Graphical Representation of Parametric Feature-Based MCAD Model Characteristics

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@staff.univpm.it

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

Abstract. Educational reforms and recent trends have pushed CAD education further into the mainstream of higher education. This has resulted in a rapidly increasing number of students enrolling in introductory level CAD courses, producing a flood of digital assignment submissions that need to be evaluated in a timely manner. This overwhelming situation has led to accelerated work on software tools for autonomous analysis and grading. However, the type and complexity of CAD model that can be analyzed, and the quality of the feedback that is generated, are still quite limited. In response to this predicament, and also efforts to reform an actual mechanical engineering CAD (MCAD) course, a new approach, framework, and software tool have been developed, based on visual representation and analysis of metrics. These novel instruments are aimed at supporting a more diverse and inclusive analysis and assessment of MCAD models, with a focus on the overall outcome a student produces, which includes model structure, history of model creation, and modeling strategy. The first part of this paper presents the novel approach and instruments for enhanced feature-based MCAD model analysis in the educational context. The second part of this paper reports on their development and implementation, and on the empirical results obtained during testing and evaluation.

Keywords: MCAD model analysis and assessment, formative feedback, competency development, visual analysis, radial visualization, knowledge discovery.
DOI: https://doi.org/10.14733/cadaps.2023.322-349

1 INTRODUCTION

Recent developments in global labor markets and efforts in the reform of higher education, along with work in educational research, are creating awareness of the most prominent shortcomings and failures of current CAD education. Such efforts have also provided new insights and recommendations, with some pointing toward the need for more educational exercises in the CAD laboratory, to transform CAD education so that it is more student centered and learning as well as practice oriented. This, in turn, requires frequent assessment and high quality and timely feedback. However, the overall checking, detailed analysis, and subsequent assessment of CAD models within an educational context are different from their counterparts in commercial and industrial settings in regard to both the goal and the assessment criteria. This is most evident within skill and competency development, which benefits considerably from reflecting on, and thus learning from, errors and mistakes. However, formative assessment and feedback are essential prerequisites to accomplish this. To promote and advance formative feedback in computer-aided design for mechanical engineering (MCAD) education, feature-based model and geometric 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 by applying criteria related to model deficiencies that are the result of wrong or inappropriately applied system commands and partial or entire modeling strategies. Deviations from computer-aided design and modeling guidelines and best practices as taught in the CAD course, and expected to be reflected in the structure and properties of student-created CAD models, also need to be included in the assessment and feedback process. This represents a task that is not only comprehensive and complex, but also considerably knowledge intensive. This is related to the fact that analysis and assessment require not only the detection and identification of deficiencies that in many cases do not violate general normative knowledge about feature-based modeling and geometric modeling, but also knowledge about the modeling goals and how they have been translated into strategies and actions.

Within the context of MCAD education, parts of the latter can usually be associated with learning goals and outcomes related to particular exercises and course assignments. In the context of parametric feature-based solid model assessment, analysis and evaluation need to be based on both feature-related properties and characteristics and the topology and geometry of the final modeling result. In particular, properties of individual features, links and relationships among features, and characteristics of feature sequences and modeling histories — that were created for producing the final model shape — can be used as a proxy for assessing models and particular modeling steps in a reflective and ex post facto manner. Currently, most commercially available CAD systems that support feature-based modeling provide interactive commands at the user interface to allow for some basic form of inquiry about model properties and the characteristics of both feature entities and topological and geometric model entities such as feature modeling tree, feature type, and related shape defining elements. However, performing a purely manual feature-based solid model assessment by using such kinds of generic system command is in many cases a sensitive task, and not only can this devolve into quite a convoluted and time-consuming process, but it is also likely to result in neglecting to capture many of the student failures within individual assignments.

2 BACKGROUND, SCOPE, AND OBJECTIVES

2.1 Background and Related Work

Influence from global labor markets and educational reforms outlined earlier, combined with recent trends in pushing CAD education further into the mainstream of higher education, have resulted in a rapidly increasing number of students with a larger than ever variety in educational backgrounds enrolling in introductory level CAD courses and producing a flood of digital assignment submissions that need to be assessed and graded in a timely manner. In response to this overwhelming situation, work on software tools for automatic grading, and also for quality and integrity checks, has been accelerated considerably in academia, and, to some extent, also in commercial sectors (cf. [6,18,30,32]).

Those efforts and recent approaches dedicated to the automation of CAD model analysis, assessment, and 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. This is apparent, for example, in approaches for technical drawings and 2D CAD files that can be found in [6,24,26,33,59]. Examples of recent approaches for 3D CAD models and related empirical studies are reported in [2,21,34]. An interesting approach to providing visual feedback for automated CAD model grading using heat maps is reported in [28]. 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 [18]. Another major limitation of those approaches to the automation of CAD model analysis and grading is the inherent functional structure of having scripts and other pre-defined computerized means that are capable only of comparing submitted student work with a reference solution and determining what is correct or not in an inventory-checking-like manner. Hereby, the data space of errors, mistakes, and deficiencies — that were introduced into the CAD models by students during model creation — remains largely ignored. However, that is presumably where potentially valuable knowledge can be extracted to gain insight into the shortcomings and difficulties with which students may have struggled. However, analyzing this side of the data space during CAD model analysis cannot be carried out purely with predefined queries and scripts, because a part of this intrinsic and non-trivial information and knowledge needs first to be discovered, before means to look for it can be codified. Here, the authors believe, visual analysis has great potential, due to its ability to combine the creativity, vast knowledge, and visual perception system capabilities of the human user with the enormous computational power of computing machinery, to further insight into data and discover valuable, previously unknown, knowledge.

Since feedback is not only a major factor in improving student attainment, but also a central function of formative assessment, its provision is essential to maintain the quality of the learning experience in higher education. However, as aptly pointed out in [54,58], feedback representing information that is provided to the student is effective only if that student is actually taking it up and acting upon it. A wider recognition and acknowledgement of the importance of this acting on feedback has led to new approaches and concepts, and eventually to a gradual shift in research focus toward actionable educational feedback (cf. [4]), taking into account both students and teachers. Here traditional work on feedback that was centered around the provision of relevant information in regard to various feedback levels linked to tasks, processes, and self-regulated learning (cf. [7,14]) is extended and considered more as a process to facilitate change in student behavior [9]. Recent work on *feedback recipience,* that is concerns about how students understand and act upon feedback, is reported in [10,19,31,37,58]. Work addressing the discrepancy between the potential and actual use of feedback, known as the *feedback gap*, can be found in [14,16,52]. Recent views on the skills that are essential in enabling students to make effective use of educational feedback, referred to as *feedback literacy*, are provided, for example, in [9]. However, with the advent of current and newly introduced technology-supported feedback approaches and systems (cf. [39]) aimed at providing autonomous processes for feedback provision at scale [50], a certain neglect of the importance of feedback recipience and actionability, as discussed, for example, in [13,20], still seems to widely prevail.

2.2 Scope and Objectives

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, those metrics are, in most cases, directly linked to the assessment criteria for CAD models created by students and submitted to teachers for assessment and 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 insufficient to support learning from errors and developing metacognitive skills related to self-assessment, and subsequent self-improvement. Recent efforts to reform an actual CAD course for mechanical engineering at the institution represented by the authors, addressed, among other matters, the development of modeling competencies with particular reference to the strategic knowledge required to create usable MCAD models (for more details see [41,42,47,48]). In particular, this major course-specific learning goal, i.e. development of the strategic knowledge and modeling skills indispensable for producing usable MCAD 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. Moreover, it especially 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 providing high quality and timely feedback for both teacher and students. Here formative assessment and formative feedback appear to offer a viable solution (cf. [11,23,27]). They are also increasingly regarded as promising and effective components within currently proposed elements for reforming higher education in science and engineering, although several barriers need to be overcome to actually make this type of feedback really valuable for students and their learning experience (see also discussions in [8,54]). Within this setting, and in the context of higher education, as outlined earlier, the assessment of student performance and results produced in CAD laboratory exercises and course assignments needs to be conducted in a computer-aided manner. This will support actual implementation, while also improving the scope and overall quality of formative assessment and feedback, but it requires new approaches and tools for feature-based MCAD model assessment.

In response to this, combined with efforts to reform an actual CAD course for mechanical engineering at the institution represented by the authors [42,43], a new approach, framework, and software tool have been envisioned and developed. This new approach is based on the visual analysis of metrics — that is combining the creativity, knowledge, and visual perception system capabilities of the human user with the power of computers to look at and make sense of the feature-based characteristics (FBC) of MCAD models. The objective is to support a more diverse and inclusive analysis and assessment of CAD models. This involves a focus on the overall outcome a student produces, which includes model structure, history of model creation, and modeling strategy. Therefore, the application of these newly developed instruments is not limited to just one final modeling solution, but also allows for the analysis of partial modeling solutions, analysis across several versions of a final modeling solution, and a combination of both. The aim of the current paper is two-fold. Firstly, it presents a novel approach for enhanced feature-based MCAD model analysis in the educational context, which is based on an analysis using interactive visualizations of graphically represented feature-based model characteristics. Secondly, it reports on the structure and concrete implementation of the advanced graphical representation of feature-based model characteristics that are integrated into a software tool — that is currently being made available to teachers on a trial basis — to enable and put into practice this novel approach supporting and elevating computer-aided feature-based MCAD model analysis and assessment, and the subsequent provision of more tailored, timely, and high-quality formative feedback.

3 APPROACH, DESIGN, AND DEVELOPMENT

3.1 Approach and Implementation

Visual analysis represents a powerful means of understanding complex data, as visual displays allow humans to make use of their cognitive capacity to perceive and study various aspects of complex information and issues simultaneously. Hence, it supports the discovery of intrinsic, non-trivial and potentially valuable knowledge hidden in the data sets being visualized. To facilitate this, an advanced graphical representation of parametric feature-based MCAD model characteristics has been developed and deployed within a modularized interactive visualization environment, which has been developed and integrated into the architecture of the feature-based CAD model analysis module. This, in turn, is an integral part of the CAD model analysis and assessment software tool system developed by the authors. Although graphical representations are a comprehensive aid, their efficiency and effectiveness strongly depend on both the data to be visualized and the information and knowledge to be communicated. Within the context outlined in this paper, a particular form of radial visualization is used, namely *Kiviat diagrams*. In MCAD model deficiency analysis, the information subject to visualization can be structured into data that are associated with objects and data that are associated with processes. The former represent the relationships between, and the basic data for, the feature types, the feature scope, and the number of entities that were used to create the MCAD models. The latter relate to the basic tenets of best practice and various aspects of feature entity creation and CAD modeling sequence planning and execution. Based on these data, certain characteristics of trends, patterns, critical situations, and deficiencies can be visually represented and analyzed.

Current design of the Kiviat diagrams is based on the visual encoding of feature-based characteristics that results in certain aspects of the MCAD models being visually represented in the form of closed polygonal profiles defined by polylines of definite size, position, shape, and color. Those colored polylines are then the representative feature-based characteristics profile, which translates into a graphical representation within the visual space through encoding and mapping of spatial properties, color, and layout design of the Kiviat diagrams. The unique polygonal structure of those feature-based characteristics, and their visual encoding, allow for an effective and efficient approach to visually representing and analyzing the relative multidimensional data space of feature-based MCAD model characteristics. To improve the efficiency of the graphical representation, the layout design of the Kiviat diagrams has been optimized through superimposed property profiles, structured diagram sector sub-division, and diagram axis rearrangement. This rearrangement is based on finding an axis permutation that results in a combination of polylines with a shape easily and quickly recognized and understood by the human visual perception system. To determine such a shape, Prägnanz and related grouping principles [35,57] have been used. In particular — during layout optimization of the Kiviat diagrams — those grouping principles were employed as heuristics to obtain the best property profile shape in regard to Gestalt theory (cf. [1]).

As the prototype implementation of the visualization environment module needs to be integrated with previous work of the authors on MCAD model software assessment tool development (cf. [43,47,48]), data management and transformation within the visualization pipeline have to operate through the CAD model and feature entity (CMFE) repository that in turn facilitates the import from and export to different parametric feature-based solid modeling environments. Within the CMFE inventory the results are used to compile model entity analysis reports. The newly developed prototype implementation features a technical architecture that leverages API-based functionality provided by commercially available CAD systems to support a modular and highly cohesive system architecture. In the current implementation, the CAD modeling environment deploys a commercially available parametric feature-based solid modeling system, namely *SolidEdge* from Siemens AG. At present, the CMFE repository is compiled by extracting CAD data from the *SolidEdge* part models using Visual Basic for Applications (VBA) functions. This extracted data is then further processed and stored in structured *Excel* files. Next, those structured *Excel* files are imported into the Microsoft *Access* RDBMS (relational database management system) by means of macros, to facilitate the creation and build-up of the CMFE inventory. Currently, the modularized visualization environment is implemented using *Excel,* the VBA environment, and a data pipeline to the CMFE inventory that is channeled through compiled sub-sets of query reports. To process and visualize data correctly, compilation, export, import, and filtering within the information visualization pipeline (see Figure 1) in regard to the CMFE repository and inventory are organized as follows.

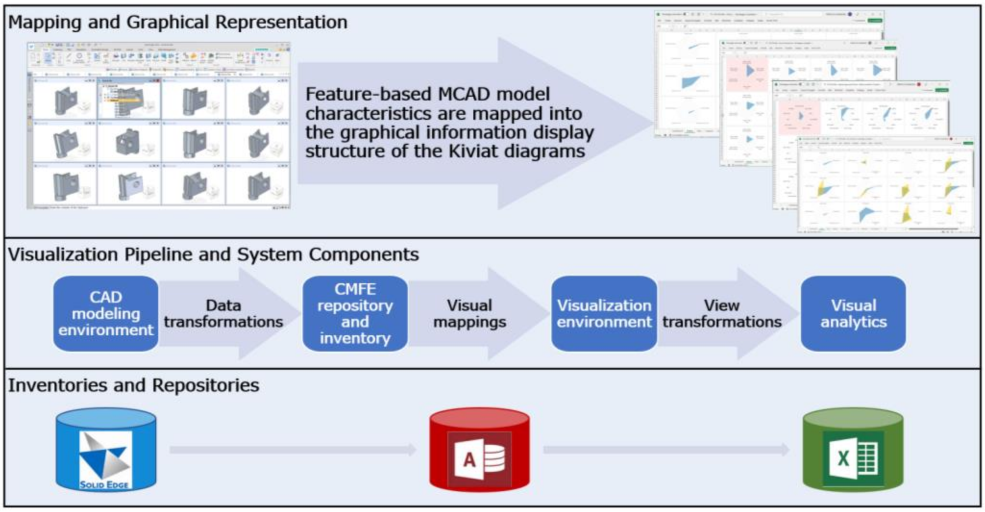
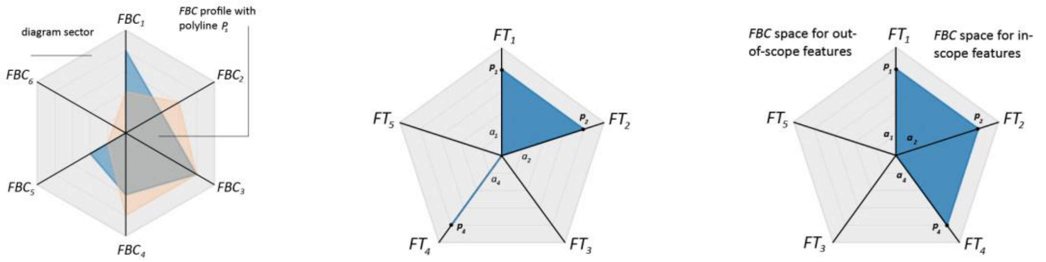


Figure 1: Overview of the information visualization pipeline and central system components.

All feature-based solid models that have been created by students are compiled and stored in the CMFE repository. This repository is structurally sub-divided into various sets of folders, with one set of folders for each exercise or course assignment. During the compilation process, information on feature entities and their related properties and meaningful characteristics, such as feature type, shape-defining topology and geometry, is extracted from the parametric feature-based solid models, codified, and stored in the form of structured files, with one file for each model. Data on parametric feature-based model entities and their properties and characteristics stored in the model repository are processed and imported into the CMFE inventory. This inventory provides 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. It should be noted that table entries for each feature entity in the model repository contain also an identifier-based link, which connects them to the geometric modeling system. Note that this link mechanism is essential to enable the implementation of a cross-linked view supporting linking and brushing (cf. [38,46,56]).

3.2 Mapping of Information and Graphical Representation

The basic graphical information display structure of the Kiviat diagrams [36] — in the literature also referred to as *star diagrams, polar charts,* and *radar charts* (see [15,51,60]) — consists of a set of equiangular axes denoted by a_k. Those axes are sometimes called *spokes*, due to their frequently being used in the literature within references to star plots, where they radiate from a common center point, with each representing one data dimension. Within the context of this study, each individual axis a_k encodes one feature-based characteristic denoted by FBC_k, together with a data point denoted by $\boldsymbol{p_k}$ located on it, at a proportional distance from the diagram center, that encodes data quantity (magnitude), as shown in Figure 2(a) and Figure 2(b). This data quantity, which graphically represents an entity count with the characteristics denoted by FBC_k, is related to the number or proportion of entities of the feature-based characteristic encoded by the axis which are verified to be present in a particular MCAD model.



(a) (b) (c)

Figure 2: Information mapping and graphical representation through a basic Kiviat diagram. From left to right: (a) Kiviat diagram with FBC space and superimposed FBC profiles, (b) Kiviat diagram with FBC space and FBC profile for generic feature types, (c) Kiviat diagram with optimized FBC space and FBC profile for generic feature types.

Where FBC_k is related to a type of feature denoted by *FT* (see Table 1 for a list of all feature types), the feature scope denoted by *FC* determines whether the feature type belongs to the set of in-scope features or the set of out-of-scope features. The feature scope is used to indicate whether a feature belongs to a feature type that is expected to be present in the MCAD model, according to the exercise requirements, and thus represents an in-scope feature type, or is not expected to be used, and thus represents an out-of-scope feature type (see also [48]). The presence of out-of-scope features is also indicated by a colored diagram background. Note that the occurrence of out-of-scope features can also indicate that particular features not required for the actual creation of the MCAD model have been used in a different context – for example as UNDO features or to recover from errors (cf. [40]). Now, feature-based characteristics, denoted as FT_k, of the MCAD model can be graphically represented by connecting the data points $\boldsymbol{p_k}$ of all axes with straight line segments across all diagram sectors to form a closed polygon chain, that is, the polyline denoted by P_s as shown in Figure 2(b).

This polyline then defines the geometries of property profiles regarding size, position, and shape within the diagram, which in turn forms a graphical representation of feature-based characteristics of the MCAD model according to requirements as outlined earlier. An illustrative example employing a Kiviat diagram based on such information mapping for feature-based characteristics consisting of just 6 generic characteristics, denoted by FBC_1, FBC_2, FBC_3, FBC_4, FBC_5, and FBC_6 is shown in Figure 2(a). Notice that concentric geometric structures such as circles or *m*-sided polygons, like the concentric regular hexagons shown in Figure 2(a), can be added to the diagram as grid lines, to improve and make more efficient the visual judging and comparison of radial distances. This results in a layout that probably led to the naming of those diagrams as radar or spider charts. Within the framework developed, the design of each diagram layout is denoted by $DL^{m,n}$ where the index *m* indicates the dimension of the diagram, that is the number of characteristics FBC_1, FBC_2, FBC_3, … FBC_m, which are encoded and graphically represented. The index *n* represents the identifier of each individual diagram layout within each set of *m*-dimensional diagram layouts. Note that due to the structure and encoding used in the diagram layout designs, as described in sub-section 3.3, the value of the index *m* of each $DL^{m,n}$ is equal to its corresponding counterpart for regular *p*-sided polygons in most of the prominent systems used to describe regular polytopes and tessellations, and to classify symmetry groups (see also discussions on the Schläfli symbol and the Coxeter notation in [12]).

To optimize the explicit encoding of the feature scope *FC* in regard to individual exercise requirements, the order of the equiangular diagram axes usually needs to be rearranged and clustered in view of features considered to be out-of-scope and in-scope. In parallel with this task, the FBC profile needs to be optimized in regard to its polyline P_s, which should represent a shape that can be recognized easily and quickly by the human visual perception system. This can be achieved by generating all possible circular permutations of the initial sequence $\{a_k\} = \{a_1,a_2,a_3,a_4,a_5\}$ of equiangular diagram axes (see again Figure 2(b)) and evaluating the resulting polylines taking into account some principles of Gestalt psychology (details are provided in the next sub-section). For the current simplified example, this process results in the permutated Kiviat diagram shown in Figure 2(c), which is based on the sequence $\{a_k\} = \{a_1,a_2,a_4,a_3,a_5\}$ of equiangular diagram axes. The graphical layout of this permutated Kiviat diagram is now capable of efficiently and effectively representing a clustered FBC property space for both the in-scope features (FT_1,FT_2,FT_4) within the first two diagram segments and the out-of-scope features (FT_5,FT_3) within the other diagram segments. The polyline of the profile in Figure 2(c) also represents a more regular shape, in form similar to a tilted rhomboid, which is easy for the human visual perception system to recognize and identify. Structural differences and outliers, indicating deficiencies and errors in the MCAD model, can now be graphically presented in an efficient and effective manner. This would be rather difficult to achieve with the polyline of the profile in Figure 2(b), which has a shape consisting of a triangle combined with a straight line.

In general, as indicated in the example previously outlined, the mapping of information to graphical elements as outlined above can be further used to create multiple closed polygonal profiles that can be superimposed within one diagram to allow comparisons among several distinct data sets with common data dimensions and respective mappings. This allows, for example, a direct comparison of feature-based characteristics across various versions of an MCAD model created to gain an insight into the progress of skill development in students. This approach is most effective when the area enclosed by one profile is entirely contained within another profile, or when transparency is used, allowing comparative analysis of relative areas without occlusion, as shown in Figure 2(a). In what follows, further details are provided and various references given to basic and advanced literature regarding the actual visual encoding and layout design, and related strategies used to optimize the discriminability and perceptual visibility of the visualizations developed. Firstly, this includes FBC property space optimization, and, in particular, FBC profile optimization based on selected principles of Gestalt psychology. Secondly, it includes the development of an optimized color scheme designed as a system default, to ensure that the color combinations used in the visualization are universally legible with a design that is clear and accessible to both viewers with color impaired vision and viewers with full color vision. Thirdly, it includes the development of an improved layout design, which was further optimized based on the principles of the rule of thirds (see also discussions in [53]).

3.3 FBC Space Organization and Visual Attribute Encoding

3.3.1 Encoding of spatial attributes

Data instances of feature-based characteristics of CAD models, for example, in the case of three components, feature scope *FC*, feature type *FT*, and feature count *FN*, are mapped to a visual space using Kiviat diagrams as follows. Firstly, the entire circular graphical space is split into n equiangular axes a_k, which are drawn radially from the diagram center to its perimeter, where n is the data dimension determined by the n different feature types *FT*, with each diagram axis a_k, for $k = 1,2,3,\ldots,n$, representing one feature type. Secondly, the feature count of FBC_k data instances for each *FT* is mapped to an entity position represented by a radial point $\boldsymbol{p_k}$ within the visual space, which is encoded as a distance r_k measured from the radial diagram center along the respective diagram axis a_k of the k-th feature type for which the feature count *FN* is encoded. Hence, the entity position $\boldsymbol{p_k}$ is defined by the pair (r, φ) of its coordinate values with the radial coordinate denoted by r and the angular coordinate denoted by φ, and computed as $r = \lambda r_k$ and $\varphi = (2\pi/n)(k-1)$, where λ is a linear coefficient used to normalize or adjust the graphical space (size) of the diagram in regard to the actual display medium. This adjustment also needs to be taken into account when FBC_k-related proportions instead of entity counts are used during encoding. Thirdly, the layout of the radial diagram is sub-divided into two diagram sector clusters, with one sector cluster containing only encoded data entities of FBC_k data instances with *FC* indicating out-of-scope features, and the other sector cluster containing only encoded data entities of FBC_k data instances with an *FC* value indicative of in-scope features. Fourthly, all points $\boldsymbol{p_k}$ can be connected with straight line segments l_k to form a polyline denoted by P_s for s = 1,2,3,…,m, with m indicating the number of MCAD models. Here, the joint polyline P_s can be formulated as an alternating sequence of joint vertices, that is the $\boldsymbol{p_k}$ and links, that is the line segments l_k, expressed as $P_s = (\boldsymbol{p_1}, l_1, \boldsymbol{p_2}, l_2, \ldots, l_n, \boldsymbol{p_{n+1}})$ with $\boldsymbol{p_1}$ and $\boldsymbol{p_{n+1}}$ sharing the same pair (r, φ) of coordinate values. This polyline P_s is the representative FBC profile, and thus the shape within the visual space that is graphically representing through the Kiviat diagram the feature-based characteristics of the s-th MCAD model. Notice that FBC space encoding is not limited to those characteristics discussed above. Further examples of concrete FBC space encoding and diagram layout designs are presented and discussed later, in the test and evaluation section.

3.3.2 Encoding of color

In what follows a brief overview is given on the design of color sets used for the graphical representation of FBC profiles. More details on the design of and user interactions with these color sets are given in [49]. The design and creation of color sets, also in the literature referred to as *color schemes* and *color palettes*, which are used for the mapping and encoding of color during visualization, can be approached using different strategies depending on the application field (cf. [25,44]). Within the visualization environment described in this paper, the various settings for the mode of color scheme relate to particular strategies that were used for the color scheme design. The principal strategy was to optimize design flexibility and customizability allowing for a personal overall color preference, and in particular for color pairing preferences, while supporting individual choices of color harmony, so that particular color schemes could be created by the user. This can be achieved by directly defining the color scheme either through the input of the parameter values of a color specification system, for example in the form of the additive primaries red-green-blue (RGB) used to produce emitted color, or by selecting particular colors using a color picker tool.

However, to optimize strategies based on discriminability and perceptual visibility as used in various visualization applications, while also attempting to encapsulate as much knowledge and insight as possible regarding the human visual perception system and the use of digital color in visualization (see also [46,55,56]), an optimized color scheme has been designed as a system default. Taking into account that color vision impairment is probably the most widespread physiological impairment (cf. [3,45]), this color scheme has been designed to ensure that the color combinations used in the visualization are universally legible. With such a barrier-free design, the visualization is clear and accessible to both the color impaired and the viewer with full color vision. The current design is based on a customized color scheme for qualitative data, which employs differences in hue to represent differences in data type. This color scheme was created with the online tool *ColorBrewer2* [22]. The legibility of the current customized color scheme design has been verified with a complementary software tool, namely *ColorOracle* [29], an open-source simulator of color-impaired vision. The test results confirmed the legibility of the color scheme design for the most frequent color vision impairment related to forms of what is commonly referred to as red-green confusion (see also deuteranopia / deuteranomaly and protanopia / protanomaly in [3,45]).

3.3.3 Optimization of diagram structure and layout design

In what follows a brief outline is given on the optimization of the diagram layout used for various visualization scenarios during the analysis examples provided in this paper. Depending on the application context, there are various methods of optimizing the layout and property profile of a radial visualization. These methods include shape moments and less complex shape descriptors (cf. [38,61]) used as comparative metrics. In some cases, the diagram axes are rearranged, resulting in what is called in the literature a *permutated chart / diagram*. However, within the application context and data mapping / encoding as described earlier, the geometry of the FBC polyline needs to be optimized less in view of the technical issues which are important for automated processes, but more in regard to what is known about human visual perception and cognitive aspects related to it.

This approach has been pursued employing the *Prägnanz* principle of Gestalt psychology. In Gestalt psychology it is assumed that, when a group of objects is observed, perception of their entirety takes precedence over the perception of individual parts (see also [1,17]). Moreover, when individual parts are perceived, they are grouped according to certain rules of perceptual grouping. In other words, perceptual grouping attempts to describe the way the human visual system determines which parts and objects of external visual stimuli belong together to form a meaningful perceptual unit. In this regard, perceptual grouping can also be considered as one process by which diverse parts of a visual scene can be aggregated into higher-order structures. In Gestalt psychology and the study of perceptual grouping, the fundamental principle, referred to as *Prägnanz* (cf. [17,57]), implying conciseness and orderliness, but also known as the principle of *good Gestalt*, is based on all the concepts described in [57], and known as the original *factors* or *principles of perceptual grouping* such as similarity, proximity, and good continuation (cf. [5]).

These principles not only allow us to predict the interpretation of how external visual stimuli are perceived, but also determine the best Gestalt possible based on what is visually given. This was explained by one of the leading Gestalt psychologists in the words, "Of several geometrically possible organizations that one will actually occur which possesses the best, simplest and most stable shape." ([35], p.138). However, Prägnanz, like some other original basic grouping principles, is still without a clear and formal definition, which is partly because Gestalt psychology and its methods are largely based on demonstrations (see again discussions in [5]). Therefore, within the work presented in this paper, Prägnanz and related grouping principles are used as heuristics to approximate a good Gestalt for the shape representing feature-based characteristics based on the FBC polyline of the reference CAD model. This, in turn, results in a permutated Kiviat diagram that is improved in view of human visual perception through the rearrangement of the diagram axes (for more details see [49]).

4 TEST AND EVALUATION

4.1 Outline

For testing and evaluation of the approach and the prototype implementation, functionality, performance, and reliability have been assessed by using parametric feature-based MCAD models that had been created by students for CAD laboratory and course exercise assignments in the previous academic year.

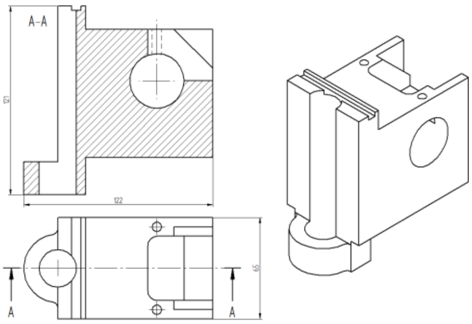


Figure 3: Outline and overall dimensions of the CAD model example from the exercise.

To keep the presentation and discussions concise and transparent, the number of MCAD models used in the examples in this paper is limited to a selection from a student cohort of 30 for one exercise. This exercise is usually administered during the first quarter of the course, requiring the design of an MCAD model with a relatively simple shape (see Figure 3) that can be created by novices using a basic set of about a dozen feature commands. The goal of this early exercise is to train students toward some of the expected learning outcomes of the course. This training includes understanding the importance of a well-designed modeling strategy with a focus more on CAD model quality rather than model shape, and an awareness of as well as adherence to various basic elements of CAD modeling guidelines and best practice. For this early exercise those basic elements include, for example, renaming all features, using non-complex profiles whenever possible, defining only fully constrained profiles, and applying extrusions first, cutouts next, and details such as rounds and chamfers at the end.

4.2 Analysis of MCAD Model Structure and Modeling Strategy

4.2.1 Layout design and FBC profile optimization

For the graphical representation of feature-based characteristics employing a Kiviat diagram based on FBC information as used in the analysis examples presented in this subsection two diagram layout designs, namely $DL^{11,1}$ and $DL^{4,2}$ were used. The latter is a basic regular tetragon-based layout design that consists of 4 diagram axes encoding a set of particular feature types, namely hole, user pattern, mirror copy, and extruded cutout. This is used for the analysis of how the modeling of the holes and circular passages / cutouts of the part model was approached by students.

Feature Type	In-Scope Feature	Out-of-Scope Feature
Chamfer		◇
Extruded Cutout	◇	
Extruded Protrusion	◇	
Hole	◇	
Mirror Copy	◇	
Mirror Part		◇
Revolved Cutout		◇
Revolved Protrusion		◇
Round	◇	
Slot		◇
User Pattern	◇	

Table 1: Overview of feature-based characteristics in regard to the feature type and feature scope.

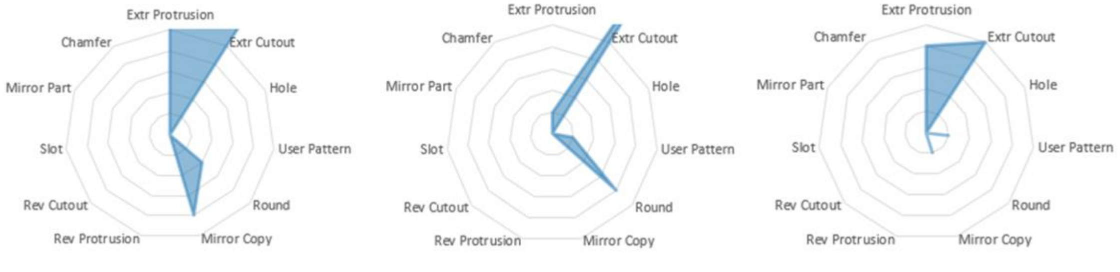
The more complex layout design $DL^{11,1}$ consists of 11 diagram axes that form a regular hendecagon FBC space for encoding the feature types listed in Table 1. Note that the feature scope, which is a more specific feature-based model characteristic based on the characteristics of the feature type, is encoded through diagram sector dependency of polygon segments. In this layout design encoding is realized through polyline segments of the polygon chain, with those plotted within sectors on the left diagram side indicating out-of-scope features and those plotted within sectors on the right diagram side indicating in-scope features.

Figure 4: Examples of permutated Kiviat diagrams with FBC space and FBC profile for 11 feature types with 6 feature types considered in-scope features and 5 feature types considered out-of-scope features.

To achieve a graphical representation with a shape that can be recognized easily and quickly by the human visual perception system, the FBC profile needs to be optimized in regard to its polyline. Following the optimization approach outlined within sub-section 3.3, a set of permutated Kiviat diagrams, with a layout employing the above-described feature-based characteristics and related information mapping and attribute encoding, has been created using a custom software tool routine developed by the authors. A selection of the permutated Kiviat diagrams created is shown in Figure 4, which also includes the diagram (encircled, upper left) that was selected as an error-free reference for the analysis described in the next sub-section.

4.2.2 Application and empirical results

Visual analysis of the FBC visualizations reveals immediately all the MCAD models, which contain an unusual number of features, indicating shortcomings in the modeling strategy used to create those models. For example, in one model an exceptionally high number of mirror copy features was found, as shown in Figure 5(a).



(a) (b) (c)

Figure 5: Exceptionally high or low numbers of certain in-scope feature types. From left to right: (a) exceptionally high number of mirror copy features, above average number of extruded protrusion and cutout features, (b) exceptionally high number of extruded cutout features and round features, low number of extruded protrusion features, (c) round features are missing.

A closer analysis helped to reveal that mirror copy features were properly used to create the pair of small holes and the pair of triangular cutouts. However, they were also used for rounding with a cutout feature and employed together with an extruded protrusion feature as a means of remedy (see also discussions on undo of modeling operations in [40]).

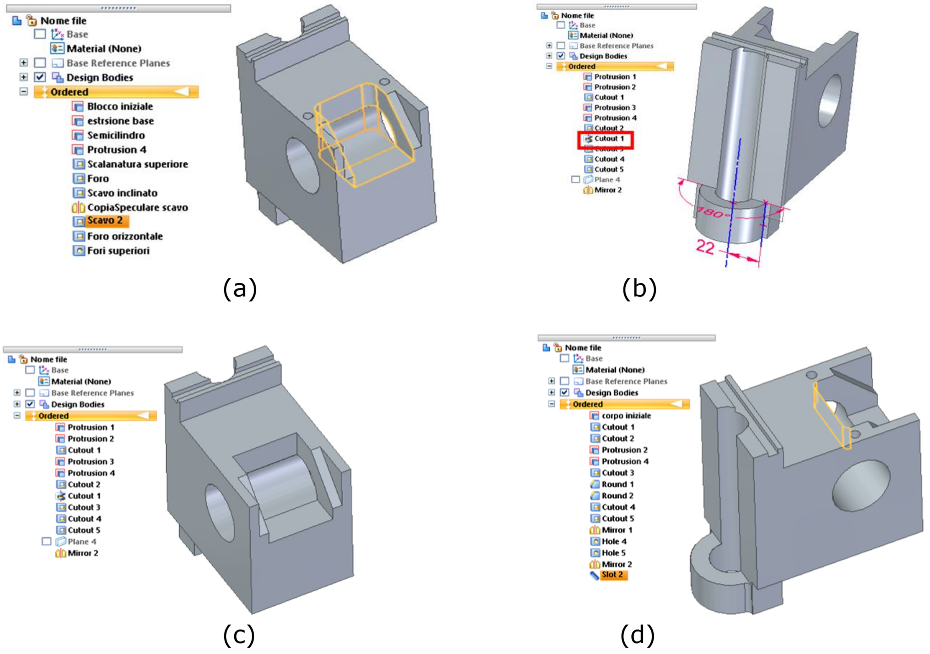
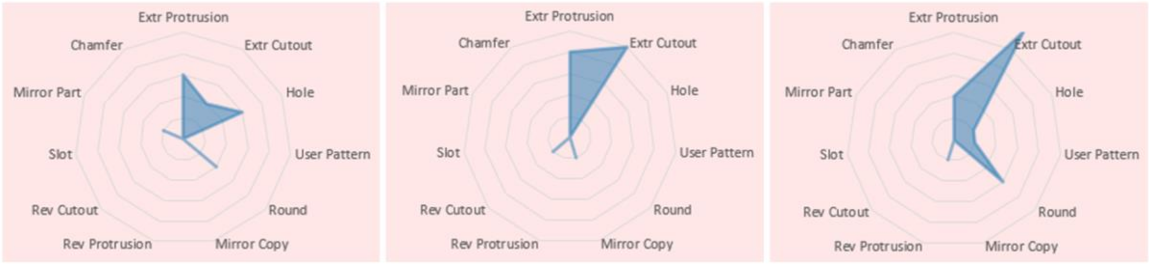


Figure 6: MCAD models with various deficiencies. From left to right: (a) rounds are included in the profile of the central rectangular prism cutout, (b) revolved cutout feature used to create the semi-circular flange, (c) a pair of small holes and all the rounds are missing, (d) slot feature used inside the central rectangular prism cutout.

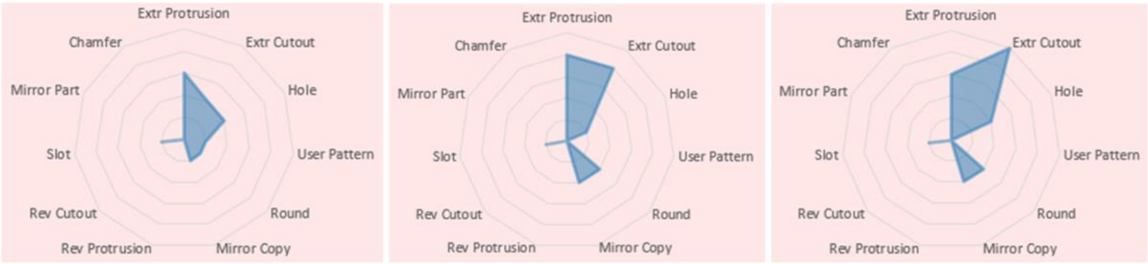
The exceptionally high number of round features shown in Figure 5(b) was found to be due to a modeling strategy where not only was every round created individually with one round feature, but round features were also incorrectly used to create the semi-circular flange. As can be seen in the FBC profile (see again Figure 5(b)), this MCAD model was created using only one extruded protrusion feature, and did not utilize any mirror or hole features. Consequently, this approach required a high number of extruded cutout features — to make up for the design shortcomings in this modeling strategy — to create the MCAD model shape. Inspection of the FBC profile shown in Figure 5(c) immediately reveals that the MCAD model was created with the appropriate numbers of each feature type as recommended, except that round features are obviously missing. A closer model inspection shows that one set of rounds was created by including it in the profile of the central rectangular prism cutout, while the second set of rounds was actually missing (see Figure 6(a)). Such a modeling approach should be avoided, because CAD model details such as rounds and chamfers need to be created with their respective feature types, to enable their suppression during computation-intensive CAD model supported processes such as finite element mesh analysis.



(a) (b) (c)

Figure 7: Various out-of-scope feature types and low numbers of certain in-scope feature types. From left to right: (a) mirror part feature and subpar number of extruded protrusion and cutout features, (b) revolved cutout feature but no round features, (c) revolved protrusion feature and high numbers of extruded cutout features and round features.

Unusual and faulty modeling strategies are easily revealed by visualizations that include out-of-scope features. For example, the FBC profile shown in Figure 7(a) indicates that a mirror part feature was used in combination with a subpar number of extruded protrusion and cutout features. User pattern and mirror copy features were not used at all. Inspecting the linked MCAD model revealed that the reason for this unusual modeling approach was the extensive use of complex sketches to first create the right-hand half of the model and then create the left-hand half through use of the mirror part feature. Figure 7(b) indicates the use of a revolved cutout feature, which was employed to create the semi-circular flange, as shown in Figure 6(b). The FBC profile also shows that no features of type hole, user pattern, or round were used. As can be seen in Figure 6(c), the pair of small holes and all the rounds were missing from the MCAD model shape. As those elements were missing and not created using other features or modeling approaches, it can be assumed that they were simply overlooked by the student. An example where a revolved protrusion feature was wrongly used to model the semi-circular flange is shown in Figure 7(c). The FBC profile also reveals that only 2 extruded protrusion features and 3 round features were used. The former deficiency has arisen because a revolved protrusion feature was used instead of an extruded protrusion feature. The latter problem is due to a modeling pattern that was also found in other MCAD models, which employs one round feature to create the two rounds inside the central rectangular prism cutout and one round feature for each round along the curved flange sides.



(a) (b) (c)

Figure 8: Slot features and exceptional numbers of certain in-scope feature types. From left to right: (a) high number of mirror copy features, (b) high number of extruded cutout features and round features, (c) high number of mirror copy features.

Analysis of the use of the slot feature provided both insight on its use by students in their modeling outcomes and knowledge about mistakes committed by students within this exercise that was new to the authors. In cases where a slot feature was used (see examples in Figure 8), its use was mostly aimed at creating the U-groove (cf. Figure 9(a)) as shown in Figure 8(a), which, however, should be created with an extruded cutout feature. In cases where a slot feature was used, this usually led to a reduced number of extruded cutout features in the model, as shown, for example, in Figure 8(a) and Figure 8(c). However, in the case shown in Figure 8(c), the slot feature was incorrectly used to create the 2 rounds inside the central rectangular prism cutout (see Figure 6(d)). This was the first time that such a modeling mistake of using a slot feature in the context outlined had been detected and revealed. Note that using a slot feature within this exercise context is not considered a fundamental mistake per se. However, it is discouraged and considered out of scope here due to the way in which the geometric properties and dimensions of this feature type are defined, which makes a slot feature less appropriate — especially for novices during this exercise — than an extruded cutout feature with its explicit profile and dimension setting.

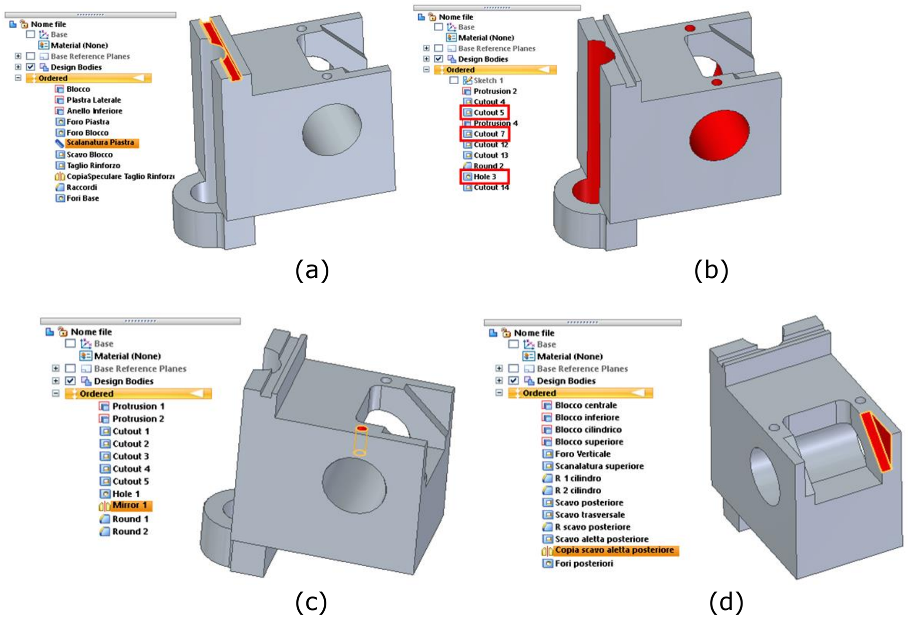


Figure 9: MCAD model with some of its polyhedral and cylindrical cutouts highlighted in red. From left to right: (a) horizontal U-groove, (b) various holes and circular passages, (c) pair of pin holes, (d) pair of triangular prisms.

Next, insight was gained on how polyhedral and circular cutouts (see Figure 9) were created by students using hole, pattern, mirror copy, and extruded cutout features. These cutouts were analyzed using the diagram layout design $DL^{4,2}$ as described earlier. Visualizations which display a FBC profile reaching across all quadrants of the diagram indicate MCAD models created in a structured manner by using the features recommended for this exercise of the type extruded cutout, mirror copy, hole, and user pattern, as shown in the example in Figure 10(a). Modeling strategies that did not include individual hole features usually employed a pattern feature to create the pair of small holes (see Figure 9(b)), and created the remaining circular passages with extruded cutout features. Due to the latter, in those cases the number of extruded cutout features is rather higher than in cases where a more structured and balanced approach (see again Figure

10(a)) was used. In those cases the FBC profile is limited to the lower half of the diagram, as shown in Figure 10(b). Modeling approaches that consist only of extruded cutout features and hole features have a typical FBC profile that is limited to the upper left-hand quadrant of the diagram, as shown in Figure 10(c). Notice that, due to a lack of mirror copy and pattern features, the number of hole features in these cases is usually rather higher than in cases where those feature types have also been used.

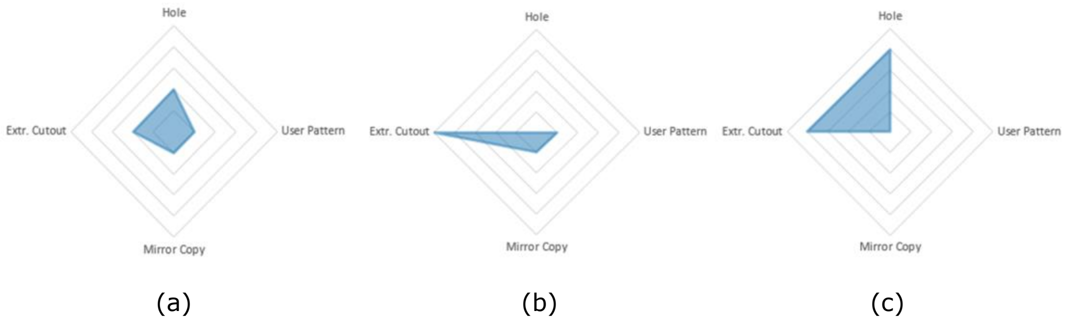


Figure 10: MCAD model creation and the use of features of type hole, user pattern, mirror copy, and extruded cutout. From left to right: (a) subpar number of extruded cutout features, (b) high number of extruded cutout features and no hole features, (c) high number of extruded cutout features and hole features, no features of type mirror copy or user pattern.

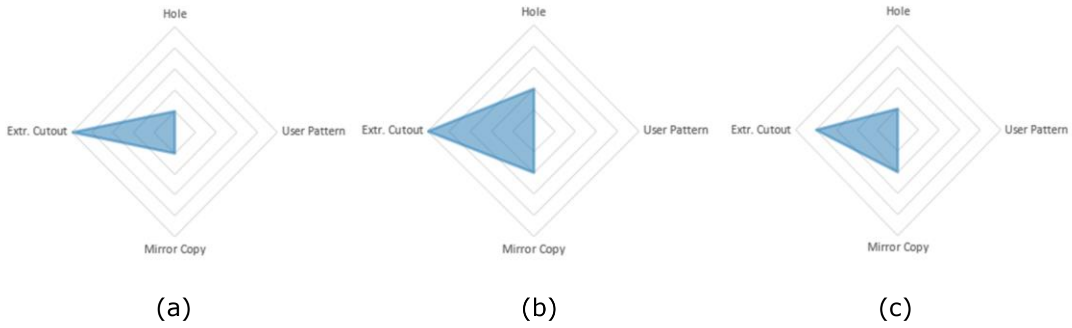


Figure 11: MCAD model creation and the use of features of type hole, user pattern, mirror copy, and extruded cutout. From left to right: (a) high number of extruded cutout features, subpar number of hole features and no user pattern features, (b) high number of extruded cutout features and mirror copy features and no user pattern features (c) high number of extruded cutout features and mirror copy features, subpar number of hole features and no user pattern features.

Alternative modeling strategies where the pair of small holes (see Figure 9(c)) has been created by means of a hole feature and a mirror copy feature have a FBC profile that is limited to the left-hand half of the diagram and features a closed polyline shaped as an isosceles triangle with its base passing vertically through the diagram center, as shown in Figure 11(a). Some variants of this modeling approach, including the use of the mirror copy feature to also create the pair of triangular cutouts (see Figure 9(d)) are shown in Figure 11(b) and Figure 11(c). In those cases the shape of the FBC profile is either an isosceles triangle or a scalene triangle depending on the number of hole features used. Notice that in cases where the number of extruded cutout features is slightly smaller than expected, usually a volume-removing out-of-scope feature has been used.

Modeling strategies that were poorly structured, using a high number of extruded cutout features, are usually graphically represented with FBC profiles limited to the lower left-hand quadrant of the diagram, as shown in Figure 12. For example, in the case shown in Figure 12(a) neither pattern features nor hole features were used. Note that the exceptionally high number of mirror copy features was due to a repeated incorrect use of this feature type, as discussed elsewhere in this paper (see again Figure 5(a)). In this context, a typical example where a mirror copy feature was used only to create the pair of triangular cutouts (see again Figure 9(d)) is shown in Figure 12(b). In cases where the modeling strategy is poorly structured and based mostly on the use of extruded cutout features, the shape of the FBC profile degenerates into a single polyline, as shown, for example, in Figure 12(c).

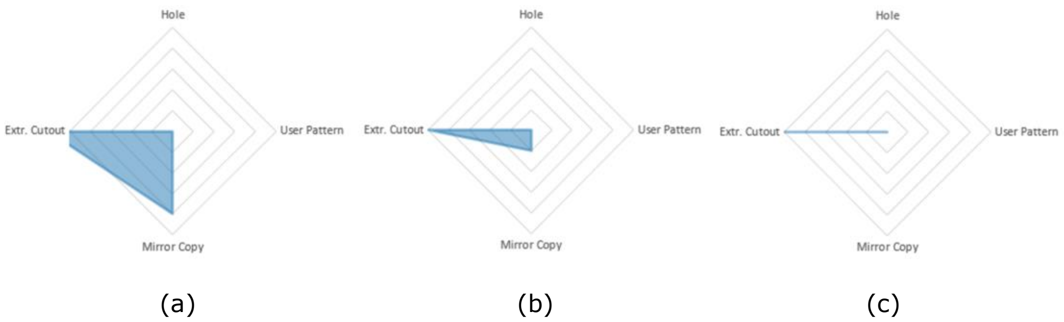
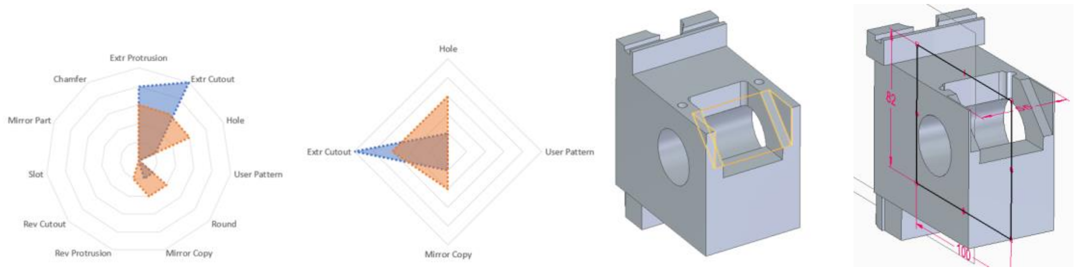


Figure 12: MCAD model creation and the use of features of type hole, user pattern, mirror copy, and extruded cutout. From left to right: (a) exceptionally high number of extruded cutout features and mirror features, no hole features, (b) high number of extruded cutout features, no features of type hole or user pattern, (c) high number of extruded cutout features, no features of type hole, mirror copy or user pattern.

The visualizations and diagram layouts as employed can also be used to analyze the progress that has been made by students and possible shortcomings still remaining in their actual modeling outcome. This is achieved by superimposing the FBC profiles of different versions of the same MCAD model that has been created and submitted. To keep the example concise and transparent, in what follows next the case of just one student model will be described. Two complete versions of this model were submitted — the initial version and the final version.

Qualitative and quantitative changes in the structure of the MCAD model between its initial complete version (FBC profiles in light brownish orange) and its final version (FBC profiles in light blue) are immediately recognizable, as shown in Figure 13(a) and Figure 13(b). As can be discerned from Figure 13(a) the modeling strategy used to create the final version of the MCAD model must have improved considerably compared with its counterpart used for creating the initial version. For example, the final version of the MCAD model does not contain any out-of-scope features. The unusually high number of hole features, round features, and mirror copy features has been adjusted to what is recommended for this exercise. However, to make sure that those changes have actually led to an improvement in the model and its creation, further analysis is required. This reveals that the use of an extruded protrusion feature, instead of a revolved protrusion feature, to create the semi-circular flange has not only eliminated the out-of-scope feature, but also increased the total number of extruded protrusion features to the level recommended for this exercise.



(a) (b) (c)) (d)

Figure 13: MCAD model in its initial and final states of development. From left to right: (a) overview of types and numbers of features used for model creation, (b) overview of features of type hole, user pattern, extruded cutout, and mirror copy used for model creation, (c) final version of MCAD model with one larger triangular prism cutout, (d) model recreation of the final version of MCAD model after parameter changes that introduced dormant deficiencies due to mistakes in feature constraints.

The adjustments in the numbers of hole features and extruded cutout features to properly create the circular passages and holes (see again Figure 9(b)) represent an additional improvement. However, the reduction in the number of mirror copy features has actually decreased the CAD model robustness, and thus quality, due to unfavorable changes in the creation of the pair of small triangular cutouts. In the initial CAD model version these were correctly created using an extruded cutout feature and a mirror copy feature. Unfortunately, in the final model version they were created with one larger triangular prism cutout as shown in Figure 13(c). This modeling strategy makes the CAD model under the feature constraints — as defined by the student — quite unstable in case of changes in feature parameters, resulting in a deficient model recreation (see example in Figure 13(d)), due to the introduction of dormant deficiencies (cf. [43]).

4.3 Analysis of Best Practices and Modeling Guidelines Compliance

4.3.1 Layout design and FBC profile optimization

For the examples presented and discussed in the sub-section on analysis in regard to compliance with best practices and modeling guidelines, the three diagram layout designs which have been used, that is $DL^{5,2}$, $DL^{5,3}$, and $DL^{6,1}$, were structured as follows.

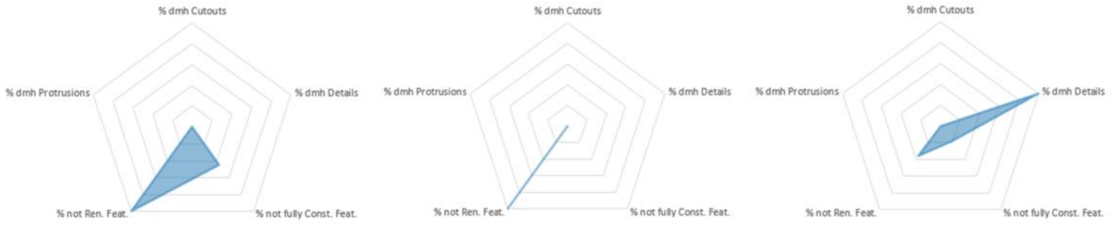
In the case of $DL^{5,2}$, the graphical representation of the FBC space has been based on the encoding of proportions within a pentagon-based layout design. Here the individual diagram axes represent, in a clockwise direction in the upper half property space, volume-adding features, volume-removing features, and detailing features that were properly positioned within the modeling history of the MCAD model. In the lower half property space, the individual diagram axes represent features that were not renamed and features that were not fully constrained.

In the case of $DL^{5,3}$, again a pentagon-based layout design was used for the graphical representation of the FBC space, though the data encoding is based on entity counts. Here the individual diagram axes represent, in a clockwise direction, profile-based features, profile-based features that were fully constrained, and the maximum number of geometric entities, constraints, and dimensions across all profile-based features of the MCAD model. The layout design has been optimized by an appropriate permutation of the diagram axes. This permutation was aimed at having both the axis for profile-based features and the axis for profile-based features that were fully constrained aligned in the upper right-hand diagram sector. Such an optimized layout design improves the efficiency of visual analysis, as cases where all profile-based features are fully constrained can be graphically represented by a polyline segment of the FBC profile in this diagram sector that is collinear to the concentric polygon lines of the diagram layout's regular pentagon.

In the case of $DL^{6,1}$, the graphical representation of the FBC space has been based on the encoding of geometric entity counts within a hexagon-based layout design. Here the individual diagram axes represent, in a clockwise direction beginning in the upper right-hand side of the property space, geometric entities that were used in extruded protrusion features, extruded cutout features, hole features, slot features, revolved protrusion features, and revolved cutout features. The layout design of $DL^{6,1}$ has been optimized by an appropriate permutation of the diagram axes. This permutation was aimed at having all data related to out-of-scope features graphically represented within the left half of the diagram, while all data related to in-scope features were graphically represented within the right half of the diagram. Since the exercise was designed to have the same complexity — that is, the same number of geometric entities — for both extruded protrusion features and extruded cutout features, an additional optimization step has been applied for the upper left-hand diagram sector, similar to the one employed in the diagram permutation outlined above for $DL^{5,3}$. This allows for an effective and efficient graphical representation of the balance in complexity — based on geometric entities used — between extruded protrusion features and extruded cutout features that were used in MCAD models.

4.3.2 Application and empirical results

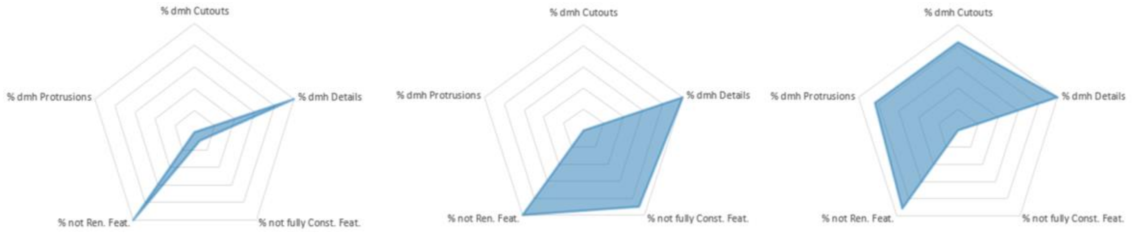
To obtain an overview of, as well as deeper insight into, the nature of compliance with best practice and modeling guidelines in regard to having features renamed and fully constrained, the basic diagram layout design $DL^{5,2}$ can be used. This also enables the structure of the modeling sequence to be examined. Analysis across all diagrams reveals immediately that a failure to rename features and the application of modeling details such as round features at the wrong time during the creation of a CAD model are the predominant shortcomings as shown in the examples in Figure 14 and Figure 15. High proportions of unrenamed features were identified across various models, regardless of the nature and number of deficiencies those models were found to contain.



(a) (b) (c)

Figure 14: Exceptionally high proportion of features not renamed, or round features wrongly placed in the modeling history. From left to right: (a) exceptionally high proportion of features not renamed and subpar proportion of features fully constrained, (b) exceptionally high proportion of features not renamed, (c) exceptionally high proportion of round features wrongly placed in the modeling history, with some features not renamed.

Even in cases where no other deficiencies were detected, high proportions of un-renamed features could sometimes be found (see Figure 14(b)). This includes cases where the high proportion of un-renamed features was the dominant deficiency, together with round features being wrongly placed within the modeling strategy, as shown in Figure 15(a). Examples, typical of cases where those shortcomings appeared in combination with additional deficiencies, such as high proportions of not fully constrained features and extruded cutout features wrongly placed within the modeling strategy, are shown in Figure 15(b) and Figure 15(c).

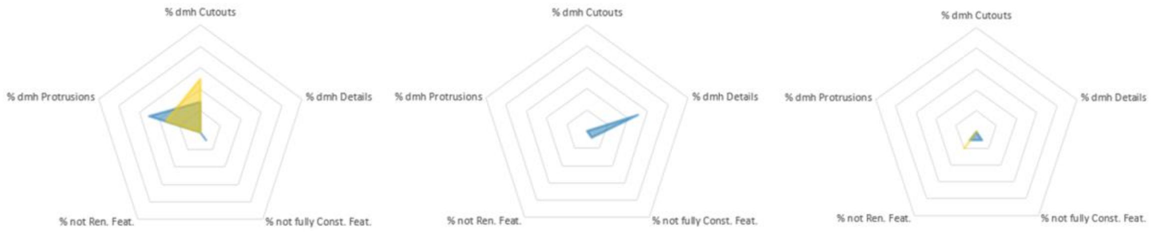


(a) (b) (c)

Figure 15: Exceptionally high proportion of various deficiencies. From left to right: (a) exceptionally high proportion of features not renamed, and round features wrongly placed in the modeling history, (b) exceptionally high proportion of features not renamed and not fully constrained, and round features wrongly placed in the modeling history, (c) exceptionally high proportion in all dimensions, except features not fully constrained.

To gain an insight on how deficiencies developed during the course of model creation, previously discussed FBC profiles (now colored in light blue) have been superimposed on their counterparts computed for the first 5 modeling steps (FBC profiles in light brownish yellow), as applied by students in each of their MCAD models. Note that to account for the fact that volume-removing features should be applied only after all volume-adding features – for this exercise, 4 extruded protrusion features – have been applied, the minimum modeling history length for placing a volume-removing feature needs to be greater than 4.

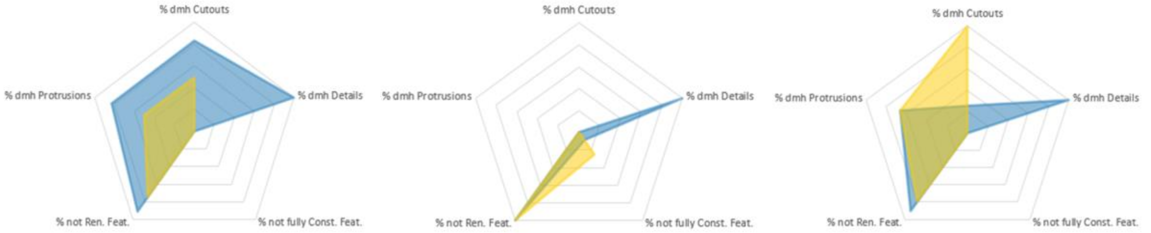
In most MCAD models, the proportions of un-renamed features did not change much during the course of model creation. That is, in cases where features were renamed from the start, students seem to have continued to rename them. In cases where features were not renamed from the start, students seem to have continued in the same manner. However, here, quite unfortunately, they are working with features which are mostly un-renamed.



(a) (b) (c)

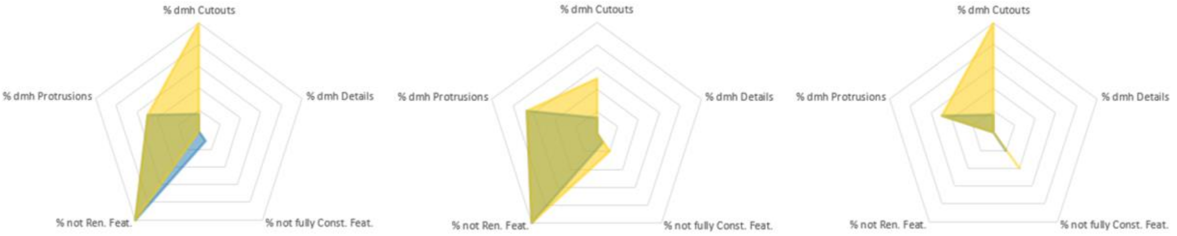
Figure 16: Proportion comparison of various FBC factors across partially created MCAD models. From left to right: (a) reduced proportion of extruded cutout features wrongly placed in the modeling history, (b) no change in proportions, (c) slightly reduced proportion of not fully renamed features.

This trend was observed in both MCAD models with only a few deficiencies, such as those shown in Figure 16, and MCAD models with a high rate of deficiencies, as shown, for example, in Figure 17. However, in the case of extruded cutout features being wrongly placed within the modeling strategy, the situation was different. Here, in most cases, there was a considerable decrease in the exceptionally high proportion of extruded cutout features, which had been wrongly placed in the modeling sequence during the first steps of model creation. This trend was observed both in MCAD models with a high rate of deficiencies, as shown, for example, in Figure 17(c) and Figure 18(a), and in those with only a few deficiencies, such as that shown in Figure 18(c). This indicates that many students start using volume-removing features too early. They need to finish modeling with volume-adding features first, as is recommended by best practice guidelines. In the case of round features being wrongly placed within the modeling strategy, however, the opposite trend was revealed. Here, in cases where MCAD models contained many deficiencies, the initially lower proportions of round features wrongly placed within the modeling sequence increased significantly, as can be seen in Figure 17.



(a) (b) (c)

Figure 17: Proportion comparison of various FBC factors across partially created MCAD models. From left to right: (a) noticeable increase in all proportions, notably in round features wrongly placed in the modeling history, (b) considerable increase in the proportion of round features wrongly placed in the modeling history, (c) considerable change in proportions of both extruded cutout features and round features wrongly placed in the modeling history.



(a) (b) (c)

Figure 18: Proportion comparison of various FBC factors across partially created MCAD models. From left to right: (a) high proportion of un-renamed features, and considerably reduced proportion of extruded cutout features wrongly placed in the modeling history, (b) high proportion of un-renamed features, and reduced proportion of extruded cutout features wrongly placed in the modeling history, (c) considerably reduced proportion of extruded cutout features wrongly placed in the modeling history.

This could be an indicator that students with a weakly structured or otherwise faulty modeling strategy are more likely to add round features to each portion of the MCAD model which they deem completed, instead of adding those shape detail defining features at the very end of the CAD model creation, as taught during the course.

Using the diagram layout design $DL^{6,1}$, some additional insight can be gained into how well students performed in guideline compliance when instructed to use features with non-complex profiles. Figure 19 shows typical graphical representations of cases where extruded cutout features with complex profiles were used. These are immediately revealed through their high values in a polyline along the diagram dimensions of their respective feature types. Complex feature profiles containing a large number of geometric entities were most pronounced within extruded protrusion features and extruded cutout features, as shown, for example, in Figure 19(a) and Figure 19(b). However, there were also cases where complex profiles were found to have been used in both extruded protrusion features and extruded cutout features, as shown in Figure 19(c).

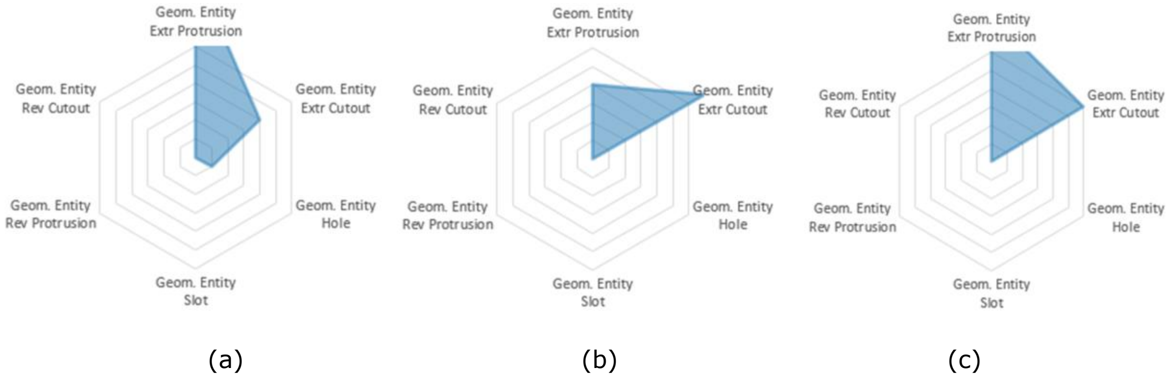


Figure 19: Exceptionally high maximal number of geometric entities used in various feature types. From left to right: (a) exceptionally high number of geometric entities used in extruded protrusion features, (b) exceptionally high number of geometric entities used in extruded cutout features, (c) exceptionally high number of geometric entities used in both extruded protrusion features and extruded cutout features.

Further analysis revealed that over half of the MCAD models that had an unusually high maximal number of geometric entities used in some of their profile-based feature types also had a high number of both un-renamed features and round features placed wrongly within the modeling history. In several cases involving extruded cutout features, this was linked to the previously outlined deficiency of including rounds within profiles (see again Figure 5(c) and Figure 8(a)).

Within this scenario an additional analysis can be conducted on the balance between the number of profile-based features used in a MCAD model and the number of those features that are actually fully constrained. Note that fully balanced MCAD models should always have an equal number of each, as all profile-based features in a model should be fully constrained. Using visualizations based on the diagram layout $DL^{5,3}$, Figure 20 shows typical FBC profiles of cases where an above average maximal number of geometric entities, constraints, and dimensions used in profile-based features had an adverse impact on the number of fully constrained features. According to the data visualized, it appears that a combined increased presence in each of the geometric entities, constraints, and dimensions leads not only to a considerable decrease in the number of fully constrained features, but also to a reduced number of profile-based features, as shown in Figure 20(c) for example.

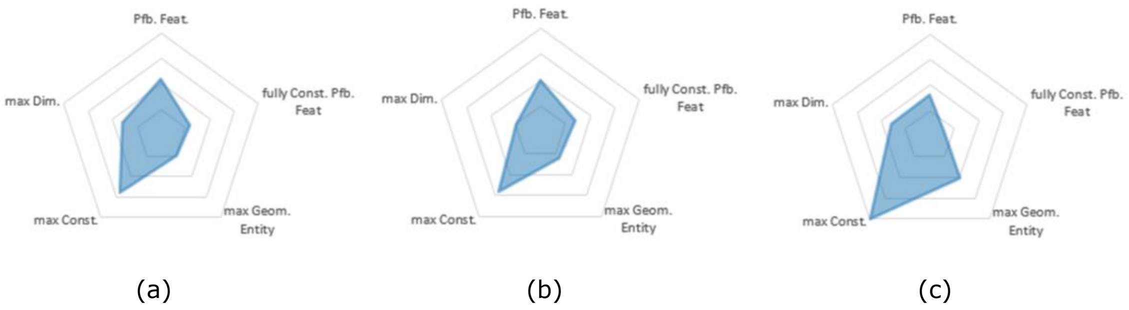


Figure 20: Unusual maximal number of geometric entities, constraints, and dimensions used in profile-based features. From left to right: (a) above average maximal number of constraints and dimensions, (b) above average maximal number of dimensions and geometric entities, (c) above average maximal number of geometric entities, constraints, and dimensions.

Returning to the example of the student model discussed in the previous sub-section (see again Figure 13), qualitative and quantitative changes have been made in the structure of the MCAD model between its initial complete version and its final version, in regard to best practices and modeling guidelines compliance. These are easily recognizable in Figure 21 and Figure 22. As can be discerned from Figure 21(a), the modeling strategy used to create the final version of the MCAD model has significantly improved in some respects over its counterpart used for creating the initial version. For example, the final version of the MCAD model contains only features that are properly renamed, while all profile-based features are now fully constrained. Notice, however, that the proportion of extruded protrusion features and extruded cutout features that are wrongly placed within the model creation history remains unchanged. Comparison of FBC factors across partial models (FBC profiles in light electric magenta and light brownish yellow) reveals some minor improvements in the initial MCAD model version related to a small proportion reduction in not fully constrained features in the latter part of the modeling history (see Figure 21(a) and Figure 21(b)). However, in the final MCAD model version, most improvements outlined earlier have already been achieved in the early phase of the model creation.

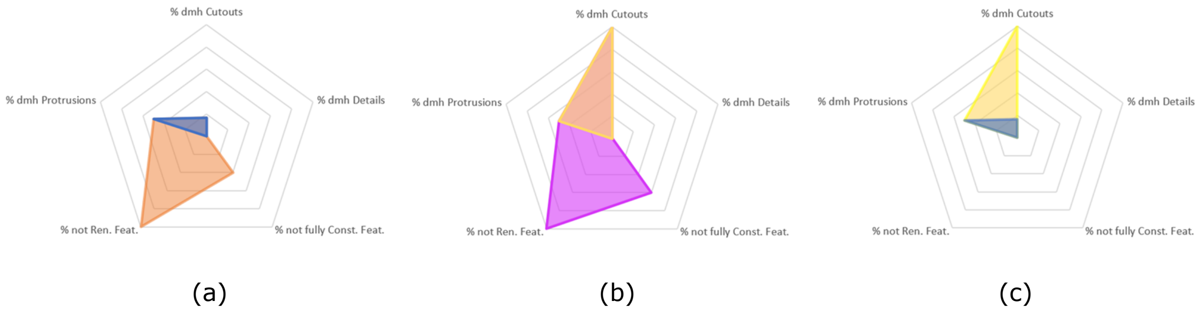
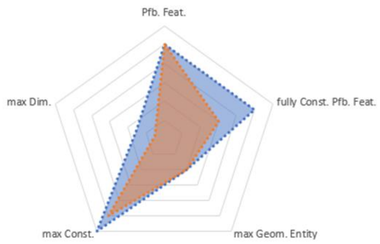
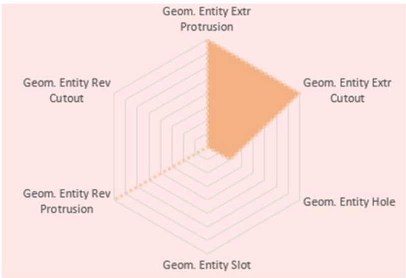
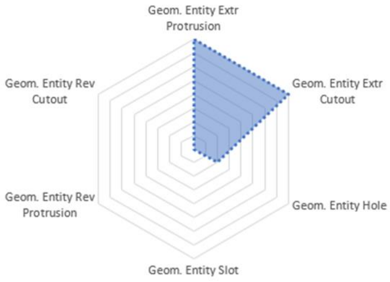


Figure 21: Proportion comparison of various FBC factors across 2 MCAD model versions. From left to right: (a) comparison of FBC factors across the initial and the final model versions, (b) comparison of FBC factors across partial models of the initial and the final model versions, (c) comparison of FBC factors across partial models of the final model version.







(a) (b) (c)

Figure 22: Comparison of various FBC factors between 2 MCAD model versions. From left to right: (a) comparison of maximal number of geometric entities, constraints, and dimensions that were used in profile-based features within the initial and the final model versions, (b) comparison of maximal number of geometric entities used in profile-based features within the initial model version, (c) comparison of maximal number of geometric entities used in profile-based features within the final model version.

That is to say that those problems were mostly solved, while improvements in regard to the proportion of extruded protrusion features and extruded cutout features that were wrongly placed within the model creation history have been achieved more toward the end of the model creation (see Figure 21(b) and Figure 21(c)).

A more detailed view of the nature of improvements between the initial and the final MCAD model versions in regard to the number of profile-based features that were fully constrained and the maximal number of geometric entities, constraints, and dimensions used in profile-based features is shown in Figure 22(a). Improvements in the modeling strategy leading to qualitative adjustments regarding the feature scope are immediately visible, as out-of-scope features (see Figure 22(b)) were eliminated from the final model version (see Figure 22(c)). Improvements in the modeling strategy leading to quantitative adjustments in all FBC factors in Figure 22(a) have resulted in an overall well-balanced MCAD model where all profile-based features are fully constrained. Note that those improvements have been achieved while the complexity of the profile-based features used for model creation remained unchanged and within the range recommended for this exercise as shown in Figure 22(b) and Figure 22(c).

5 CONCLUSIONS AND FUTURE WORK

Within the work presented, the approach, framework, and structures used for the design and development of a novel and advanced interactive visualization system for feature-based MCAD model characteristics have been outlined and discussed. Promising outcomes achieved, based on the assessment of empirical results from functionality and reliability checks and experimental evaluation, were as follows. First, the graphical representation of feature-based MCAD model characteristics, as implemented and provided within the interactive visualization tool component, supported efforts to increase cognitive productivity and throughput during model analysis and assessment. With the interactive visualization serving as an interface between the human MCAD model assessor and the computer system, casting of visual patterns, which is required for visual queries, enabled model analysis in a way that would be impossible to achieve without such a computerized visual aid. Second, the interactive visualization-based approach also facilitated the discovery of meaningful and previously unknown knowledge during computer-aided MCAD model analysis and subsequent formative feedback creation. In particular, insight on data related to model deficiencies and errors was considerably increased, as well as elevated, through enabling the detection of relations, patterns, and trends, which otherwise would have remained largely overlooked. Third, in regard to these discoveries, unprecedented possibilities were opened up by revealing links between individual types of MCAD model deficiencies and particular patterns and trends that could be associated with errors and mistakes committed by students during model creation. This, in turn, enabled the formation of pointers toward more evidence-driven and student-needs-oriented improvements to the current MCAD course and its exercises and CAD laboratory assignments. Based on the promising outcomes achieved so far, an experimental full integration of the approach, with implementation within the model analysis and assessment processes of the current MCAD course, is planned for the next academic year.

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

REFERENCES

[1] Arnheim, R.: Gestalt Psychology, in: Craighead, W. E., Nemeroff, C. B. (eds.), The Concise Corsini Encyclopedia of Psychology and Behavioral Science, John Wiley and Sons, Hoboken, NJ, USA, 2004, 401-402.
[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] Birch, J.: Diagnosis of Defective Colour Vision, Butterworth-Heinemann, Boston, MA, USA, 2001.
[4] Boud, D.; Molloy, E.: Feedback in Higher and Professional Education: Understanding it and doing it well, Routledge, New York, NY, USA, 2013.
[5] Brooks, J. L.: Traditional and New Principles of Perceptual Groupings, in: Wagemans, J. (ed.), Oxford Handbook of Perceptual Organization, Oxford University Press, New York, NY, USA, 2015, 57 - 87.
[6] 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
[7] Butler, D. L.; Winne, P. H.: Feedback and self-regulated learning: A theoretical synthesis, Review of Educational Research, 65 (3), 1995, 245-281. https://doi.org/10.3102/00346543065003245
[8] Carless, D.: Differing perceptions in the feedback process, Studies in Higher Education, 31(2), 2006, 219-233. https://doi.org/10.1080/03075070600572132
[9] Carless, D; Boud, D.: The development of student feedback literacy: Enabling uptake of feedback, Assessment & Evaluation in Higher Education, 43 (8), 2018, 1315-1325. https://doi.org/10.1080/02602938.2018.1463354
[10] Carless, D.: Feedback loops and the longer-term: Towards feedback spirals, Assessment & Evaluation in Higher Education, 44 (5), 2019, 705-714. https://doi.org/10.1080/02602938.2018.1531108
[11] Cizek, G. J.; Andrade, H. L.; Bennett, R. E.: Formative Assessment: History, Definition, and Progress, in: Andrade, H. L., Bennett, R. E., Cizek, G. J. (eds.), Handbook of Formative Assessment in the Disciplines, Routledge, New York, NY, USA, 2019, 3-19.
[12] Conway, J. H.; Burgiel, H.; Goodman-Strauss, C.: The Symmetries of Things, CRC Press, Roca Baton, FL, USA, 2008.

[13] Corrin, L.; De Barba, P.: How do students interpret feedback delivered via dashboards?, in: Proceedings of International Conference on Learning Analytics and Knowledge, March 16 - 20, Poughkeepsie, NY, USA, 2015, 430 - 431. https://doi.org/10.1145/2723576.2723662
[14] Dawson, P.; Henderson, M.; Ryan, T.; Mahoney, P.; Boud, D.; Phillips, M.; Molloy, E.: Technology and Feedback Design, in: Spector, M. J., Lockee, B. B., Childress, M. D. (eds.), Learning, Design, and Technology, Springer, Cham, Switzerland, 2018, 1- 45. https: //doi.org/10.1007/978-3-319-17727-4_124- 1
[15] Draper, G. M.; Livnat, Y.; Riesenfeld, R. F.: A survey of radial methods for information visualization, IEEE Transactions on Visualization and Computer Graphics, 15(5), 2009, 759-776. https://doi.org/10.1109/TVCG.2009.23
[16] Evans, C.: Making sense of assessment feedback in higher education, Review of Educational Research, 83 (1), 2013, 70–120. https://doi.org/10.3102/ 0034654312474350
[17] Eysenck, M.: Fundamentals in Cognition, Psychology Press, Hove, UK, 2018.
[18] 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.
[19] Faulconer, E.; Griffith, J.; Gruss, A.: The impact of positive feedback on student outcomes and perceptions, Assessment & Evaluation in Higher Education, 47 (2), 2022, 259–268. https://doi.org/10.1080/ 02602938.2021.1910140
[20] Gašević, D.; Dawson, S.; Siemens, G.: Let's not forget: Learning analytics are about learning, TechTrends, 59 (1), 2015, 64 -71. https: //doi.org/10.1007/s11528- 014- 0822- x
[21] 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.
[22] Harrower, M.; Brewer, C. A.: 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
[23] Hattie, J.; Timperley, H.: The power of feedback, Review of Educational Research, 77(1), 2007, 81-112. https://doi.org/10.3102/003465430298487
[24] Hekman, K. A.; Gordon, M. T.: Automated grading of first year student CAD work, ASEE Computers in Engineering Journal, 24(2), 2014, 16-24.
[25] Hu, G.; Pan, Z.; Zhang, M.; Chen, D.; Yang, W.; Chen, J.: An interactive method for generating harmonious color schemes, Color Research and Application, 39(1), 2014, 70–78. https://doi.org/10.1002/col.21762
[26] Ingale, S.; Anirudh, S.; Bairaktarova, D.: CAD platform independent software for automatic grading of technical drawings, in: Proceedings of the 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.
[27] Irons, A.: Enhancing Learning through Formative Assessment and Feedback, Routledge, Abingdon, UK, 2008.
[28] 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
[29] Jenny, B.; Kelso, N. V.: Color Design for the color vision impaired. Cartographic Perspectives, 58, 2007, 61-67. https://doi.org/10.14714/CP58270
[30] Johnson, W. R.: Detecting plagiarism in SolidWorks CAD courses, in: Proceedings of the ASEE Annual Conference and Exposition, June 23 -27, Salt Lake City, UT, USA, 2018, Paper-No.: 23898. https://doi.org/10.18260/1-2--30286
[31] Jonsson, A.: Facilitating productive use of feedback in higher education, Active Learning in Higher Education, 14 (1), 2013, 63-76. https://doi.org/10.1177/ 1469787412467125
[32] Joo, S.-H.: Assessment of three dimensional CAD models using CAD application programming interface, in: Proceedings of the ASME International Mechanical Engineering Congress and

Exposition, November 9 -15, Pittsburgh, PA, USA, 2018, Paper-No.: IMECE2018-87776. https://doi.org/10.1115/IMECE2018-87776
[33] Khaleel, M. F.; Sharkh, M. A.; Kalil, M.: *A cloud-based architecture for automated grading of computer-aided design student work using deep learning,* in: Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, August 30 - September 2, London, ON, Canada, 2020, pp. 1-5. https://doi.org/10.1109/CCECE47787.2020.9255825
[34] 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.
[35] Koffka, K.: Principles of Gestalt Psychology, Hardcourt, Brace and Company, New York, NY, USA, 1935.
[36] Kolence, K. W.; Kiviat, P. J.: Software unit profiles and Kiviat figures, ACM SIGMETRICS Performance Evaluation Review, 2(3), 1973, 2–12. https://doi.org/10.1145/1041613.1041614
[37] Kung, F. Y. H.; Scholer, A.: Message framing influences perceptions of feedback (in)directness, Social Cognition, 36 (6), 2018, 626-670. https://doi.org/10.1521/soco.2018.36.6.626
[38] Leu, J.-G.: Computing a shape's moments from its boundary, Pattern Recognition, 24(10), 1991, 949-957. https://doi.org/10.1016/0031-3203(91)90092-J
[39] Liu, D. Y.-T.; Bartimote-Aufflick, K.; Pardo, A.; Bridgeman, A. J.: Data-Driven Personalization of Student Learning Support in Higher Education, in: Peña-Ayala, A. (ed.) Learning Analytics: Fundaments, Applications, and Trends. Studies Systems, Decision and Control, Vol. 94, Springer, Cham, Switzerland, 2017. https://doi.org/10.1007/978-3-319-52977-6_5
[40] Lv, X.; He, F.; Yan, X.; Wu, Y.; Cheng, Y.: Integrating selective undo of feature-based modeling operations for real-time collaborative CAD systems, Future Generation Computer Systems, 100, 2019. https://doi.org/10.1016/j.future.2019.05.021
[41] Mandorli, F.; Borgo, S.; Wiejak, P.: From form features to semantic features in existing MCAD: An ontological approach, Advanced Engineering Informatics, 44, 2020, 101088. https://doi.org/10.1016/j.aei.2020.101088
[42] Mandorli, F.; Otto, H. E.: A Systematic Approach to Innovative MCAD Education Based on Negative Knowledge Development and Formative Feedback, in: Rizzi, C., Campana, F., Bici, M., Gherardini, F., Ingrassia, T., Cicconi, P. (eds.), Lecture Notes in Mechanical Engineering (Design Tools and Methods in Industrial Engineering II), Springer, Cham, Switzerland, 2021, 839-850. https://doi.org/10.1007/978-3-030-91234-5_85
[43] Mandorli, F.; Otto, H. E.: Improving the learning experience within MCAD education: A tool for students to assist in self-assessment during modeling exercises, Computer-Aided Design and Applications, 19(3), 2022, 534-560. https://doi.org/10.14733/cadaps.2022.534-560
[44] Meier, B. J.; Spalter, A. M.; Karelitz, D. B.: Interactive color palette tools, IEEE Computer Graphics and Applications, 24(3), 2004, 64–72. https://doi.org/10.1109/MCG.20041297012
[45] Mollon, J. D.; Pokorny, J.; Knoblauch, K.: Normal and Defective Colour Vision, Oxford University Press, New York, NY, USA, 2003.
[46] Munzner, T.: Visualization, Analysis and Design, CRC Press, Boca Raton, FL, USA, 2014.
[47] 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
[48] 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
[49] Otto, H. E.; Mandorli, F.: Advancing formative assessment in MCAD education: The visual analytics of parametric feature-based solid models, Advanced Engineering Informatics, 48, 2021, 101308. https://doi.org/10.1016/j.aei.2021.101308

[50] Pardo, A.; Jovanovic, J.; Dawson, Gašević, S. D.; Mirriahi, N.: Using learning analytics to scale the provision of personalised feedback, British Journal of Educational Technology, 50 (1), 2019, 128-138. https: //doi.org/10.1111/bjet.12592

[51] Pinzger, M.; Gall, H.; Fischer, M.; Lanza, M.: Visualizing Multiple Evolution Metrics, in: Proceedings of the ACM Symposium on Software Visualization, May 14-15, St. Louis, MO, USA, 2005, pp. 67–75. https://doi.org/10.1145/1056018.1056027

[52] Pitt, E.; Norton, L.: 'Now that's the feedback I want!' Students' reactions to feedback on graded work and what they do with it, Assessment & Evaluation in Higher Education, 42 (4), 2017, 499-516. https://doi.org/10.1080/ 02602938.2016.1142500

[53] Schlatter, T.; Levinson, D.: Visual Usability: Principles and Practices for Designing Digital Applications, Morgan Kaufmann, Waltham, MA, USA, 2013.

[54] Shute, V. J.: Focus on formative feedback, Review of Educational Research, 78(1), 2008, 153-183. https://doi.org/10.3102/0034654307313795

[55] Stone, M. C.: A Field Guide to Digital Color, A. K. Peters / CRC Press, Boca Raton, FL, USA, 2003.

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

[57] Wertheimer, M.: Investigations on Gestalt principles, in: Spillmann, L. (ed.), On Perceived Motion and Figural Organization, MIT Press, Cambridge, MA, USA, 2012.

[58] Winstone, N. E.; Nash, R. A.; Rowntree, J.; Michael Parker, M.: 'It'd be useful, but I wouldn't use it': Barriers to university students' feedback seeking and recipience, Studies in Higher Education, 42 (11), 2017, 2026-2041. https://doi.org/10.1080/03075079.2015.1130032

[59] Younes, R.; Bairaktarova, D.: ViTA: A flexible CAD-tool-independent automatic grading platform for two-dimensional CAD drawings, International Journal of Mechanical Engineering Education, 50(1), 2022, 135-157. https://doi.org/10.1177/0306419020947688

[60] Zanabria, G. G.; Nonato, L. G.; Gomez-Nieto, E.: iStar(i*): An interactive star coordinates approach for high-dimensional data exploration. Computers and Graphics, 60, 2016, 107-118. https://doi.org/10.1016/j.cag.2016.08.007

[61] Zhang, D.; Lu, G.: Review of shape representation and description techniques, Pattern Recognition, 37(1), 2004, 1-19. https://doi.org/10.1016/j.patcog.2003.07.008