAD models are shown in Figure 6. As a first step, all CAD models that already had deficiencies that could be detected without the software tool, such as features that were invalid or under-constrained, were removed. Hence, 45 CAD models out of the initially selected 59 CAD models remained for the test and assessment reported next.

	Dimensions Profile	Dimensions Sketch	Dimensions Finite Protrusion	
CAD Model A	196	203	101	500
CAD Model B	117	38	57	212
	313	241	158	712

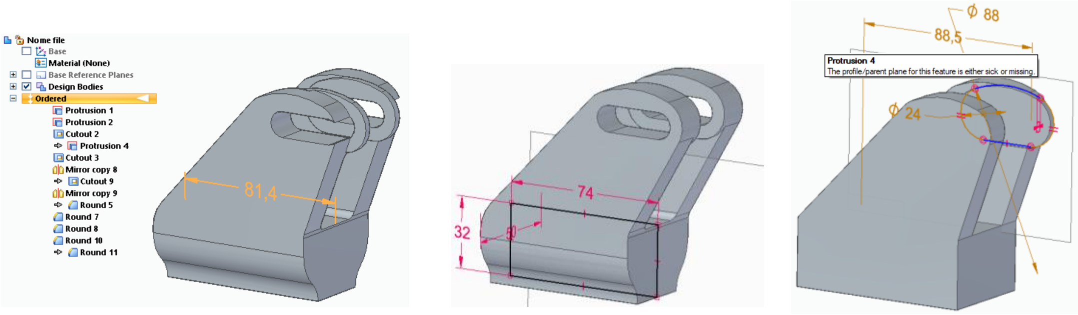
Table 2: Overview of the type and number of dimensions tested during CAD model assessment.

A total of 712 dimensions of examples of CAD Model A and CAD Model B (see details in Table 2) were analyzed during the simulation-based assessment. As a result of this analysis, 27 CAD models were found to contain dormant deficiencies (see details in Table 3), and this result was confirmed by an independent manual analysis that was conducted by the authors, which concluded with the same outcome.

Dormant Deficiency	Detected	Not Detected	
CAD Model A	16	16	32
CAD Model B	11	2	13
	27	18	45

Table 3: Number and margins of CAD models with and without dormant deficiencies detected.

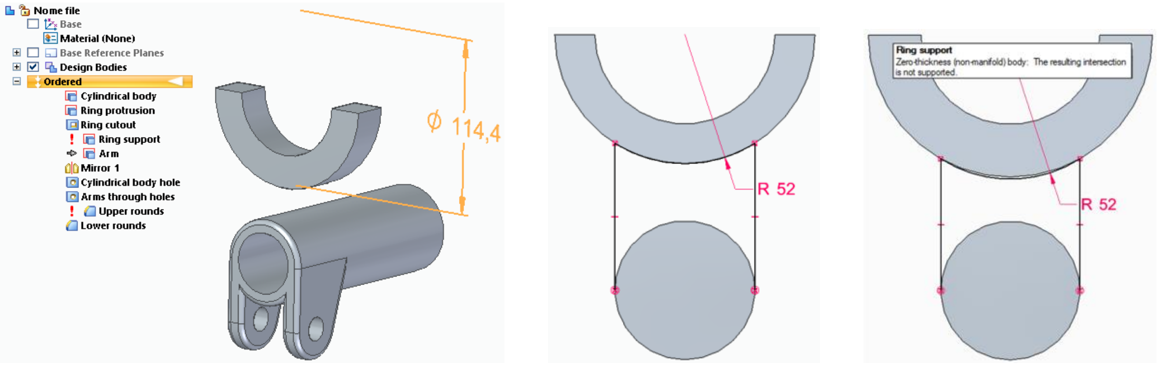
Some examples of type I dormant deficiencies that were automatically detected by the software tool and verified by human assessors are shown in Figure 7 and Figure 8. In the example shown in Figure 7(a), the dormant deficiency was activated by a change in the dimension value from 74 mm to 81.4 mm (see Figure 7(a) and Figure 7(b)). Regeneration of the CAD model resulted in feature warnings (see Figure 7(c)) and deficient geometry in the form of an incomplete shape. The reason for this dormant deficiency can be traced back to an error in associativity related to the reference plane of the extruded protrusion feature that could not properly be created again after regeneration of the CAD model (see again Figure 7(c)).



(a) (b) (c)

Figure 7: Example of a CAD Model A type I dormant deficiency. From left to right: (a) regenerated CAD model containing failed features and geometric deficiencies in the form of incomplete shape, (b) state of the CAD model before the alteration of a critical dimension, (c) state of the regenerated CAD model after the alteration of a critical dimension.

Here, a re-selection of the reference plane for the protrusion feature, based on the inner planar surface of the link yoke, can resolve the problem. This not only removes the dormant deficiency, but also makes the CAD model more robust and alterable.

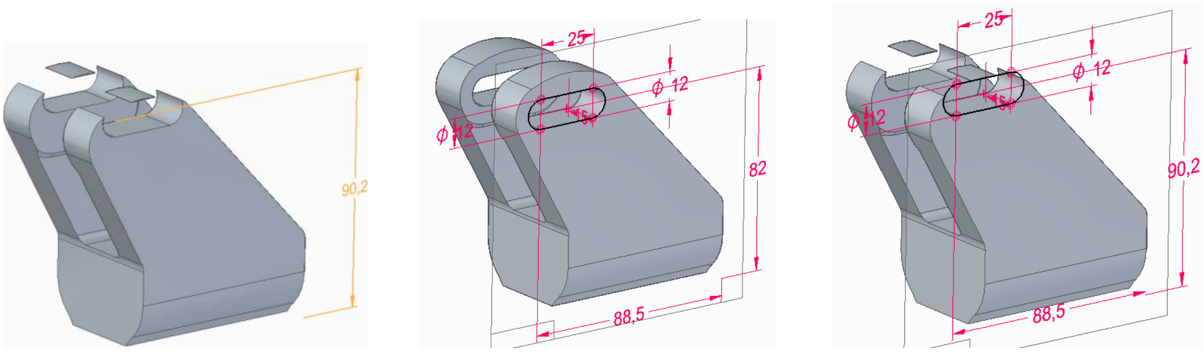


(a) (b) (c)

Figure 8: Example of a CAD Model B type I dormant deficiency. From left to right: (a) regenerated CAD model containing failed features and geometric deficiencies in the form of incomplete shape, (b) state of the CAD model before the alteration of a critical dimension, (c) state of the regenerated CAD model after the alteration of a critical dimension.

In the example shown in Figure 8(a), a change in the dimension defining the radius of the U-shaped pipe support, modeled with the feature named ring protrusion, caused the re-creation of the feature named ring support to fail, due to a non-manifold geometry issue (see Figure 8(c)). This deficiency is related to an error in the profile definition of the ring support, which, although it is fully constrained, employs a fixed dimension in its radius definition, as shown in Figure 8(b). When the radius of the U-shaped pipe support (the ring protrusion feature) is increased, this will create a geometry with two single contact points (see again Figure 8(c)), resulting in a non-manifold geometry. Note that the situation can easily be recovered by replacing the fixed radius dimension in the profile of the ring support with a geometric constraint that creates an inter-feature dependency between the ring support feature and the ring protrusion feature based on equal curvature within their spatial contact zone.

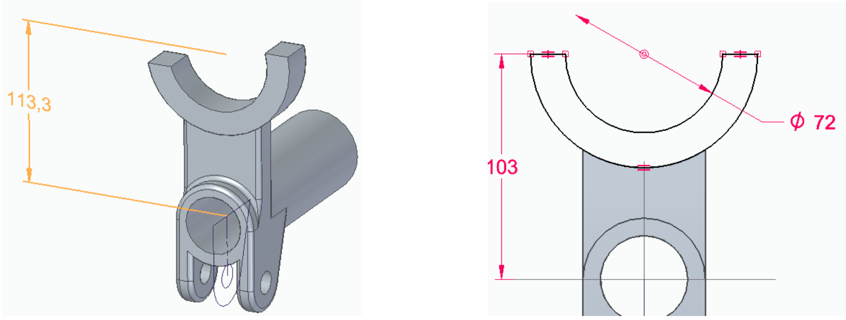
A concrete example of a type II dormant deficiency that was detected using the software tool and also confirmed by human assessment is shown in Figure 9(a), and is further discussed below. As type II dormant deficiencies do not directly affect the status of features, resulting only in geometric deficiencies, the CAD system is unable to detect any error in the CAD model. However, when the software tool is applied, the effects of those dormant deficiencies can be made visible during the alteration simulation. For example, increasing one dimension relating to the height of the extruded cutout feature used to model the cutout of the link yoke head from a value of 82 mm, as shown in Figure 9(b), to a value of 90.2 mm as shown in Figure 9(c), results in a regenerated CAD model that contains unexpected as well as unintended outcomes in the form of incoherent geometry, and thus a deficient shape. Closer examination of this dormant deficiency reveals that the cutout feature profile was linked directly to this dimension, which was a fixed value. In such a modeling situation, to prevent this dimension from becoming critical and thus introducing a dormant deficiency, the feature profile should have been positioned in respect to some key point of the yoke geometry and not explicitly linked to a dimension with a fixed value.



(a) (b) (c)

Figure 9: Example of a CAD Model A type II dormant deficiency. From left to right: (a) regenerated CAD model containing geometric deficiencies in the form of incoherent shape, (b) state of the CAD model before the alteration of a critical dimension, (c) state of the regenerated CAD model after the alteration of a critical dimension.

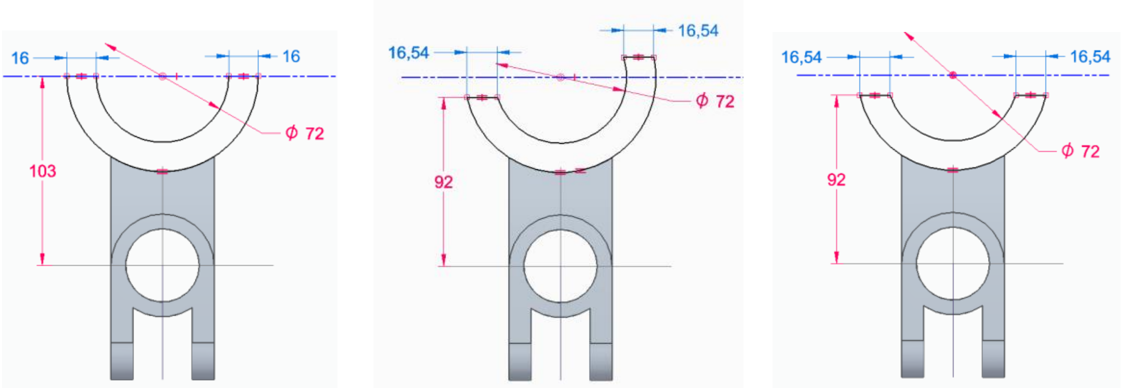
It will be recalled that, according to the assessment results automatically generated by the software tool, 18 CAD models (see again Table 3) were found to be free of any errors, including dormant deficiencies. However, this result also needs to be verified. Another independent manual analysis was conducted by the authors. Through this manual human-based analysis it could be confirmed that indeed 17 of the CAD models were without any defect in regard to the metrics and assessment criteria applied. However, one CAD model within the CAD Model B set was found to have a type III dormant deficiency, as shown in Figure 10(a). Note that, although this type of deficiency is not explicitly included in the automatically generated and compiled basic CAD model analysis reports of the software tool, it can easily be detected using visual feedback within the VSD during the tool-assisted CAD model assessment.



(a) (b)

Figure 10: Concrete example of a CAD Model B type III dormant deficiency. From left to right: (a) regenerated CAD model with a shape inconsistent with the design requirements, (b) state of the CAD model before the alteration of a dimension.

Detailed analysis of this CAD model revealed several issues, as follows. This dormant deficiency needs to be seen in an intra-feature context and relates to its dependency on how the profile of the extrusion feature used to create the U-shaped pipe support has been defined. This profile was found to have constraints as follows. The larger circle must have the same diameter as the U-shaped pipe support, indicating an inter-feature constraint that has been defined in the proper way. Next, the two circles are set to be concentric with the diameter of the inner circle driven by the critical dimension (see Figure 10(b)). Additionally, the two segments connecting the circles are defined to be both equal in length and horizontal. Note that this dimension also defines the distance between the left segment and the horizontal reference plane. This results in a fully constrained profile.



(a) (b) (c)

Figure 11: Example of various CAD model states in regard to a type III dormant deficiency. From left to right: (a) state of the CAD model before the alteration of a critical dimension, (b) actual state of the regenerated CAD model after the alteration of a critical dimension, (c) intended state of the regenerated CAD model after alteration that is consistent with design requirements.

However, if we alter the value of this dimension associated with the left segment (see Figure 11(a)), the new solution is not unique. This is due to the fact that two solutions can be generated, both with segments having the same length and located symmetrically in respect to the horizontal axis that passes through the center of the circles, as shown in Figure 11(b) and Figure 11(c). Note that both the horizontal axis and the dimensions (both in blue) of the individual segments have been added to annotate the example.

4.3 Formative Usability Assessment

Usability testing of the software tool during development was conducted with 15 volunteer students, subdivided into two groups, one group of 10 and one group of 5 students (see also [20,29,30]). The first group was comprised of 2 first-year male graduate students and 8 undergraduate students (25% female). The second group was comprised of male undergraduate students only. The undergraduate students were sophomores just beginning the second year of university studies. Since the tests and surveys in these small formative usability studies were not conducted in an anonymous manner, and to reduce any bias in the student user feedback, the only students who were allowed to participate were those who had finished the course and completed the final exams for the previous academic year.

Testing of the software tool was carried out by asking the students to perform tasks that were real and meaningful to them, that is the testing and assessing of CAD models created by novices. These tasks were embedded within scenarios, which were realistic descriptions of the CAD modeling exercises in the current CAD course. In case students wanted to test the tool on CAD models other than their own, CAD models similar in complexity to those in the course exercises were provided. Data from the test users were collected through a formal channel using questionnaires within a survey, and through an informal channel using electronic mail to exchange notes containing additional comments and suggestions. The student users were asked to rank on a five-point Likert scale (strongly agree, agree, neither agree nor disagree, disagree, or strongly disagree) the following statements.

1. I found the software tool useful for understanding where deficiencies are located.

2. I found the software tool useful for understanding which are critical situations.

3. I found the software tool useful for understanding how to correctly constrain a feature.

4. I found the software tool useful for improving my modeling capabilities.

5. I found the software tool useful for improving my planning of correct modeling strategies.

6. I found the software tool easy to use.

7. I found the material and information provided on how to use the software tool sufficient.

In addition, students were given the opportunity to provide general comments and particular suggestions on how to improve the software tool, and which additional information or feedback they would like it to provide. Students also had to indicate which CAD models they had used during the tests, that is only their own, the ones optionally provided to them for the test tasks, or both.

For the usability test tasks, the majority (80%) of the students used either the CAD models provided and their own CAD models, or only their own. Only 3 students decided not to use their own CAD models. This, among other factors, clearly indicates that there was a genuine interest and engagement in the test tasks and the experiences to be obtained. The software tool, its functionality, interface design and perceptible affordances, and the user experiences during the test tasks were exceptionally well received by all student users. The answers to the survey did not contain any negative responses, as shown in Figure 12. Also, the known tendency that respondents are sometimes reluctant to express strong opinions, and thus are likely to distort the results by gravitating to the neutral midpoint response, was not observed. Results of particular questions were as follows.

The results of ranking statements 1, 2, and 3, concerning the usefulness of the tool in supporting understanding, yielded a median of 5 on a five-point scale for each statement. In particular, for statement 1, the number of students who strongly agreed was four times higher than those who agreed. In the case of statement 2, this relationship was 2:1. In the case of statement 3, the number of students who strongly agreed was about the same as the number who agreed (see again Figure 12). This response data clearly indicates that the software tool is capable of being put to use as designed. It is interesting to note here that the strongest user confirmations were expressed within the ratings for statement 1 and statement 2, which correctly reflects some of the main objectives of the software tool as a central supportive resource, namely enabling and supporting the location and analysis of deficiencies and critical situations. Further aspects of the usefulness of the software tool in regard to improving CAD modeling capabilities and the planning of correct modeling strategies also received encouraging positive ratings, with a median of 4 for both statement 4 and statement 5. For question 6, almost all (over 93%) of the students agreed or strongly agreed that the software tool was easy to use, with a median of 5. This student user response data further confirms that the software tool is not only serviceable for the purpose for which it was designed, but also easy to use. Hence, students can use their mental capacities and energy to better focus on the tasks at hand, that is CAD model assessment, and to reflect on, as well as learn from, errors. Responses for question 7 also yielded a median of 5, though analysis of the open-ended questions and the comment section of the survey revealed that students had particular ideas for improving the material on how to use the software tool. These, and other suggested improvements, are further detailed below in the qualitative analysis of the survey data.

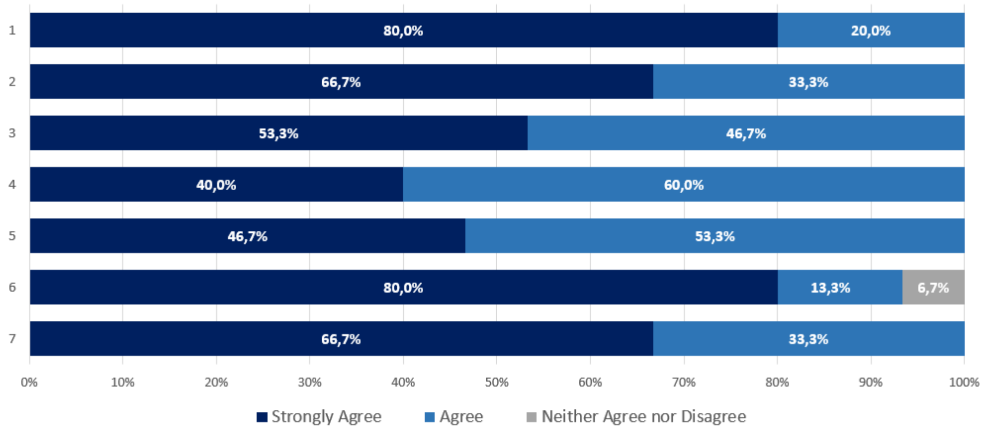


Figure 12: Graphical representation of proportions of selected answers from the student survey.

Qualitative analysis of the student user comments and the open-ended survey questions on how to improve the software tool, and which additional information or feedback the tool should provide, led to two themes, namely training material improvements and software tool improvements. The former were mostly suggestions in regard to the online tutorial that is provided with the software tool. Suggestions such as the inclusion of more than one CAD model in the tutorial examples were immediately translated into editing of the online tutorial, so that it now employs various CAD models. Suggestions regarding software tool improvements could be grouped into three categories as follows.

Firstly, it was suggested that the application domain should be extended. Students suggested that it should also be possible to use the software tool in the domain of direct modeling and CAD model assemblies. In the case of the former, as the data structure and functionality of the modeling domain are quite different from their counterparts in parametric feature-based modeling, this would require a complete re-design and re-development of the software tool. In case of the latter, from a technical point of view, the current software tool domain based on part models could be extended to include assemblies. The authors will look into this issue and then decide if the efforts required are justified by the benefits in regard to the current CAD course exercises that are relevant for assembly modeling.

Secondly, it was suggested that the information and feedback provided should be increased. Student users suggested that it would be helpful to have more information provided by the software tool on how to correct the errors once they have been located and identified in the CAD model. This issue of CAD model assessment tools also providing feedback on how to design error recovery strategies is a non-trivial problem that transcends the educational context. From a technical perspective, any solution in this direction requires a considerable amount of codified CAD domain knowledge, which is usually not available in current CAD model assessment and grading tools regarding the context of CAD model robustness and alterability. From an educational perspective, those tools should help in finding errors in CAD models, in order to help and support students in reflecting on and learning from mistakes. Hence, developing knowledge and understanding about what not to do and how to avoid as well as recover from mistakes are elements of skill and competency development that are not supposed to be short-circuited by software tools that also automatically suggest how to correct any errors that students have committed during CAD model creation.

Thirdly, it was suggested that the status quo should be respected and maintained. Several of the student users seem to be content with the software tool and its interface as they are, and so they wanted to continue with the existing state of affairs. This was expressed in comments such as "keep it simple as it is", and "great work and good as it is". These responses, though, do not come as a total surprise after analysis of the survey data. As was shown earlier, most student users were quite satisfied with what the software tool does, how it does it, and the feedback it provides. Hence, from their perspective as users, they did not want a high-quality user experience to be adversely affected by changes that might not offer much added value. This, however, does not mean that there is no need for reassessment, or room for improvement.

As can be expected from small usability test studies, the formative usability assessment was effective in providing a first insight into and some understanding of the concrete expectations and actual engagement of student users with the software tool. The study also helped to reveal what students like and what seems to work best for them during tool-assisted CAD model analysis, reflection on errors, and self-assessment. In this regard the authors are not planning to make any profound changes to the software tool and its user interface before more data and information from metrics, statistics, and qualitative assessment are available to guide informed decisions on tool and learning experience design improvements. This will be enabled through summative usability assessment. This larger study, which is a part of the second step within the two-step project, requires a large number of users, and thus will involve, on a voluntary basis, the entire student cohort from the current CAD course of the academic year that has just started.

5 CONCLUSIONS AND FUTURE WORK

This paper reports on efforts made and results achieved within the first step of a two-step project to improve the current quality and outcomes of CAD education for mechanical engineering at institutions of higher education. In particular, it aims to support the self-assessment and self-adjustment efforts of students, and to offer a better way of making sense of experience and domain knowledge while practicing the design and creation of robust, alterable parametric feature-based solid models. This has been approached by a re-design of the learning experience, taking into account means dedicated to assisting students in self-assessment during CAD modeling exercises. This has led to the development of the novel concept of dormant deficiencies, a key metric for CAD model assessment and for student core behavior evaluation. To support putting into practice this learning-experience-oriented approach and the key metric required, a new software tool has been designed and developed specifically to match the needs of students in regard to assisting their self-assessment efforts. Due to the requirement specification and the novel key metric, the framework and structures designed and used for the development of the software tool are considerably different from usual CAD model grading tools. The former is more dynamic, process-oriented, and simulation-based, while the latter is more static, object-oriented, and with an application focus on the final outcome, that is the finalized CAD model.

The functionality, performance, and reliability of the current implementation of the software tool have been successfully tested and evaluated using CAD models that had been submitted by students as results of CAD laboratory exercises and course assignments administered within a recently reformed CAD course in mechanical engineering, which is offered by the department where the authors operate. The software tool performed without any flaws or malfunctions. It was able to detect all dormant deficiencies in each CAD model subject to the analysis test. The software tool was also able to correctly reveal a type III dormant deficiency, which, however, is beyond the educational context of the current CAD course for freshmen. However, the authors are pondering the possibility and benefits of employing this software tool in CAD exercises related to another engineering course aimed at graduate students. The usability of the software tool and its interface design have been evaluated during development through formative usability studies, which also included surveys. The structure and questionnaires employed in the usability studies and surveys, and the feedback and results obtained, are also reported in this paper. Results of the formative usability test not only helped to reveal what student users expect and like in regard to the software tool, but also what works best for them. This has led, among other revisions, to the modification of the online tutorial to better accommodate the needs of the students.

In regard to the second step of the two-step project, currently preparations are under way to fully integrate the newly developed key metric and to deploy the student software tool within the CAD course in this coming academic year. These efforts will include a systematic introduction of the key metric and the software tool during the course lessons, a multimedia-based online tutorial, and an online survey based on questionnaires. The questionnaires are structured so as to gain insight based on feedback from the students' perspectives on what worked and what did not. They will also indicate any further learner needs, which have been overlooked and give guidance as to how the tool can be improved and its use expanded, and how it can be introduced smoothly into the CAD course. The questionnaires in the revised survey structure are based on those used previously in the formative usability assessment, but are extended to capture additional elements. For example, they ask questions related to student user experience and satisfaction. It is hoped that, through this summative study, which involves a large number of student users, feedback and results can be used to produce metrics, statistics, and qualitative information that give a more detailed understanding of and insight into the issues of functionality, usability, and user experience, and how these are related to actual improvement of the exercise learning experience, as well as achieving the outcomes at which the course is aimed.

Ferruccio Mandorli, http://orcid.org/0000-0003-4864-5265
Harald E. Otto, http://orcid.org/0000-0002-4580-0429

REFERENCES

[1] Aubin, J.-P.; Frankowska, H.: Set-Valued Analysis, Birkhäuser, Boston, MA, USA, 2009.
[2] Ault, H. K.; Fraser, A.: A comparison of manual vs. online grading for solid models, in: Proceedings of the 120th ASEE Annual Conference and Exposition, June 23-26, Atlanta, GA, USA, 2013, Paper-No.: 7233.
[3] Ault, H. K.; Linjun, B.: Solid modeling strategies – analyzing student choices, in: Proceedings of the 121th ASEE Annual Conference and Exposition, June 15-18, Indianapolis, IN, USA, 2014, Paper-No.: 9242.
[4] Barbero B. R.; Pedrosa, C. M.; Samperio, R. Z.: Learning CAD at university through summaries of the rules of design intent, International Journal of Technology and Design Education, 27(3), 2017, 481-498. https://doi.org/10.1007/s10798-016-9358-z
[5] Barnum, C. M.: Usability Testing Essentials, Morgan Kaufmann, Cambridge, MA, USA, 2020.
[6] Branch, R. M.: Instructional Design: The ADDIE Approach, Springer, New York, NY, USA, 2009.
[7] Beineke, L.W.: Families of Graphs and Digraphs, in: Gross, J. L., Yellen, J., Zhang, P. (eds.), Handbook of Graph Theory, CRC Press Boca Raton, FL, USA, 2014, 21-30.
[8] Bryan, J. A.: Automatic grading software for 2D CAD files, Computer Applications in Engineering Education, 28(1), 2020, 51–61. https://doi.org/10.1002/cae.22174
[9] Company, P.; Contero, M.; Otey, J.; Plumed, R.: Approach or developing coordinated rubrics to convey quality criteria in MCAD training, Computer Aided Design, 63, 2015, 101–117. https://doi.org/10.1016/j.cad.2014.10.001
[10] Diestel, R.: Graph Theory, Springer, Berlin, Germany, 2017.
[11] Dumas, J. S.; Redish, J. C.: A Practical Guide to Usability Testing, Intellect, Portland, OR, USA, 1999.
[12] Garland, A. P.; Grigg, S. J.: Evaluation of humans and software for grading in an engineering 3D CAD course, in: Proceedings of the 126th ASEE Annual Conference and Exposition, June 16-19, Tampa, FL, USA, 2019, Paper-No.: 26525.
[13] Garrett, J. J.: The Elements of User Experience, Peachpit (Pearson Education Division), Berkeley, CA, USA, 2011.
[14] Gonzáles-Lluch, C.; Company, P.; Contero, M.; Camba, J. D.; Colom, J.: 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), 2017, 1643 – 1653.
[15] Gonzáles-Lluch, C.; Company, P.; Contero, M.; Pérez-López, D.; Camba, J. D.: On the effects of the fix geometric constraint in 2D profiles on the reusability of parametric 3D CAD models, International Journal of Technology and Design Education, 29, 2019, 821 - 841. https://doi.org/10.1007/s10798-018-9458-z
[16] Gross, J. L.; Yellen, J.; Anderson, M.: Graph Theory and its Applications, CRC Press / Taylor and Francis Group, Boca Raton, FL, USA, 2019.
[17] Harrower, M., Brewer, C. A. (2003). ColorBrewer.org: An online tool for selecting colour schemes for maps, The Cartographic Journal, 40(1), 2003, 27-37. https://doi.org/10.1179/000870403235002042
[18] Hattie, J.; Timperley, H.: The power of feedback, Review of Educational Research, 77(1), 2007, 81-112. https://doi.org/10.3102/003465430298487
[19] Hekman, K. A.; Gordon, M. T.: Automated grading of first year student CAD work, ASEE Computers in Education Journal, 24(2), 2014, 16-24.
[20] Hwang, W.; Salvendi, G.: Number of people required for usability evaluation: The 10±2 rule, Communications of The ACM, 53 (5), 2010, 130-133. https://doi.org/10.1145/1735223.1735255
[21] Ingale, S.; Anirudh, S.; Bairaktarova, D.: CAD platform independent software for automatic grading of technical drawings, in: ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, August 6 – 9, Cleveland, OH, USA, 2017, Paper-No.: DETC2017-67612. https://doi.org/10.1115/DETC2017-67612

[22] Irons, A.: Enhancing Learning through Formative Assessment and Feedback, Routledge, Abingdon, UK, 2008.
[23] ISO 9241-11:2018: Ergonomics of Human-System Interaction – Part 11: Usability: Definitions and Concepts, ISO/TC159/SC4, 2018.
[24] Jaakma, K.; Kiviluoma, P.: Auto-assessment tools for mechanical computer aided design education, Heliyon, 5(10), 2019, e02622. https://doi.org/10.1016/j.heliyon.2019.e02622
[25] Jenny, B.; Kelso, N. V.: Color Design for the color vision impaired, Cartographic Perspectives, 58, 2007, 61-67. https://doi.org/10.14714/CP58270
[26] Johnson, J.: Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines, Morgan Kaufmann, Waltham, MA, USA, 2020.
[27] Kirstukas, S. J.: Development and evaluation of a computer program to assess student CAD models, in: Proceedings of the 123rd ASEE Annual Conference and Exposition, June 26-29, New Orleans, LA, USA, 2016, Paper-No.: 15834.
[28] Kurosu, M.; Kashimura K.: Apparent usability vs. inherent usability: Experimental analysis on the determinants of the apparent usability, in: Proceedings of the ACM CHI Conference Companion on Human Factors in Computing Systems, May 7-11, Denver, CO, USA, 1995, 292-293. https://doi.org/10.1145/223355.223680
[29] Lewis, J. R.: Sample sizes for usability tests: Mostly math, not magic, Interactions, 13(6), 2006, 29-33. https://doi.org/10.1145/1167948.1167973
[30] Lewis, J. R.: Usability: Lessons learned … and yet to be learned, International Journal of Human-Computer Interaction, 30(9), 2014, 663-684. https://doi.org/10.1080/10447318.2014.930311
[31] Lidwell, W.; Holden, K; Butler, K.: Universal Design Principles, Rockport Publishers, Beverly, MA, USA, 2010.
[32] Mandorli, F.; Otto, H. E.: Systematic support of learning from errors and negative knowledge development in MCAD education: Empirical analysis of student feedback, Computer-Aided Design and Applications, 17(2), 2020, 384-406. https://doi.org/10.14733/cadaps.2020.384-406
[33] McCarthy, j.; Wright, P.: Putting 'felt-life' at the centre of human-computer interaction (HCI), Cognition, Technology & Work, 7, 2005, 266-271. https://doi.org/10.1007/s10111-005-0011-y
[34] McNamara, N.; Kirakowski, J.: Functionality, usability, and user experience: Three areas of concern, Interactions, 13(6), 2006, 26-28. https://doi.org/10.1145/1167948.1167972
[35] Molich, R.; Nielsen, J.: Improving a human-computer dialogue, Communications of the ACM, 33(3), 1990, 338-348. https://doi.org/10.1145/77481.77486
[36] Moyers, S.: The F pattern: Understanding how users scan content, UX Magazine, 7, 2017, Paper-No.: 1687.
[37] Munzner, T.: Visualization, Analysis and Design, CRC Press, Boca Raton, FL, USA, 2014.
[38] Otey, J.; Company, P.; Contero, M.; Camba, J. D.: Revisiting the design intent concept in the context of mechanical CAD education, Computer-Aided Design and Applications, 15(1), 2018, 47-60. https://doi.org/10.1080/16864360.2017.1353733
[39] Otto, H. E.; Mandorli, F.: A framework for negative knowledge to support hybrid geometric modeling education for product engineering, Journal of Computational Design and Engineering, 5, 2018, 80-93. https://doi.org/10.1016/j.jcde.2017.11.006
[40] Otto, H. E.; Mandorli, F.: Parametric feature-based solid model deficiency identification to support learning outcomes assessment in CAD education, Computer-Aided Design and Applications, 18(2), 2021, 411-442. https://doi.org/10.14733/cadaps.2021.411-442
[41] Pütz, C.: Teaching CAD with a pedagogical system ranging from videos to individual tutoring, Journal of Industrial Design and Engineering Graphics, 12 (1), 2017, 21–26.
[42] Schlatter, T.; Levinson, D.: Visual Usability: Principles and Practices for Designing Digital Applications, Morgan Kaufmann, Waltham, MA, USA, 2013.
[43] Schmidt, G.: Relational Mathematics, Cambridge University Press, New York, NY, USA, 2011.

[44] Sim, S. K.; Duffy, A. H. B.: A new perspective to design intent and design rationale, in: Workshop Notes for Representing and Using Design Rationale, Artificial Intelligence in Design, August 15-18, Lausanne, Switzerland, 1994, 4 – 12.

[45] Tractinsky, N.; Katz, A. S.; Ikar, D.: What is beautiful is usable, Interaction with Computers, 13(2), 2000, 127–145.

[46] Ware, C.: Visual Thinking for Design, Morgan Kaufmann, Burlington, MA, USA, 2008.

[47] Ware, C.: Information Visualization: Perception for Design, Elsevier, Cambridge, MA, USA, 2020.

[48] Wiggins, G.; McTighe J.: Understanding by Design, Association for Supervision and Curriculum Development (ASCD), Alexandria, VA, USA, 2005.

[49] Wong, Y. L.; Siu, K. W. M.: Assessing Computer-Aided Design Skills, in: Khosrow-Pour, M. (ed.), Encyclopedia of Information Science and Technology, IGI Global, Hershey, PA, USA, 2018, Volume 10, 7382-7391.

[50] Yablonski, J.: Laws of UX: Using Psychology to Design Better Products & Services, O'Reilly, Sebastopol, CA, USA, 2020.

[51] Yan, C.; Yu, B.; Hui, Y.; Zhou, F.: CAD education of college students with engineering practice and team work, EURASIA Journal of Mathematics, Science, and Technology Education, 13(7), 2017, 5569-5576. https://doi.org/10.12973/eurasia.2017.01010a

Appendix

This appendix provides an overview of the data flow between the CAD modeling environment and the CAD model alteration simulator shown in Figure A1, with the latter being a module within the software tool. In addition to this data flow overview operational details of some functional blocks of the software tool are provided in form of pseudocode as shown in Figure A2.

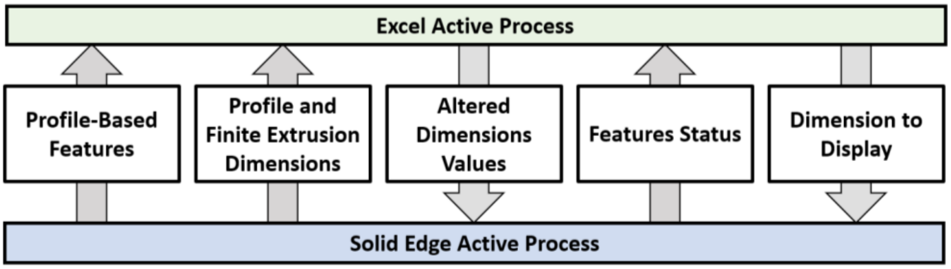


Figure A1: Overview of the data flow between the CAD modeling environment and the CAD model alteration simulator.

Run Test Procedure

```
# run test procedure
Sub RunTest()

set objFeatures = SolidEdge.FeaturesList
AlterationFactor = Excel.AlterationFactor

for each objFeature in objFeatures Do
    # retrieve feature dimensions from profile-based features
    if isProfileBased(objFeature) Then
        Dimensions = objFeature.Profile.Dimensions
        if objFeature.ExtrusionType = FiniteType Then
            Dimensions = Dimensions & objFeature.Extrusion.Value
    end if
end if

    # alter dimensions
    for each Dimension in Dimensions Do
        StartValue = Dimension.Value
        DeltaValue = StartValue * AlterationFactor
        #initial condition
        DeficientFeaturesInfo = SolidEdge.GetDeficientFeaturesInfo
        Excel.UpdateReport(DeficientFeaturesInfo)
        #increment dimension
        SolidEdge.SetDimension(Dimension, StartValue + DeltaValue)
        DeficientFeaturesInfo = SolidEdge.GetDeficientFeaturesInfo
        Excel.UpdateReport(DeficientFeaturesInfo)
        #reset initial condition
        SolidEdge.SetDimension(Dimension, StartValue)
        DeficientFeaturesInfo = SolidEdge.GetDeficientFeaturesInfo
        Excel.UpdateReport(DeficientFeaturesInfo)
        #decrement dimension
        SolidEdge.SetDimension(Dimension, StartValue - DeltaValue)
        DeficientFeaturesInfo = SolidEdge.GetDeficientFeaturesInfo
        Excel.UpdateReport(DeficientFeaturesInfo)
        #reset initial condition
        SolidEdge.SetDimension(Dimension, StartValue)
        DeficientFeaturesInfo = SolidEdge.GetDeficientFeaturesInfo
        Excel.UpdateReport(DeficientFeaturesInfo)
    next Dimension
next objFeature
```

Figure A2: Pseudocode-based description of some functional blocks of the software tool.