Surface Model Deficiency Identification to Support Learning Outcomes Assessment in CAD Education

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

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

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

ABSTRACT

Recent efforts to reform CAD education are aimed, among other things, at increasing the development of strategic knowledge and modeling skills within CAD competency. This requires better structured and more frequent assessment and feedback than can be achieved with current, mostly summative-based, techniques. Here, formative assessment and formative feedback appear to offer a viable solution. Unfortunately, within CAD education, dedicated techniques and tools are not yet available to support the implementation of formative assessment, and, in particular, to assist learning goal and outcome oriented assessment of CAD models produced by students. Moreover, those frameworks and tools for CAD model analysis and evaluation that are available and deployed within commercial and industrial settings cannot be directly used in educational settings, due to differences in assessment criteria and evaluation goal settings. The aim of the current paper is, firstly, to present a novel approach for surface model assessment in the educational context, which is based on deficiency analysis in relation to learning outcomes, and, secondly, to report on the implementation and application of a newly developed software tool to enable and put into practice this novel surface model assessment approach.

Keywords: formative feedback, reflection on performance and outcome, competency development, strategic knowledge built-up, geometric CAD model usability.
DOI: https://doi.org/10.14733/cadaps.2019.429-451

1 INTRODUCTION

Recent efforts to introduce changes in the way course curricula and teaching are designed and executed in education, particularly in science and engineering at institutions of higher education, appear to be influenced as well as driven by two principal forces. The first of these forces is changes in a progressively technology-influenced postmodern society with its complex global labor market, and the second force is the results of work in educational research and cognitive science on how students learn. This situation is intensified by concerns related to the increasing gap between student learning outcomes that are achieved with classic, though apparently outdated, teaching approaches in higher education and the vigorously rising demand for professionals with sophisticated skills and competencies in highly competitive markets. In the context of computer-aided design (CAD) and digital product modeling, this translates, according to trends and studies, into a focus on the development and implementation of restructured curricula and alternative teaching approaches that are more student centered and learning oriented, and thus are better structured to efficiently and effectively match actual student learning outcomes with skills and competencies related to, among others, spatial ability and mental visualization, cognitive model composition, meta-cognitive processes including planning, predicting, and revision, and modeling strategies (see also [8,40]). Within the context of CAD education for mechanical engineering (MCAD), assessing CAD models is quite a delicate and highly time-consuming activity, which requires, among many other competencies, the ability to discriminate efficiently between trivial errors, i.e. errors that have been committed by students due to carelessness and inadvertence while performing the exercise, and more serious errors, i.e. errors that have occurred due to a lack of knowledge and understanding of the domain subject. These prerequisites are too often overlooked in current efforts to restructure MCAD education, which involves integrating an increasing number of CAD laboratory exercises and practical assignments with course lectures, in an attempt to advance the development of CAD modeling skills and expertise. Although representing a well-intended educational goal, this implementation will be neither effective nor efficient if it lacks adequate provision of formative feedback.

However, the ability to face these challenges adequately is impeded by the increasing spread of mass education and the soaring relevance of geometric modeling and CAD skills and competency for graduates from technical and engineering institutions of higher education (cf. [1,11,18]), thus pushing CAD courses into the basic education programs of many faculties. This represents an unfortunate situation, which results, in many cases, in demands to accommodate cohorts of 100 or more students within one CAD course during one semester or in a single academic year. In such course settings, the provision of any form of feedback, in particular formative feedback, in regard to student performance, and the achievement of learning goals and outcomes, is very limited, especially at an individual and personalized level, where it is considered to be most effective. These circumstances are especially grave when it comes to feedback provision in regard to CAD modeling exercises and CAD laboratory performance, which necessitates the individual assessment of CAD models created by students, thus requiring time and personnel resources that in many cases within present-day educational practice do not match current capacities of faculty, equipment or tools.

2 BACKGROUND, SCOPE, AND OBJECTIVES

2.1 Background and Related Work

Challenges, some of which have been made explicit and briefly discussed elsewhere in this paper, which are currently faced within CAD education, and efforts to restructure its curricula, while also trying to benefit from recent developments in improved alternative teaching approaches (see also literature review and discussions in [1,12,29,43,46]), have been addressed and tackled within discipline-based educational research from several directions.

A recent approach to transforming and advancing adaptive expertise development in CAD education by integrating contextual exercises was presented in [24]. A study on the transfer of learning between 3D modeling systems is reported in [41], and provides results with implications for the design of educational programs in regard to delineating between system dependent and system independent skills and knowledge. Efforts to address improvements in pedagogical approaches for 3D CAD are presented in [10,11], and these address CAD expertise development by supporting strategic knowledge development and the improvement of spatial ability. In particular, to support the latter, integration of a number of strategies has been proposed. These include pre-exposure to perceptual differentiation, experience with manipulative tasks, and the use of sketching. Work on a theoretical framework and heuristics for best practice is introduced in [36], addressing issues of developing both the capacity to generate cognitive models and the ability to decompose geometric elements, leading to better cognitive handling of modeling concepts as well as achieving design intent in the context of parametric modeling system oriented CAD pedagogy.

To translate the potential and benefit of those encouraging approaches into educational practice, however, also requires better structured and more frequent assessment and feedback than can be achieved with traditionally employed summative assessment and feedback techniques. Here, formative assessment (cf. [7,21,35,38]) and formative feedback (cf. [9,19,39]) appear to offer a viable solution (see also [28]), and are increasingly regarded as promising and effective components within instructional practices currently proposed for reforming higher education in science and engineering. Unfortunately, within CAD education, dedicated techniques and tools to support the implementation of formative assessment are not available yet, in particular to assist learning goal and outcome oriented assessment of CAD models produced by students. Moreover, those frameworks and tools for CAD model analysis and evaluation that are available and deployed within commercial and industrial settings cannot be directly used in educational settings. This is due to differences in assessment criteria and evaluation goal settings, focusing mostly on issues related to application context, quality, and interoperability of three-dimensional CAD models created with parametric history-based solid modeling systems (cf. discussions and tool reviews in [2,5,15,17,25,44,45]). Additional work on the assessment of CAD model surface errors and quality has been reported in [3,23].

2.2 Scope and Objectives

Recent efforts to reform an actual CAD course, which is currently a part of the curriculum for the Laurea degree in mechanical engineering at the institution represented by the authors, addressed, among other issues, the development of modeling competencies with particular reference to the development of the strategic knowledge required for creating usable CAD models in the field of hybrid geometric modeling (cf. [32]). In particular, this major course-specific learning goal, i.e., the development of the strategic knowledge and modeling skills indispensable for producing usable CAD models, requires better teaching techniques that reach beyond the usual lecture-based presentation of domain specific factual knowledge, with students mostly in the role of passive learners. However, notably, it also requires assessment techniques and feedback which are capable of adequately and frequently measuring the gap between actual student learning outcomes as achieved and learning goals as pre-assigned, while also informing both teacher and students of the results in a high-quality and timely manner.

As the concept of a *usable* model is highly context-dependent, it can be approached from different dimensions and levels of abstraction. Within the work presented in this paper, three hierarchically structured levels related to current computer-aided product development processes have been identified, namely the geometric level, the analysis level, and the functional level. At the geometric level, a model is considered usable if it does not contain any severe geometric defects or spatial anomalies which could impede the role of the model of being used in further steps of the modeling process. For example, the shape of a model is considered usable at the geometric level if its geometry is free of geometric deficiencies such as self-intersecting surfaces. At the analysis level, a model is considered usable if it meets all the requirements necessary to perform a particular model analysis. For example, a model can be considered usable when its shape is sound and structured so as to allow for the conducting of a finite element mesh analysis or a computer-aided engineering analysis. At the functional level, a model is considered usable if it meets all the requirements for the manufacturability, assemblability, and functioning of an individual component or assembly that its geometric representation was designed for and implemented. For example, the shape of a model is considered usable at the functional level if it allows for injection molding production. For any model to be considered usable at a particular level, a necessary pre-condition is that it is considered usable at the geometric level. Due to the fact that the CAD course, at present, is provided mostly to students who are novices in both geometric modeling and engineering, issues of model usability are currently approached from within spatial composition and shape, namely at the geometric level.

Previous studies of the authors, described for example, in [27,32,33], reported on the theoretical foundation and development of concepts and an integrated framework, which were employed in the design and introduction of a novel teaching approach. This approach integrates negative knowledge and expertise as crucial elements, with traditional teaching methods in combination with methods of formative assessment/feedback, to support CAD competency development. Within this approach, and with educational settings as outlined above, the assessment of student performance and results relating to CAD laboratory exercises and course assignments needs to be conducted in a computer-aided manner. This will facilitate actual implementation and improvement of the scope and overall quality of formative assessment and feedback, but requires new approaches and tools for surface model assessment. In this regard, the focus and aim of the current paper are as follows: firstly, to present a novel approach for surface model assessment in the educational context, which is based on deficiency analysis in relation to learning goals and outcomes; and, secondly, to report on the technical architecture and concrete implementation, as well as the application, of a newly developed software tool to enable and put into practice this novel surface model assessment approach.

3 APPROACH, FRAMEWORK, AND IMPLEMENTATION

3.1 Outline and Approach

As discussed elsewhere in this paper, analysis and assessment of CAD models within the context of education are different from their counterparts in commercial and industrial settings in regard to goal and assessment criteria definitions, and thus to the approach taken. This is most evident within formative assessment, while also resonating within summative assessment. To support formative feedback in education, CAD model assessment needs to consider the quality of a model not only in terms of the absolute criteria that are associated with technical domain knowledge, but also in terms of criteria relating to model deficiencies that are the result of wrong or inappropriately applied modeling strategies. This represents a task that is far from trivial, as it requires not only the detection and identification of deficiencies that in many cases do not violate general normative knowledge about geometric modeling (see also discussions on realism errors in [17]), but also knowledge about the modeling goals and how they have been translated into actions. Within an educational context, parts of the latter can usually be associated with learning goals and outcomes related to particular exercises and course assignments (see also structural outline in Fig. 1).

However, in the context of surface model assessment, analysis and evaluation are based mainly on the topology and geometry of the final modeling result, while using characteristics of individual curves and patches which were created for producing the final model shape, as a proxy for assessing particular modeling steps in a reflective and ex post facto manner. Currently, most commercially available CAD systems provide interactive commands at the user interface to allow for some basic form of inquiry about model properties and characteristics of geometric model entities such as model closure and curvature graphs. However, performing a purely manual surface model assessment by using such kinds of generic system command is in many cases a sensitive task, which can devolve into quite a convoluted and time-consuming effort. Moreover, only one model can be analyzed at a time. Hence, there is a risk of putting in place different sets of assessments on individual models which were actually created for one and the same exercise or course assignment, and thus, in fact, relate to the same set of learning goals and outcomes.

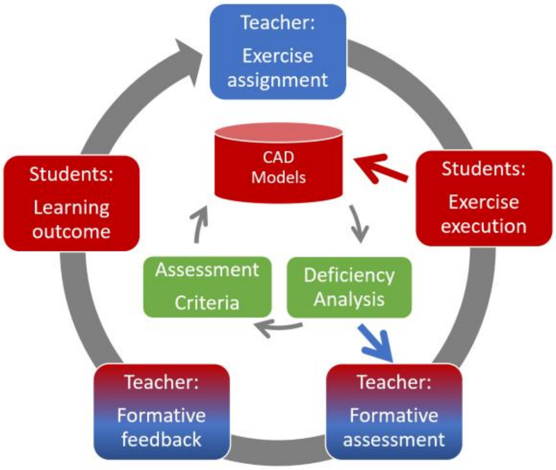


Figure 1: Structural outline of the integration of computer-aided deficiency analysis with formative assessment and formative feedback.

3.2 Framework and Software Tool Development

In order to support surface model assessment while avoiding shortcomings as outlined, a semi-automatic software tool has been developed that operates tasks in four process stages, namely compilation and export, import and filtering, enquiry and analysis, and visual analytics and assessment, as follows:

All surface models that have been created by students are compiled and stored in a repository. This repository is structurally subdivided into sets of different folders, with one set of folders for each exercise or course assignment. During the compilation process, information on geometric entities and their related meaningful characteristics, such as entity degree, number of control points, and curvature radius, is extracted from the surface models, codified, and stored in the form of text files, with one file for each model. Data about the geometric model entities and their characteristics, which are stored in the model repository, are processed and imported into a CAD model inventory. This CAD model inventory features a lattice-based data structure, which is structurally organized as various linked entity tables. Data compiled from CAD models associated with a particular exercise or course assignment are assigned to one particular cluster of entity tables. Note that table entries for each geometric entity in the model repository contain a unique entity identifier which is also used internally by the geometric modeling system. This facilitates a backtrack mechanism that is used to support human-based visual analytics and assessment of entities within the original data source, namely the CAD models in the modeling environment.

To enable a computer-aided search and identification of deficiencies in surface-based CAD models, filter functions that are associated with assessment criteria are provided at the user interface of the software tool. Those functions operate directly on the data of geometric entities and their characteristics, which were previously compiled and stored in the inventory. The assessment criteria which are employed are related to the expected learning goals and outcomes of the individual exercises and course assignments. Final overall assessment, which still requires human intervention and expertise, is supported by providing backtracking functionality for entities along with the model entity analysis results that were obtained in the previous task. Each entity in question, and most importantly those found by the software tool to be deficient, can be located in the original CAD model and made visible for further inspection and assessment by a human expert such as a course instructor.

3.3 System Structures and Implementation

3.3.1 Technical components and architecture

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 Fig. 2. The overall software tool design is based on a modular open system structure (MOSS), which operates through the CAD modeling environment, the CAD model geometric entity (CMGE) repository, and the CAD model geometric entity inventory. Within the current implementation, a commercially available non-uniform rational B-spline (NURBS) based surface modeling system in the mid-range was adopted as the modeling environment, namely a *Rhinoceros 3D* from Robert McNeel & Associates. The import/export modules, as depicted in Fig. 2, are implemented within the CAD modeling environment as a set of Python Scripts.

The CMGE repository includes both the original CAD model files, as created by the students, and the text files containing the actual CMGE data that were extracted from the CAD models. Those CMGE data are organized and stored within individual folders, with one folder for each assigned exercise, as indicated by the folder icons denoted by Ex-j shown in Fig. 2. The CMGE inventory has been implemented by using the Microsoft Access database system. Access macros are used to import the CMGE text files and to create the related Access-based tables (denoted by T_Ex-k in Fig. 2), where one table is created for each exercise type, containing the entire set of geometric entities extracted from all the models created by students for a specific assignment. The assessment criteria that were used for the CAD model deficiency analysis are specified and implemented using Access queries (denoted by Q_Ex-jk in Fig. 2).

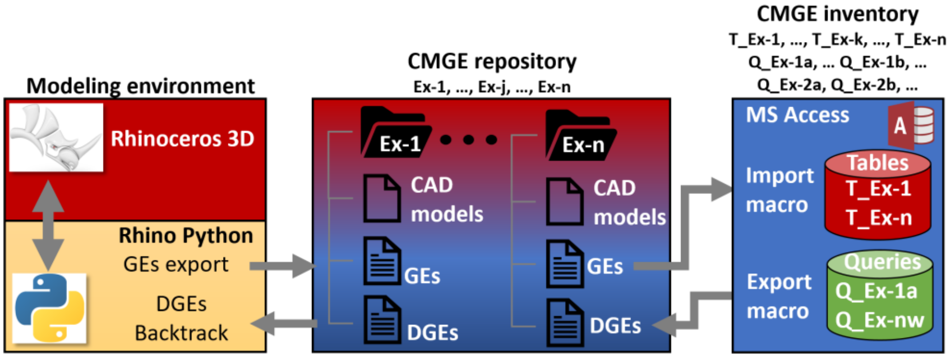


Figure 2: Overview of technical architecture of the software tool.

Those queries inspect and filter the tables of the CMGE inventory to detect and identify deficient geometric entities and the related CAD models within which they originally occurred. Export macros are used to export the unique CAD system-internal identifiers of all the deficient geometric entities detected during the analysis, to allow for backtracking of these deficient entities to the original CAD models.

The modular open system structure of the software tool as developed and implemented allows for an unobtrusive CAD model assessment that neither alters nor compromises in any way the original CAD models. Moreover, after the CMGE repository has been compiled, computer-aided assessment can be performed as many times as deemed necessary without essentially requiring actual access to the original CAD models subject to assessment. These features, among others, not only afford both interactive and batch processing, but also allow for distributed and shared model assessment through the provision of controlled remote access to the repository. Since the formulation of the model assessment criteria, as designed and used in the software tool, is independent of the hardware and software platform employed within the CAD modeling environment, the latest versions of CAD hardware and software platforms can be used for the benefit of education and course work, while avoiding any impact on the actual model assessment criteria as implemented. Modification of the latter is required only if changes in the course-specific learning goals become necessary, which would then also most likely propagate as changes required in the exercises and/or course assignments. However, should technical compatibility issues arise, due to the use of newly available CAD hardware and software platforms, modifications would be limited to the export/import module interfacing the CAD modeling environment with the CMGE repository.

3.3.2 CMGE repository, inventory, and filter functions

As mentioned earlier, the *Rhinoceros* system files of the CAD models, as created by the students, are grouped and stored in different individual folders, one for each exercise assignment. The batch-based module of the software tool that is operationalizing the export of geometric entities (see GEs export in Fig. 2) from within the surface-based CAD modeling system to provide data required as input for the creation of the CMGE repository, is implemented as a Rhino Python script that needs to run once for each assignment which is subject to CAD model analysis and assessment. This batch-based module operates in an iterative manner over all the CAD models contained within a folder and creates the CMGE repository related text files.

For each model contained in a folder, the batch procedure performs six basic computational steps as follows:

- opening and loading of the Rhino CAD model file.
- collecting and copying of all the curve and surface entities of the CAD model's individual separated layers to prevent any accidental editing and destruction of the original models.
- collecting and copying of all polycurves and polysurfaces of the CAD model and disaggregating them into single curve and surface entities to allow for the analysis of each single geometric entity, while preserving the original aggregated structure as created by the student.
- creation and opening of the CMGE repository associated text file, with the same name and in the same folder as the original Rhino CAD model file.
- transferring of all the copied geometric entities to the CMGE repository associated text file.
- saving and closing of the Rhino CAD model file.

The CMGE repository associated text file format is defined by 27 fields of variable length, within which each is separated by a semicolon symbol. A complete, more detailed specification of the fields of this text format is reported in the Appendix. The Access import macro is used for accessing and transferring the CMGE repository data related to each specific exercise assignment, and to compile and store all the geometric entities extracted from the individual CAD models related to an assignment in a single table-based lattice data structure. The CMGE inventory will then contain one table for each assignment to be assessed.

In regard to the design and specification of filter functions, criteria used for CAD model inspection are based on the identification of sets of geometric entities that meet specific characteristics, which in turn are related to specific values of particular fields in the table records of the CMGE inventory. Depending on the combination of filters applied, the returned set of entities can either directly contain the deficient entities as detected or contain entities that are required as input for further computation, such as set-based subtraction and intersection, to obtain the actual deficient geometric entities being sought after. For example, the query associated with the filter function for detecting closed polysurfaces is defined by filtering out those records in the CMGE inventory which have for the 'periodic' field a value equal to CLOSED_PS. In a similar manner, the query used for detecting a planar trimmed surface is defined by filtering out the records that have PLANAR as a value for the 'planar' field and TRIMMED as a value for the 'trim' field. A set-based subtraction (in this example the computation of the model set containing at least one closed polysurface and a non-planar trimmed surface), can be implemented by means of a LEFT JOIN query, having the minuend set as left table of the join and the subtrahend set as right table, while constraining the right records to be NULL. If a set of previously computed entities needs to be further filtered according to an additional condition, for example by imposing a minimum value for the surface area, then this can be implemented by adding a less than or equal to constraint (denoted by '<= value') to the 'area' field.

4 EMPRICAL RESULTS FROM EDUCATIONAL PRACTICE

4.1 Outline

Various modeling exercises are provided within the recently restructured MCAD course employing a novel teaching approach which systematically utilizes negative knowledge in addition to traditional lectures and tutorials. These modeling exercises are individually designed for various learning goals. 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, namely the modeling of a plastic mineral water bottle, and gradually increase in complexity of modeling task and object shape according to progress made in the course and the domain subject being taught. Modeling objects which are the subjects of the exercise series include USB thumb drives, trackball-like pointing devices, and enclosures for wall-mounted hand dryers and handheld hair blow dryers.

The current prototype implementation of the software tool has been successfully tested and validated using a set of 464 CAD models, which have been compiled into a CMGE repository of 33,189 geometric entities. These CAD models were submitted by students as results of CAD laboratory exercises and course assignments administered within a CAD course which was offered in the previous academic year by the department where the authors operate. The initial testing of the software tool and its application within the analysis and assessment of surface models covered all learning goal groups and related learning outcomes as stipulated for the course work. Validation of the software tool in regard to its accuracy and robustness in detecting CAD model deficiencies was carried out by human experts, who, in parallel with the application of the software tool, also performed a manual inspection for each individual CAD model used during testing. In what follows, a summarized overview consisting of three parts is presented on how learning goals and outcomes related to positive knowledge and negative knowledge are formed and used to design the CAD modeling exercise reference, which in turn is used for the specification and application of filter functions to detect CAD model deficiencies. Note that, due to limits regarding the length of the manuscript, presentation and discussion of selected material will be confined to the first exercise, which is related to modeling the plastic mineral water bottle, where students submitted a total of 118 CAD models, composed of 9,710 geometric entities.

4.2 Learning Goals and Outcomes

The main learning goal of the exercise presented and discussed as a representative example in this paper is to create a closed polysurface model of the subject of design, which is, for the exercise as outlined above, a plastic mineral water bottle. This surface-based 3D model needs to be good enough to enable it to be used within exercises that are scheduled later in the course, where the import of a surface-based solid model into a parametric feature-based CAD system and the subsequent modeling of simplified parts/components related to blow molding processes, such as blow molding cavities, are part of the exercise. Students are explicitly informed about the learning goals/outcomes, those requirements, and the fact that they have to use their own CAD models created during the mineral water bottle modeling exercise. Another educational reason for engaging students in 3D (solid) model creation during early exercises in surface modeling is related to the goal of domain-specific concept development. As soon as possible students should develop a knowledge of what a surface model and a solid model represent in a given context, as a proper understanding of these fundamental concepts is one prerequisite for the successful development of strategic knowledge and CAD competency throughout the CAD course.

Learning outcomes are subdivided according to the recently restructured course structure (cf. [32]) into two groups, namely learning outcomes related to positive knowledge and learning outcomes related to negative knowledge. Note that in the given context as outlined, negative knowledge is used as a form of knowledge about what is wrong and what is to be avoided during performance in certain situations. The course also builds on existing conceptions of negative knowledge as developed in research on expert systems, knowledge management, and professional learning, as well as expertise development. For an overview and more details, see also discussions in [16,30,31,34].

Learning outcomes related to positive knowledge

Students should be able to create a valid 3D closed polysurface, which can be converted into a solid model. To achieve this outcome, students must be capable of defining adjacent patches with at least G^0 continuity. This in turn requires the development of subject knowledge about the various characteristics of patches created by using different modeling commands and the variations of input parameters for the same modeling command. Students also need to be able to properly handle and keep under control various aspects related to minimum curvature radius of curves and surfaces used during modeling.

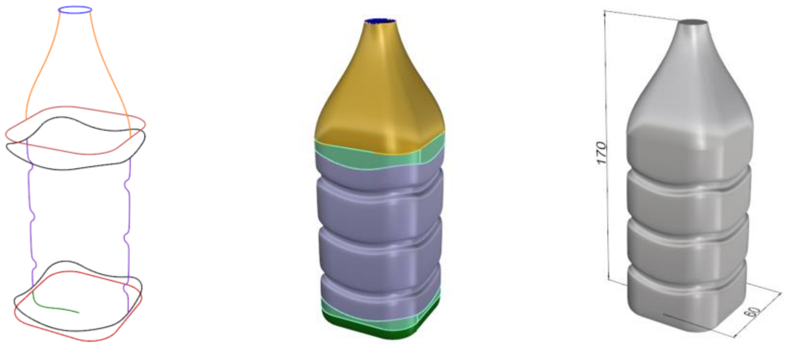
Learning outcomes related to negative knowledge

Students should have developed a capacity to recognize critical situations related to the various characteristics of the modeled patches. In particular, they should have mastered the art of recognizing and identifying deficiencies that may prevent adjacent patches of a surface-based 3D model from having at least G^0 continuity. For example, among the critical situations students should recognize are those which are most likely to introduce wrinkles and spikes into the model.

4.3 The CME Reference

The CAD modeling exercise (CME) reference consists of three basic components, namely the reference modeling approach, the reference CAD model, and the reference deficiencies. Those reference structures are used as a means of embodiment of information and knowledge about important facets of the basic goals, outcomes, and concepts and methods that are relevant for each individual exercise. The CME reference serves, among other things, as a backdrop or foundation for providing a reference for a domain and problem space. In the case of an ill-defined problem space (cf. [14,26]) such as computer-aided design, it is of considerable value for various purposes, especially for the assessment of produced outcomes.

The reference modeling approach is structured in a manner similar to a form of template solution, aimed at providing a means of affording a reference frame on know-how for correctly putting individual elements of strategic knowledge and modeling command application together. First, the modeling strategy requires the creation of the silhouette and section curves, and the layout used later for surface formation, as shown in Fig. 3(a). The layout is composed of the basic horizontal sections colored in red and black, the silhouette of the bottle body (colored in purple), the bottle neck (colored in orange), the bottle cap (colored in blue), and the bottle bottom (colored in green) section curves. Second, the curves layout previously created to outline the surface formation is then used to create the different patches that define the bottle shape with its main components, as shown in Fig. 3(b), namely the bottle body (colored in purple), the bottle neck (colored in light amber), the bottle transitions (colored in light green), the bottle bottom (colored in dark green), and the bottle cap (colored in blue). Finally, a closed polysurface is created by the geometric sewing of the different patches, resulting in a 3D surface-based solid model as shown in Fig. 3(c).

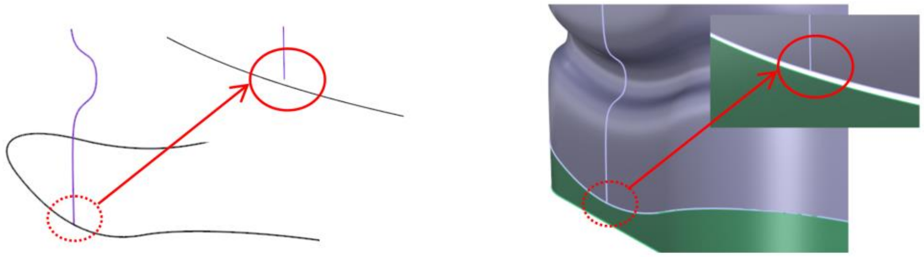


(a) (b) (c)

Figure 3: Silhouette curves, patch layouts, and surface formation from the reference modeling approach. From left to right: (a) silhouette curves and horizontal sections, (b) set of all patches defining the complete surface geometry, (c) entire bottle model consisting of all patches unified into a valid closed polysurface.

The reference CAD model consists of the actual CAD model created according to the modeling approach as outlined within the reference approach. At this point it needs to be made explicit that, once the curves layout and resulting surface formation have been created, the order as well as the type of commands being used later on to create the CAD model are neither fixed nor unique. This circumstantial condition is due to the fact that both curves and boundary edges of the previous modeling activity can be used as input for further patch modeling, thus resulting in different possible solutions. At this point during the actual exercise, students then have to define their own individual strategy to accomplish the goal of creating the properly modeled set of patches, which will eventually result in a closed polysurface if a union command on the patches as modeled can be successfully applied.

The reference deficiencies are structured as a form of information and knowledge repository on what can go wrong during a modeling exercise and what kind of errors are most likely to be committed by novices, and subsequently translate into known deficiencies being inflicted on a CAD model. Some examples to illustrate the nature and composition of this reference structure can be briefly outlined as follows. As the examples discussed in this paper are related to an exercise that is part of the first quarter of the course, one can expect students, being actual novices, to have some difficulty with managing and handling the correct input of coordinate values and the related interactive selection of computer-aided means such as system commands for keyword-based value input, snap to grid, and snap to key points, which are specifically designed for supporting those activities. Errors during the input of those coordinate values and geometric modeling data can result in severe deficiencies, which will have an adverse effect on the correct adjacency of curves. This represents a detrimental situation, which, in most cases, subsequently translates into adjacency deficiency among patches, thus negatively affecting the correct topology of patches.



(a) (b)

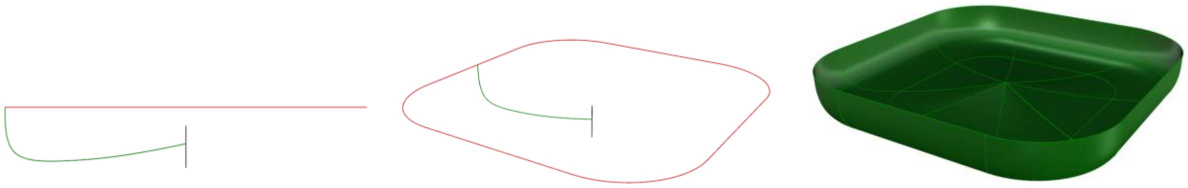
Figure 4: Examples of wrongly connected curves and resulting patch deficiency. From left to right: (a) curves with enlarged section presenting a gap, (b) associated patch surfaces with poor adjacency with enlarged section presenting a gap.

Fig. 4(a) and Fig. 4(b) show a selection of representative examples of wrongly connected curves used for patch surface formation. The gap between the vertical and horizontal curves as shown in Fig. 4(a) is 0.058mm, resulting in a poor adjacency condition between patches. This situation can be detected and filtered out within the CMGE inventory by means of a filtering query that will create the set of all CAD models which contain at least one closed polysurface (*periodic* = CLOSED_PS). Fig. 5(a) and Fig. 5(b) show the effect on results of a surface of revolution due to a wrong selection of the axis of rotation during modeling. Such deficient entities can be filtered by means of a query looking for a degenerated patch ('singular' field value equal to TRUE) or by checking non-singular patches (*singular* = FALSE) that are periodic or just closed in *U* or *V* direction (*periodic* ≠ OPEN_S) with a value of |*lU2* – *lU1*| or |*lV2* – *lV1*| in the [214, 215] reference range. Note that details on individual fields, their types, and their description as used within filter functions are given in the Appendix.

(a) (b)

Figure 5: Examples of deficiencies resulting from a wrongly selected vertical axis of rotation. From left to right: (a) bottom surface with a very small crack at its center, (b) bottom surface with enlarged section presenting a very small crack at its center.

Further deficiencies that can be expected in CAD models submitted by students for this exercise are related to their as yet not fully developed understanding of the concept of curvature continuity within a patch. This leads to an underestimation, or even total neglect, of the role and importance of this geometric modeling concept. A representative example, often encountered, is that the wrong modeling of the tangent condition of a section curve (see Fig. 6(a) and Fig. 6(b)) will result in a bottom surface with a spike, as shown in Fig. 6(c). Such deficiencies can be detected and filtered by means of a query looking for singular patches (*singular* = TRUE), with a small value that is not equal to zero (*Rcmin* ≠ 0 and *Rcmin* <= 1).



(a) (b) (c)

Figure 6: Faulty modeling of the tangent condition leading to a deficient bottom patch. From left to right: (a) elevation view of center axis and input curves, (b) perspective view of center axis and input curves, (c) bottom surface with a spike at its center.

Note that, in later CAD laboratory exercises, the occurrence of these deficiencies decreases significantly, as the concept of curvature continuity within a patch occurs repeatedly and in increasing depth throughout the lessons of the CAD course.

Errors most commonly committed by novices during the modeling of the top patch are related to using a wrong input curve and subsequently selecting a wrong system command to create the patch surface. Using a silhouette curve as the input curve and then employing the CAD system command that creates a surface from a boundary curve is a common example of this mistake. However, according to the CME reference, the correct way to create the top patch is to use as the input curve the circular curve defining the annular rim on top of the bottle and then employing the system command that creates a trimmed surface from a planar boundary. Mistakes that can be expected towards the end of the CAD model creation include frequent omission of actually modeling the top patch. Another error frequently observed during CAD modeling exercises is not joining the patches of the surface model to create a closed polysurface, which prevents the forming of a valid solid model.

4.4 Assessment and Framing of Feedback

4.4.1 Necessary and sufficient conditions

For modeling within surface-based CAD systems, featuring a closed polysurface generally constitutes a necessary and sufficient condition for a 3D CAD model to be converted into a solid model. However, in the context as outlined earlier, in order for solid models to also be considered fair (cf. continuous and monotonic curvature segments in [13]) and usable, additional conditions are required, because in those cases, the presence of a closed polysurface is only a necessary condition, but not a sufficient condition.

In a similar manner, in general, conditions of planar and trimmed surfaces alone are neither a necessary nor a sufficient condition for determining whether a 3D surface-based CAD model represents a solid model. However, within the exercise discussed in this paper, requirements for students to correctly create a CAD model that meets the exercise frame, demand, among other conditions compulsory, the presence of at least one planar and trimmed surface, namely the top patch surface (simplified bottle cap) that is required to close the CAD model. Therefore, within this context, for the analysis of a 3D surface-based CAD model, the condition of a planar and trimmed surface represents a necessary condition, which, however, is not sufficient. Note that the evaluation of assertions employing conditions or combinations of conditions deemed necessary, sufficient, or both, is used for the design and application of filter functions. However, as the final model assessment is based not merely on the formal truth-functional interpretation of logical implications derived from those assertions, but more on the actual validity of the assertions and the circumstances that lead to them, issues of what is referred to in the literature as *material implication* (cf. [4]) can be avoided.

4.4.2 Detection and analysis of CAD model deficiencies

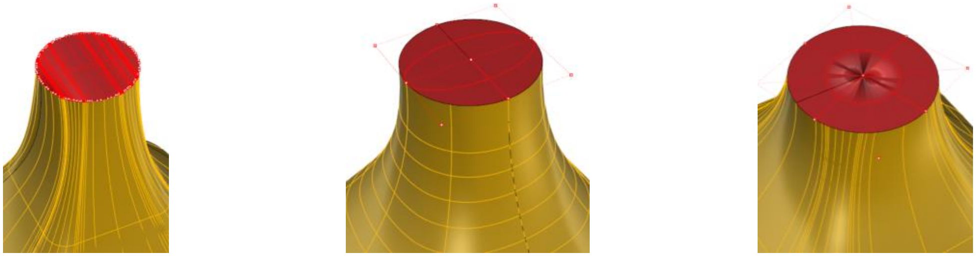
CAD model analysis: Part I

First, all CAD models in the exercise need to be checked to see whether they contain at least one closed polysurface, which represents a necessary condition for a 3D surface model to be regarded as a solid model. This condition can be detected and filtered by means of a query looking for records with a 'periodic' field value equal to CLOSED_PS. Note that, under the surface modeling system currently in use, this condition is also a sufficient condition. Application of the filter functions in the software tool analyzing the characteristics of geometric entities in the CMGE inventory in regard to this condition returned 72 CAD models. This means that 61% of CAD model submissions for this exercise produced a 3D solid model and 39% (46 CAD models) did not qualify as a solid model. As the goal of the CAD model analysis is to provide input for learning outcome assessment and formative feedback, additional CAD model analysis needs to be performed.

CAD model analysis: Part II

According to the CME reference, in addition to the creation of a valid solid model, two major components of the exercise relating to learning goals and outcomes were the correct modeling of both the top patch and the bottom patch. Searching for missing and wrongly modeled top patches also has been put into practices as filter functions. The functions implement the associated necessary conditions through analyzing the characteristics of geometric entities in the CMGE inventory in regard to featuring at least one planar and trimmed surface ('planar' field value equal to PLANAR and 'trim' field value equal to TRIMMED). Application of those filter functions returned 102 CAD models (86%) satisfying and 16 CAD models (14%) not satisfying these conditions. Applying filter functions that implement a conjunctive combination of the above outlined necessary conditions regarding a valid solid model and a properly modeled top patch based on the intersection of the previously computed four CAD model sets, returned results as follows.

Of the 72 solid models, one model was found with either a missing or a wrongly modeled top patch. As CAD models with missing or deficient top patches are usually not valid solid models within the context described, this result represents an unexpected situation which was further investigated as described later in this sub-section. Of the 46 non-solid models, 15 were found with either a missing or a wrongly modeled top patch. With the results obtained so far, which had considerably reduced the number of CAD models deemed to be deficient and thus requiring a detailed assessment, some manual visual analysis was still necessary, and this was conducted using the DGE backtrack functionality of the software tool. This human expert driven analysis yielded results as follows. 24 CAD models were found to contain severe deficiencies, which prevented the application of a correct join operation for the individual surface patches, to produce a valid closed polysurface. 12 CAD models were found to be still in a modeling state, where just the application of the union operation was required to produce a closed polysurface. Within the 15 CAD models out of the 46 non-solid models that were found to suffer from either a missing or a wrongly modeled top patch, 10 CAD models were found to be without a top patch, while 5 contained a top patch which had been wrongly modeled. In 2 cases, the dimension was wrong according to the exercise requirements, resulting in patch surfaces being too large. In the remaining 3 cases, a wrong modeling strategy had been used, resulting in a sequence of wrong modeling commands, leading to the complex patch geometry being faulty and very different in structure from the planar and trimmed patch required.



(a) (b) (c)

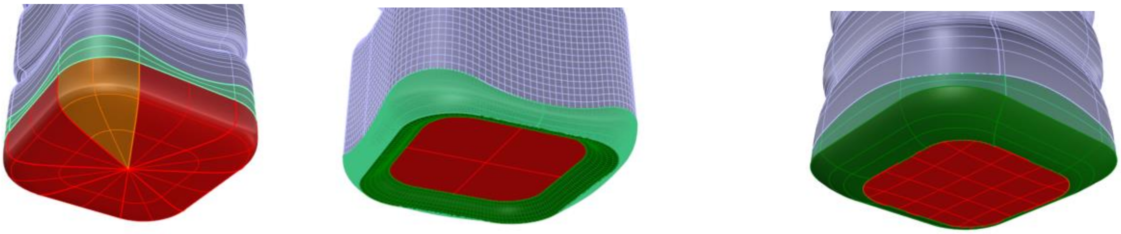
Figure 7: Examples of deficient top patch surfaces. From left to right: (a) non-planar boundary patch, (b) complex patch with 2 sets of overlapping control points, (c) complex patch with 8 overlapping control points at its center.

In particular, the top patch surface shown in Fig. 7(a) has been modeled as a non-planar boundary patch using the faulty open boundary of the bottle rim. The top patch surface shown in Fig. 7(b) has been modeled by applying a completely unexpected and obscure strategy, resulting in a patch with 1 joint edge and 2 boundary edges employing two sets of overlapping control points consisting of 2 and 4 control points, respectively. Finally, the top patch shown in Fig. 7(c) has been modeled by applying the revolution of a straight segment, resulting in a patch with 3 boundary edges, 1 joint edge and 8 overlapping control points at its center. Note that, in order to show clearly all the different types of patches which have been created by students during the modeling of the bottle body, along with all the deficient top patches, all the examples in Fig. 7 are shown with an enlarged section view which makes visible the patterns of the isoparametric curves.

In the case of the solid model that was found to have a missing or wrongly modeled top patch, visual analysis revealed that this model in fact had a top patch that was correctly connecting its patch surfaces to all the surfaces of the adjacent patches, eventually allowing for the creation of a closed polysurface. However, as the geometric structure of this patch was neither a planar nor a trimmed patch, it was considered deficient in regard to the CME reference for this exercise, although it does not represent in itself an error from the viewpoint of geometric modeling.

CAD model analysis: Part III

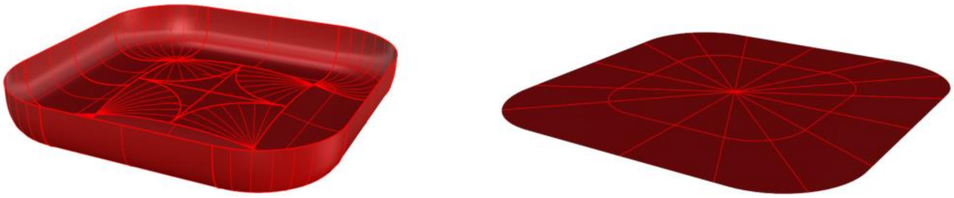
Within the third part of this CAD model analysis, model assessment is directed towards deficiencies in regard to the modeling of the bottom patch. According to the CME reference for this exercise, the bottom surface should be modeled as a revolution patch, with a section curve having one of its ends aligned in a coincident manner with the axis of rotation, resulting in a so-called *degenerated patch* (cf. [42]). The filter functions of the software tool, which inspect the CMGE inventory, revealed that 81 CAD models (69%) had one degenerated patch ('singular' field value equal to YES), 5 CAD models (4%) had more than one degenerated patch, and 32 CAD models (27%) were without any degenerated patch. A more detailed visual analysis of the 5 CAD models containing more than one degenerated patch showed, that only one actually had a deficient bottom surface, which consisted of 8 degenerated patches (Fig. 8(a)), while the others were typical cases of filter interference, as they were the same CAD models that were detected earlier as having deficient top patches, as they were modeled as revolving patches instead of planar and trimmed patches.



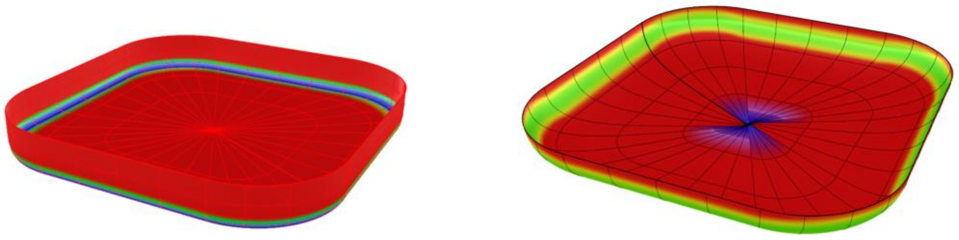
(a) (b) (c)

Figure 8: Examples of wrongly modeled bottom patches. From left to right: (a) polysurface made of 8 patches, (b) non-compliant planar trimmed bottom patch, (c) non-compliant NURBS trimmed bottom patch.

Now, the remaining 32 CAD models without any degenerated patch required further analysis. According to the CME reference for this exercise, there are basically two major reasons, from the perspective of a novice, which may cause the correct modeling of the bottom patch to go wrong. First, the bottom patch is sometimes modeled as a planar trimmed patch (see examples in Fig. 8(b), Fig. 8(c), Fig. 9(b), and Fig. 9(c)). Second, the bottom patch creation is correctly approached by modeling attempts aimed at creating a revolution patch, but they end up in a deficient surface that contains an open boundary due to tiny cracks, most likely to appear in the center of the patch (see Fig. 9(d)). Applying filter functions to search for CAD models with a planar bottom patch in the previously computed set of 32 deficient CAD models yielded 7 CAD models. A human expert driven visual analysis, using the DFG backtrack functionality of the software tool, confirmed the correct detection of this anticipated deficiency in all 7 CAD models. Now, the remaining set of 25 deficient CAD models can be analyzed using filter functions that scan for singular (*singular* = YES) NURBS patches (*SubType* = NURBS_S) with a particular surface size ('area' field value within a reference range) and a curvature that is smaller than 1.5 mm (*Rcmin* <= 1.5). Note that this time filter interference similar to that previously encountered and outlined can be avoided by searching for particular, more specific, patches, which have a surface size that is larger than 2000 mm^2. This additional size threshold value is derived from the CME reference and is directly related to the CAD modeling exercise. At this point, however, it needs to be mentioned that one drawback of this method of using additional size-related threshold parameters in the filter functions is that there is a chance of missing CAD models that have no explicit topological or geometrical deficiency *per se*, but contain an error in their dimensioning. Fortunately, this situation was contained by additionally applied filter functions, which scanned for dimension-related deficiencies. This precaution revealed 6 CAD models with bottom patches of an incorrect size.



(a) (b)



(c) (d)

Figure 9: Examples of bottom patches depicting shape together with related isoparametric curves and a curvature map. From left to right: (a) self-intersecting patch wrongly modeled using a sweep command, (b) planar patch wrongly modeled using a revolution command, (c) revolution patch modeled using the correct command but with a wrong geometrical design, resulting in a flat bottom, (d) revolution patch with spikes at its center due to wrong positioning of the revolving curve end in respect to the axis of rotation.

Analysis of the remaining deficient CAD models by employing filter functions similar to those described earlier produced results as follows. 6 CAD models contained a bottom patch with an open boundary, 8 CAD models contained self-intersecting patches, and 1 CAD model was incomplete, as it lacked a bottom patch, as well as lacking other geometric entities. Deficiencies such as incomplete CAD models and bottom patches with open boundaries are to be expected in this context.

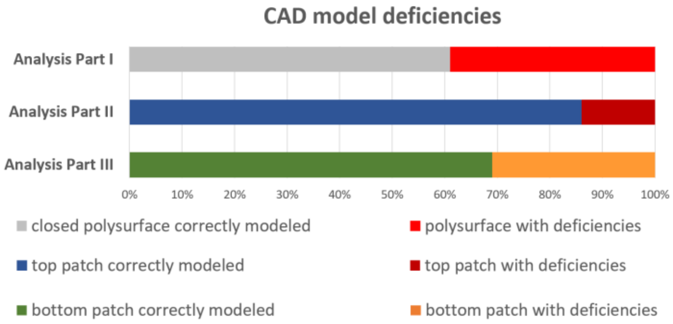


Figure 10: Graphical representation of proportions of correctly modeled closed polysurfaces and related main surface patches and their deficient counterparts corresponding to CAD model analysis part I, part II, and part III.

However, the 8 deficient CAD models, which contained self-intersecting bottom patches (see example in Fig. 9(a)) were totally unexpected and warrant further analysis. Using the DFG backtrack functionality, manual inspection of each of the 8 deficient CAD models revealed that the most common error producing such an unexpected result was the use of a wrong modeling command, in the form of a sweep command along a rail curve instead of a revolution command along a rail curve. A graphical summary of the proportions of CAD model deficiencies in each of the main categories, as discussed above in the three-part CAD model analysis, in the form of a stacked bar chart (cf. [20]) is given in Fig. 10.

4.4.3 Feedback scaffolding and input for course improvement

In general, feedback can be conceptualized as information provided by an agent, such as a teacher or system, regarding particular characteristics of one's performance. As such, feedback is a form of response to, and thus a consequence of, performance. Within the educational context, feedback is one of the most powerful influences on learning and competency development. As feedback is an integrated part of the teaching and learning process, it needs to provide information specifically relating to the processes and tasks of learning to reduce the gap between what is understood and what is aimed to be understood ([37]). Studies reported, for example, in [6,22] showed that feedback was most effective, while also leading to greater student engagement and increased achievement, if it was provided extensively and was specific to tasks, indicating how to perform better and more effectively. Therefore, information provided through feedback needs to be related to learning goals and outcomes and about their attainment in relation to specific tasks and performances. Within the formative feedback framework, as developed for the recently restructured CAD course, currently three dimensions are addressed that correspond to the focus at which feedback is directed (see also discussions in [6,19]). The nature and some basic characteristics of those three dimensions can be outlined as follows.

The first dimension of the feedback is related to information provision about the product or solution that was created during a task or performance. In the given context, this represents the CAD models which were created by students for this exercise. Here the focus is mostly determined by the main learning goal and the learning outcomes related to positive knowledge as described elsewhere in this paper. Results from the assessment presented earlier that are relevant for this dimension of the feedback can be compiled into information on what kinds of errors have been introduced into the CAD models, what are the most likely causes of them, and, related to this, where and how improvement can be approached from the viewpoint of a student's learning efforts and experience. For example, let us consider the deficiencies relating to the lack of a closed polysurface in cases where only the final application of the union command to convert the patches into a closed polysurface was missing, or the cases of a missing top patch where most probably its creation has just been forgotten or overlooked. These cases can be used to remind students about the incompleteness of the solution they have created in regard to the exercise requirements, which in turn have been designed according to the learning goals and outcomes. The quality of the solution produced, i.e. the CAD model created for the exercise, can be improved considerably simply by avoiding those errors based on omission, that can be achieved through double-checking what has been created and comparing it with what was supposed to be created.

The second dimension of the feedback is related to providing information about the process used to create the product or solution, which is here the design and modeling employed to produce a CAD model according to the exercise requirements. Here, the focus is determined by the main learning goal and in some part by elements of the learning outcomes related to both positive knowledge and negative knowledge. Here, for example, deficiencies have been detected in the creation of the bottom patch, relating to mistakes in the modeling approach, as well as the design and execution of individual modeling steps. These reflect on errors in the process of creating the solution, i.e., the CAD model, and shortcomings in both know-how on the proper use of commands in respect to the modeling task being considered, and strategic knowledge as employed. Concrete instances relating to the former are the wrong selection of the axis of rotation and the direction of the tangent of the revolving curve, and the wrong positioning of the revolving curve end in respect to the axis of rotation. Concrete instances relating to the latter are the inappropriate and faulty attempts to create a bottom patch containing a non-compliant planar surface by falsely rotating a planar or partially planar section curve. Also, the mistake of trying to create the bottom patch by modeling a sweep along a rail curve can be added as a further example. Those mistakes and errors, derived from the assessment of CAD model deficiencies detected during the creation process, can be translated into formative feedback information in the form of general recommendations and some more specific advice. In cases where the CAD model was incomplete or almost finished, some general pointers can be given on how to complete the modeling process, thus eventually creating a solution that meets the basic exercise requirements. In particular, where there are errors in the modeling approach, for example in the six cases where the top patch was modeled using a faulty strategy, information can be given on what is wrong and some advice offered on alternatives (if applicable) to remedy the situation. In those cases where mistakes were made in using individual modeling commands within a correct modeling strategy, resulting in a deficient model, such as in the case of using an incorrect axis of rotation for modeling the surface of revolution for the bottom patch, concrete advice can be given on both the nature of the mistake and corrective measures to be taken.

The third dimension of the feedback is related to information provision about self-evaluation, confidence, and the skill and competency development associated with them. Here the focus is mostly determined by the learning outcomes related to negative knowledge in combination with some elements of the main learning goal. In reference to the concrete examples discussed within the second dimension of feedback, to elaborate more on the feedback focus toward what not to do in certain situations and to provide know-how on recognizing critical situations and thus providing ways of avoiding repetition of those mistakes, information can be given as follows. For example, in regard to the correct selection of the rotational axis and the correct positioning of the rotational profile ends in respect to the selected axis of rotation, additional advice can be offered as part of a critical situation description when teaching about the creation of a surface of revolution, which is required for correctly modeling the bottom patch. This advice can then be combined with guidance on how not to approach the creation of the bottom patch when modeling the planar patch surface, because employing this modeling approach requires a strategy that is not only different from what is compulsory for this exercise, but also likely to result in a non-compliant patch surface geometry. To support students in developing negative knowledge and expertise is an important component within overall competency development. Advancing a novice's development of negative knowledge not only improves the ability to correctly self-evaluate, but also increases confidence, because, in addition to knowing what to do and how to do it, the student also gains knowledge in parallel about what not to do under certain conditions (see also discussions on negative knowledge and confidence development in [16,30,32]).

Besides providing input for feedback, the detection and assessment of deficiencies also offers valuable input for improvement of the software tool and some components of the CAD course itself. In the case of the software tool, unexpected deficiencies such as surface-based solid models that contain a valid closed polysurface but lack a properly modeled top patch, and bottom patches that were created by using a command for a sweep along rail curves, provide guidance for the re-design of current filter functions, as well as the design of future additional functions. These are then better capable of performing a more focused search of the CMGE repository for those newly identified deficiencies. Also, current efforts are aimed at designing filter functions that take into account the lengths of particular boundary elements and the patch surface sizes. The purpose is to filter very small surfaces while searching for NURBS-based untrimmed patch surfaces that are not singular. This will aid the explicit detection of patches with deficiencies related to small cracks in revolution-based surfaces of wrongly modeled bottom patches. These efforts too are benefiting from the results obtained through the application of the software tool.

In the case of the CAD course itself, assessment results for unexpected mistakes have provided valuable input for guiding efforts to enhance the material contents used for the course lectures. Examples of this are the wrong application of and faulty input parameter setting for the modeling command used to create a patch surface of revolution, and tactical mistakes within the related modeling approach, reflecting, among other things, on shortcomings in the strategic knowledge developed by students up to that stage within the course. Concrete cases of modifications made include, for example, improvements aimed at material related to the teaching of negative knowledge, in particular an extension of situation boxes (cf. [32,33]) used to teach know-how on recognizing critical modeling situations and what not to do in those situations, in order to avoid committing errors. Another example is improvements aimed at material related to the teaching of positive knowledge, where the material and the emphasis related to the geometric modeling domain concept of surface continuity have been re-arranged in contents and order of priority and relevance.

5 CONCLUSIONS AND FUTURE WORK

In this paper, the approach, structures, and technical architecture developed and used for the design and actual implementation of an innovative and untried software tool have been outlined and discussed. The tool is aimed at supporting a learning outcomes oriented assessment of three-dimensional surface models within the context of CAD education. This novel approach is based on the computer-aided detection of deficiencies in CAD models created by students as an outcome of modeling exercise assignments. The modular open system structure of the software tool as developed and implemented allows for an unobtrusive CAD model assessment that neither alters nor compromises in any way the original CAD models. Moreover, after compilation of the CMGE repository, computer-aided assessment can be performed as many times as deemed necessary without essentially requiring actual access to the original CAD models subject to assessment. A compiled selection of examples was given to illustrate the translation and application of central concepts of the framework and the technical architecture of the software tool and how these relate to and interact with exercise-specific learning goals and outcomes and the assessment of actual CAD models as created by students according to concrete exercise requirements.

Test and evaluation of the experimental prototype of the software tool produced promising theoretical and empirical results, which supported in several ways the shaping of insight and offered pointers for future work. Detection and assessment of several unexpected deficiencies in CAD models created by students stemmed from, among other things, shortcomings in their understanding of domain subject concepts and lack of know-how about correctly applying certain modeling commands. One example of this was an inability to properly provide correct input parameters to create a singular rotational patch surface. Assessment of the CAD model analysis results also revealed that students have difficulty in properly using certain modeling commands within their modeling strategies. In this regard, shortcomings also became evident in knowing what not to do and being able to recognize critical situations during the modeling process. These insights led to the consideration of modifications in the style of presentation and in the focus on fundamental domain concepts such as surface continuity. As a result, some improvements and re-arrangements of the lecture material taught in the CAD course regarding competency and skill development, related to both positive and negative knowledge, are under way. In particular, improvements are planned in the teaching of the relationships among modeling goals and system commands, and the actual outcome of the latter. In progress is also an extension of the teaching material related to know-how development on avoiding mistakes and grave errors by expanding situation boxes, while taking into account the newly gained insight on shortcomings related to negative knowledge as outlined above.

Experience and understanding acquired during the experimental use of the prototype software tool also provided some valuable pointers for future work aimed at the improvement of both the framework and the software tool implementation. The detection of several quite unexpected deficiencies in the student-created CAD models indicates a need to extend the current framework in regard to methods and definitions currently used to search for deficiencies. In regard to the software tool, this requires modification of some of the filter functions, to enable an explicit and more effective, as well as efficient, method of operation in detecting an expanded range of deficiencies. Those modification efforts should, in turn, provide some constructive input for work planned to extend the internal structure of the CMGE repository and inventory. Here, work is planned to provide extra data fields representing additional properties of geometric entities, which can support the detection of overlapping control points, self-intersecting surfaces, and very small cracks in polysurfaces. Based on the results of the evaluation of the experimental prototype system, preparations are under way to fully integrate and deploy the software tool in the coming academic year to support formative assessment and formative feedback within the recently reformed CAD course in mechanical engineering.

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

REFERENCES

[1] Adnan, M. F.; Daud, M. F.; Saud, M. S.: Contextual Knowledge in Three Dimensional Computer Aided Design (3D CAD) Modeling: A Literature Review and Conceptual Framework, in: Proceedings of the International Conference on Teaching and Learning in Computing and Engineering, April 11-13, Kuching, Sarawak, Malaysia, 2014, 176-181. https://doi.org/10.1109/LaTICE:2014.41

[2] Barequet, G.; Duncan, C. A.; Kumar, S.: RSVP: A Geometric Toolkit for Controlled Repair of Solid Models, IEEE Transactions on Visualization and Computer Graphics, 4(2), 2006, 162–177. http://dx.doi.org/10.1109/2945.694983

[3] Beier, K.-P.; Chen, Y.: Highlight-Line Algorithm for Realtime Surface-Quality Assessment, Computer-Aided Design, 26(4), 1994, 268–277. http://dx.doi.org/10.1016/0010-4485(94)90073-6

[4] Bennett, J.: A Philosophical Guide to Conditionals, Clarendon Press, Oxford, UK, 2003.

[5] Bischoff, S.; Kobbelt, L.; Kumar, S.: Structure Preserving CAD Model Repair, Computer Graphics Forum, 24(3), 2005, 527–536. http://dx.doi.org/10.1111/j.1467-8659-2005.00878x

[6] Black, P.; Wiliam, D.: Assessment and Classroom Learning, Assessment in Education, 5(1), 1998, 7-75. http://dx.doi.org/10.1080/09695980050102

[7] Black, P.; Wiliam, D.: In Praise of Educational Research: Formative Assessment, British Educational Research Journal, 29(5), 2003, 623–637. https://doi.org/10.1080/0141192032000133721

[8] Branoff, T. J.; Hartman, N. W.; Wiebe, E. N.: Constraint-Based Solid Modelling: What do Employers Want our Students to Know?, Engineering Design Graphics Journal, 67(1), 2003, 6 – 11.

[9] Carless, D.: Differing Perceptions in the Feedback Process, Studies in Higher Education, 31 (2), 2006, 219–233. https://doi.org/10.1080/03075070600572132

[10] Chester, I.: Teaching for CAD Expertise, International Journal of Technology and Design Education, 17(1), 2007, 23-35. https://doi.org/ 10.1007/s.10798-006-9015-z.

[11] Chester, I.: 3D-CAD: Modern Technology – Outdated Pedagogy?, Design and Technology Education: An International Journal, 12(1), 2007, 7-9.

[12] Elvira, Q.; Imants, J.; Dankbaar, B.; Segers, M.: Designing Education for Professional Expertise Development, Scandinavian Journal of Educational Research, 61(2), 2017, 187– 204. http://dx.doi.org/10.1080/00313831.2015.1119729

[13] Farin, G.: Curves and Surfaces for CAGD: A Practical Guide, Academic Press, San Diego, CA, USA, 2002.

[14] Goel, V.; Pirolli, P.: The Structure of Design Problem Spaces, Cognitive Science, 16(3), 1992, 395-429. https://doi.org/10.1207/s15516709cog1603_3

[15] Gonzáles-Lluch, C.; Company, P.; Contero, M.; Camba, J. D.: A Survey on 3D CAD Model Quality Assurance and Testing Tools, Computer-Aided Design, 83, 2017, 64-79. https://doi.org/ 10.1016/j.cad.2016.10.003

[16] Gartmeier, M.; Bauer, J.; Gruber, H.; Heid, H.: Negative Knowledge: Understanding Professional Learning, Vocations and Learning, 1(2), 2008, 87-103. http://dx.doi.org/10.1007/s12186-008-9006-1

[17] Gu, H.; Chase, T. R.; Cheney, D. C.; Bailey, T.; Johnson, D.: Identifying, Correcting, and Avoiding Errors in Computer-Aided Design Models which Affect Interoperability, Journal of Computing in Information Science in Engineering, 1(2), 2001, 156-166. https://doi.org/10.1115/ 1.1384887.

[18] Hartman, N.: Integrating Surface Modeling into the Engineering Design Graphics Curriculum, Engineering Design Graphics Journal, 70 (1), 2005, 16 – 22.

[19] Hattie, J.; Timperley, H.: The Power of Feedback, Review of Educational Research, 77(1), 2007, 81-112. https://doi.org/10.3102/003465430298487

[20] Heiberger, R. M.; Robbins, N. B.: Design of Diverging Stacked Bar Charts for Likert Scales and Other Applications, Journal of Statistical Software, 57(5), 2014, 1-32. https://doi.org/ 10.18637/jss.v057.i05

[21] Irons, A.: Enhancing Learning through Formative Assessment and Feedback, Routledge, Abingdon, UK, 2008.
[22] Kluger, A. N.; DeNisi, A.: The Effects of Feedback Interventions on Performance: A Historical Review, a Meta-Analysis, and a Preliminary Feedback Intervention Theory, Psychological Bulletin, 119(2), 1996, 254-284. https://doi.org/10.1037/0033-2909.119.2.254
[23] Liu, J.: Calculation of Profile Error for Complex Surface, Measurement, 48, 2014, 183–186.
[24] Liu, K.; Peng, X.; McGary, P.; Yalvac, B.; Ozturk, E.; Johnson, M. D., Valverde, L.: Integration of Contextual Exercises in Computer-Aided Design Education, Computer-Aided Design and Applications, 12(S1), 2015, 13-21. https://doi.org/10.1080/16864360.2015.1077071
[25] Lockett, H.; Guenov, M.: Similarity Measures for Mid-Surface Quality Evaluation, Computer-Aided Design, 40, 2008, 368–380. https://doi.org/10.1016/j.cad.2007.11.008
[26] Lynch, C.; Ashley, K. D.; Pinkwart, N.; Aleven, V.: Concepts, Structures, and Goals: Redefining Ill-Definedness, International Journal of Artificial Intelligence in Education, 19(3), 2009, 253–266.
[27] Mandorli, F.; Otto, H. E.: Negative Knowledge and a Novel Approach to support MCAD Education, Computer-Aided Design and Applications, 10(6), 2013, 1007–1020. http://dx.doi.org/ 10.3722/cadaps.2013.1007-1020
[28] Marsh, P.: What is Known about Student Learning Outcomes and How Does It Relate to the Scholarship of Teaching and Learning?, International Journal for the Scholarship of Teaching and Learning, 1(2), 2007, Article 22. https://doi.org/10.20429/ijsotl.2007.010222.
[29] Menary, G. H.; Robinson, T. T.: Novel Approaches for Teaching and Assessing CAD, in: Proceedings of the 17th International Conference on Engineering Education, August 21-26, Belfast, Northern Ireland, UK, 2011, Paper-No.: MO.P.SA 11.228.
[30] Minsky, M.: Negative Expertise, International Journal of Expert Systems, 7(1), 1994, 13-19.
[31] Oser, F. K.; Näpflin, C.; Hofer, C.; Aerni, P.: Towards a Theory of Negative Knowledge (NK): Almost-Mistakes as Drivers of Episodic Memory Applications, in: Bauer, J. and Harteis, C. (eds.), Human Fallibility: The Ambiguity of Errors for Work and Learning, Springer, Dordrecht, The Netherlands, 2012, 53-70. http://dx.doi.org/10.1007/978-90-481-3941-5_4
[32] Otto, H. E.; Mandorli, F.: Integration of Negative Knowledge into MCAD Education to Support Competency Development for Product Design, Computer-Aided Design and Applications, 14(3), 2017, 269-283. https://doi.org/10.1080/16864360.2016.1240448
[33] 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(1), 2018, 80–93. https://doi.org/10.1016/j.jcde.2017.11.006
[34] Parviainen, J.; Eriksson, M.: Negative Knowledge, Expertise and Organisations, International Journal of Management Concepts and Philosophy, 2(2), 2006, 140-153. http://dx.doi.org/10.1504/ IJMCP.2006.010265
[35] Roos, B.; Hamilton, D.: Formative Assessment: A Cybernetic Viewpoint, Assessment in Education, 12(1), 2010, 7 – 20. https://doi.org/10.1080/0969594042000333887
[36] Rynne, A.; Gaughran, W.: Cognitive Modeling Strategies for Optimum Design Intent in Parametric Modeling, Computers in Education Journal, 18(3), 2008, 55-68.
[37] Sadler, R.: Formative Assessment and the Design of Instructional Systems, Instructional Science, 18(2), 1989, 119-144. https://doi.org/10.1007/BF00117714
[38] Sadler, D. R.: Formative Assessment: Revisiting the Territory, Assessment in Education, 5(1), 1998, 77–84. https://doi.org/10.1080/0969595980050104
[39] Shute, V. J.: Focus on Formative Feedback, Review of Educational Research, 78(1), 2008, 153 –189.
[40] Weaver, G. C.; Burgess, W. D.; Childress, A. L.; Slakey, L. (eds.): Transforming Institutions: Undergraduate STEM Education for the 21st Century, Purdue University Press, West Lafayette, IN, USA, 2016.
[41] Wiebe, E. N.: Transfer of Learning between 3D Modeling Systems, Engineering Design Graphics Journal, 67(3), 2003, 15 – 28.

[42] Wolter, F.E.; Tuohy, S. T.: Curvature Computations for Degenerate Surface Patches, Computer-Aided Geometric Design, 9(4), 1992, 241-270. https://doi.org/10.1016/0167-8396(92)90033-L

[43] Xue, D.: Teaching CAD in Mechanical and Manufacturing Engineering Programs – An Experience at University of Calgary, in: Proceedings of the Canadian Design Engineering Network (CDEN) Conference, Kaninaskis, Alberta, Canada, 2005, Paper-No. 9.

[44] Yang, J.: A Procedural Approach of Inspecting CAD Model Errors, Engineering Computations, 21(7), 2004, 736–747. https://doi.org/10.1108/02644400410548378

[45] Yang, J.; Han, S.: Repairing CAD Model Errors Based on the Design History, Computer-Aided Design, 38, 2006, 627–640. https://doi.org/10.1016/j.cad.2006.02.007

[46] Ye, X.; Peng, W.; Chen, Z.; Cai, Y.-Y.: Today's Students, Tomorrow's Engineers: An Industrial Perspective on CAD Education, Computer-Aided Design, 36(14), 2004, 1451-1460. https://doi.org/10.1016/j.cad.2003.11.006

APPENDIX

A description of the CMGE repository associated text file format is given in this appendix. The basic structure consists of 27 fields of variable length for each data record entry. If has been designed to store information about geometric entities related to curves, polycurves, surfaces, and polysurfaces. Note that some fields can have a different interpretation, depending on the type of entity being stored. In cases where the field is not significant or not applicable for the type of entity stored, the field value is set to a default value.

Field name	Field type	Description
File	String	name of the Rhinoceros model file
ID	String	Rhinoceros unique ID of the entity
Type	Integer	Rhinoceros coding of the entity type (4=curve, 8=surface, 16=polysurface, etc.)
SubType	String	specific entity type (LINE, ARC, CIRCLE, ELLIPSE, NURBS_C, POLYCURVE, PLANE, CYLINDER, CONE, SPHERE, TORUS, NURBS_S
Span	String	YES if the curve or the surface is single span; NO if the curve or the surface is not single span; NON_UNIFORM_DG, UNIFORM_DG, NOT_SMOOTH for polycurve; number of patches for polysurface
gU	Integer	curve degree – surface degree in u direction
gV	Integer	surface degree in v direction; 0 in case of curve
nPtU	Integer	curve number of Ctrl points – surface number of Ctrl points in u direction
nPtV	Integer	surface number of Ctrl points in v direction; 0 in case of curve
nPtT	Integer	total number of Ctrl points
trim	String	TRIMMED/UNTRIMMED surface; blank for curve
planar	String	PLANAR if the curve or the surface lay on a plane; 3D otherwise
periodic	String	PERIODIC_C, CLOSED_C OPEN_C for curves; PERIODIC_U, PERIODIC_V, CLOSED_U, CLOSED_V, OPEN_S for surfaces; OPEN_PS, CLOSE_PS for polysurface
rational	String	RATIONAL_C, NON_RATIONAL_C for curves; RATIONAL_S, NON_RATIONAL_S for surfaces
singular	Boolean	TRUE if the surface has at least one degenerated boundary, FALSE otherwise; blank for curve
lU1	Float	length of first boundary edge of a surface (-1 if singular); length of a curve
lV1	Float	length of second boundary edge of a surface (-1 if singular); 0 for curve
lU2	Float	length of third boundary edge of a surface (-1 if singular); 0 for curve
lV2	Float	length of fourth boundary edge of a surface (-1 if singular); 0 for curve
area	Float	area of a surface; volume of a closed polysurface; 0 for curve
density	Float	ratio length/nPtT for curves, area/nPtT for surfaces
Rcmin	Float	approximated minimum curvature radius for curve (-1 straight line); approximated minimum curvature radius along U or V for surface (0 planar surface)

Rcmax	Float	approximated maximum curvature radius for curve (-1 straight line); 0 for surface
Lmin	Integer	Number of local minimum radius for curve (-1 straight line); 0 for surface
Lmax	Integer	Number of local maximum radius for curve (-1 straight line); 0 for surface
Inflect	Integer	Number of inflection points on a curve (-1 straight line); 0 for surface
Sinter	Boolean	TRUE if the curve is self-intersecting, FALSE otherwise; blank for surface

NON_UNIFORM_DG = polycurve made of curves with individual degrees
UNIFORM_DG = polycurve made of curves with identical degrees
NOT_SMOOTH = closed curve with G^0 continuity
PERIODIC_C, = periodic curve
NURBS_C / NURBS_S = NURBS curve / NURBS surface
CLOSED_C / OPEN_C = closed / open curve
PERIODIC_U / PERIODIC_V = periodic surface along U/V direction
CLOSED_U / CLOSED_V = closed surface along U/V direction
OPEN_S = open surface
OPEN_PS / CLOSED_PS = open / closed polysurface
RATIONAL_C / NON_RATIONAL_C = rational / not rational curve
RATIONAL_S / NON_RATIONAL_S = rational / not rational surface