COMPUTER-AIDED DESIGN & APPLICATIONS, 2017
VOL. 14, NO. 3, 269–283
http://dx.doi.org/10.1080/16864360.2016.1240448

Integration of negative knowledge into MCAD education to support competency development for product design

Harald E. Otto and Ferruccio Mandorli

Polytechnic University of Marche, Italy

ABSTRACT
With a steadily increasing use of CAD systems within digital prototypes in product design and development, the capacity to create viable geometric models that can be used in various computer-aided engineering processes is becoming an indispensable necessity. However, with the current trend of commercial CAD systems increasingly to promote hybrid geometric modeling environments, this requirement poses a new challenge for education, as it requires a teaching strategy that goes beyond the sum of subject learning in surface modeling and solid modeling. In this paper, a novel teaching approach is introduced, which integrates negative knowledge as one crucial element in combination with traditional teaching methods to support competency development that reaches beyond the acquisition of basic modeling skills and domain knowledge.

KEYWORDS
Negative expertise; learning from errors; transversal abilities; curriculum design; hybrid geometric modeling; shape engineering

1. Introduction

Nowadays computer-aided design (CAD) systems are deployed widely in the field of industrial engineering. They are used to create virtual prototypes, in order to support designers during decision-making activities, while also being utilized for product documentation purposes. Depending on the specific application domain, a virtual prototype can be defined in slightly different ways. However, one key characteristic, among several others, which is shared between the different prototype definitions, is that the virtual prototype is always built around a geometric model. In order to be suitable and remain viable throughout the whole product development process, virtual prototypes need to be shared among different systems of the same type, such as different CAD systems or different modeling modules within the same system (see also discussions in [18]). The same also applies for different types of systems such as systems for CAD, FEM and CAE.

The problem of sharing virtual prototypes is not a mere matter of data exchange, which could then be related mostly to the definition of appropriate file formats, which have been developed and significantly improved during the past three decades. In fact, many issues of computer-based exchange and sharing of digital virtual prototypes are intrinsically related to the characteristics of the core geometric model and the possibility of mapping or converting one individual geometric model representation into another. The problem is that modeling systems usually employ a particular type of geometric model or geometric model representation, which is in most cases different for each system. Examples of different geometric model types and related geometric model representations include exact solid models with boundary representation, approximated solid models based on volume sub-division representation, exact surface models based on either a NURBS/spline representation or a sub-division surfaces representation, approximated surface models based on polygonal or triangular mesh representation, and even hybrid models, where different representations co-exist within one and the same modeling system. These examples constitute the majority of types currently to be found in industrial practice.

Due to the situation outlined above, competency in the geometric modeling required to support modern computer-aided product design and development requires domain knowledge and modeling skills that reach beyond defining nice-looking models that feature the right shape and dimensions. In this context, of increasing importance and relevance is the capability to develop models that can be effectively and efficiently mapped from one representation to another: in other words, to be suitable, as well as remaining viable, for further use in downstream tasks of design and production

CONTACT Harald Otto h.e.otto@univpm.it; Ferruccio Mandorli f.mandorli@univpm.it

processes. The diffusion and application of CAD systems in industrial engineering have been supported by, and have benefited from, the introduction of specific CAD courses at engineering faculties of institutions of higher education (cf. [3, 23, 24]). In many cases, those courses are based on laboratory and practical exercises, aimed at providing training in the use of a geometric modeling system. However, adequately addressing the most challenging objective, namely teaching how to produce models that are *well designed*, to be used within the product development process, still has many shortcomings. This represents an unfortunate situation that arises mainly because it is not always evident how to convey the required information to a student body, which in the case of CAD courses at engineering faculties, lacks the geometric model representation and computer science background that is required to fully understand the technicalities related to the translation of a model from one representation to another.

In this paper a novel teaching framework is proposed, which employs negative knowledge, also including means for developing situation awareness based on learning how to recognize critical modeling situations and prevent actions that may lead to deficiencies rendering a CAD model useless for subsequent engineering tasks. Among the advantages of the proposed approach is that the students do not need to become experts in geometric model representation in order to be aware of and understand the different types of possible shortcomings affecting model sharing. The remainder of the paper is organized as follows. In Section 2 some background is given together with an outline of scope and objectives. This is followed by discussions on negative knowledge and expertise along with the introduction of a novel approach and newly developed concepts central to its framework in Section 3. Next, in Section 4 a compilation of application examples related to the context of MCAD education is presented. A brief summary with conclusions and the outlook for future work is given in Section 5.

2. Background, scope and research objectives

2.1. Current CAD education and hybrid modeling environments in practice

Modern hybrid geometric modeling systems are a typical example of CAD systems sharing multiple model representations. In the case of commercially available systems, these usually consist of NURBS-based surface models and B-rep solid models. Over the last decade, the development of hybrid modeling systems has increased significantly. One major reason for such development is related to the support of several different design processes, such as shape engineering, mold and cavities design for injection molding processes, and fixture design, which can be provided by solid-surface hybrid models and related hybrid modeling commands. Typical modeling operations of a hybrid geometric modeling environment are comprised of the extraction of surfaces from solids, the conversion of surfaces into bulk or sheet solids, and the interoperability between the two different types of models, using, for example, surfaces to cut solids and vice-versa. However, in order to successfully apply such kinds of modeling operations, some basic geometric requirements must be preserved.

Unless the hybrid model is geometrically sound, it is impossible to proceed further toward more specialized models to be used for analysis and simulation. In particular, one critical aspect is related to the conversion of a surface model into a solid model. This model conversion is usually achieved by means of two main types of modeling commands, namely *add thickness*, which adds a thickness to a surface to convert it into a sheet solid, and *make solid*, which converts a set of closed surfaces that define a bounded volume, into a solid. In both cases, the main issue for the underlying conversion algorithms is the *quality* of the surface to be converted. In this context, the *quality* of the surface depends on the correct sewing among the patches that define the surface and the level of continuity both of each single patch and between adjacent patches. In order to preserve the surface quality, individual modeling systems adopt different approaches. One method represents a *strict* approach, which prevents a command from generating a *low quality* surface if the input is not adequate. When the system users are novices, this method has a tendency to lead to frustrating situations of their not understanding why the application of commands for the conversion of a surface model into a solid model failed. Another method represents more a *relaxed* approach, where any kind of surface can be generated, leaving the quality control to the user. Again, when the users are novices, this method has a tendency to lead to the assumption that the model created is of good quality, while actually it is not. At this point it needs to be highlighted that a model of overall good quality is not the automatic and sole result of a sequence of correctly applied and executed geometric modeling commands, but is profoundly related to the correct design of the geometric model employing an appropriate modeling strategy.

Traditional curricula in CAD education, like in many other engineering disciplines, were mainly concerned with the dissemination and development of procedural knowledge and skills in the form of *knowing that* or *knowing what* related to the operation of CAD systems (cf. [12, 23]). Unfortunately, such an educational philosophy does not provide the pedagogy necessary to support development of geometric modeling competency. It fails to help develop and instill basic expertise for recognizing critical modeling situations and sufficient strategic knowledge relating to the know-how of design/modeling strategies and how to choose between them. For example, in the context of hybrid modeling, when novices approach the geometric modeling process, quite commonly the awareness of the deficiencies introduced into CAD models by the implementation of a wrong strategy arises too late to avoid critical situations and severe errors, preventing the model from being used in the downstream tasks. A typical situation where this happens is during the shape engineering process, when the external (aesthetic) layer, the so-called *skin* of an object, is first modeled as a surface and then transformed into a solid shell with thickness, to which appropriate features are added employing a typical solid modeling approach, in order to obtain the final geometry of the component. If the skin surface was not modeled with the right accuracy, and geometric properties as required, the commands aimed at adding thickness to the model will fail. In such a context, novices are not able to proceed with the modeling process and have no knowledge on how to recover. In such cases, most of the time, novices do not even have a clue as to why they failed or where they made a mistake.

2.2. Competency and negative knowledge

In most professional occupations, including engineering, expertise consists of acquired skills and knowledge in a specific domain (cf. [8, 21]). In general, experts, in contrast to novices, exhibit a tendency to organize their knowledge within a holistic framework allowing for a fast perception of the significance of situations and possible consequences of actions (cf. [17]). With increased expertise in a domain, cognitive processes become more and more responsive to situational cues, rather than being determined by abstract rules (see also [2, 12, 22]). The ability to do something well and being capable of solving problems, based on acquired skills and knowledge, i.e. expertise, represents an essential component of performance and achievement in a given domain and is conceived as competence (see also discussions in [7, 8]). Performing efficiently while committing almost no serious mistakes, i.e. knowing how to avoid grave errors and approaches that are inefficient in certain situations, is an essential feature of professional engineering expertise. This knowing what not to do in certain situations is attributed to knowledge referred to as *negative knowledge*.

Research undertaken and studies on the theoretical foundations and concepts of negative knowledge can be traced back to work in three different fields. In artificial intelligence, Minsky [10,11] argues, in his work on negative expertise, that a great deal of what experts know about how to achieve goals and how to avoid disasters lies in knowing about what can go wrong in their domain and which actions might cause trouble and are thus better avoided. In education, the work of Oser and Spychinger [13] on the practice of error culture uses a contrastive approach to define negative knowledge as a type of knowledge that relates to information on false facts and inappropriate action strategies. This approach can be seen as pointing towards negative knowledge as a form of meta-knowledge revealing a regulative impact on positive knowledge. In the examples discussed in their work, the authors also stress the importance of practical experience within a concrete work context, as that is the primary method of obtaining negative knowledge (see also [9,14]). In knowledge management, the work of Parviainen and Eriksson [16] focused on the declarative aspect of negative knowledge, the *knowing what not to know*, which is in contrast to the by nature more procedural aspect of *knowing what (not) to do*. In their work they distinguished two types of not-knowing relating to the informed and uninformed methods of an individual lacking knowledge relevant to expertise. This distinction addresses in the former case an awareness by the individual of his lack of relevant knowledge, while the latter case supposes both a lack of relevant knowledge and a lack of awareness of this very fact (see also discussions in [1,6]). More details on the declarative and procedural aspects of negative knowledge can be found in recent work by Gartmeier et al. [4,5]. This work discusses relationships with meta-cognition, and the epistemic potential to enable new insights into various knowledge-related and learning-related fields. It also considers the support given to improving certainty in how to proceed in a task, to increasing efficiency during performance, and to enhancing the depth and quality of reflection on actions and performance.

2.3. Aims for disclosing an innovative departure for MCAD education

The traditional approach to CAD education is based on the teaching of system commands, the interaction with user interfaces, domain subject tutorials, and best practices, with the overall aim of developing sufficient domain knowledge, know-how, and skills to operate a modern CAD system. However, from an educational point of view, the issues related to hybrid geometric modeling represent a new challenge, as they require innovative teaching methodologies capable of supporting the development of strategic know-how and basic domain expertise, which are beyond their counterparts in individual geometric modeling fields. One of the major drawbacks of the traditional teaching approach is that when students have to face new modeling situations, not explicitly encountered during training, due to their being novices, they usually do not recognize that certain strategies may lead to design and modeling situations best avoided. This is due to the fact that tutorials and best practices usually teach "*what to do*" (positive knowledge), though in many situations being aware of "*what not to do*" (negative knowledge) becomes equally important to achieving a desired outcome.

In this paper an innovative teaching approach is presented, which aims at overcoming shortcomings as outlined above, while facilitating the development of strategic know-how. The novelty of the proposed method lies in the integration of traditional methods, which are based on positive knowledge, with aspects related to negative knowledge. Here the use of negative knowledge is aimed at supporting the development of both the awareness of critical situations and the capacity and expertise to know not only how not to select inappropriate actions, but also how to avoid critical situations. One key aspect of the work presented is related to the definition of concepts and structures within the novel framework. This key aspect addresses the elements and structures, which form negative knowledge that can, eventually, be translated into negative expertise.

3. Framework and concepts

3.1. Outline and approach

The design and realization of a novel teaching approach, which integrates negative knowledge as one crucial element within current MCAD education, have been approached by addressing concept mapping, framework development and implementation as follows. First, in order to define what constitutes an error or mistake to be avoided in respect to a particular situation and the quality of a CAD model, some elements of the concept of negative knowledge have been mapped to the concept of geometric model deficiency. This concept is used to form normative knowledge as a qualitative measure to help express certain characteristics of situations during modeling. These characteristics usually lead to models being poorly structured and are thus best avoided.

Second, by formulating negative knowledge as an element of strategic knowledge aimed at developing awareness of and constraining actions within critical situations that would otherwise lead to errors and mistakes, the concept of a situation has been formed as a set of relations associated with particular sets of model configurations, action constraints, anticipated failures, and individual goals and sub-goals. In this context, of particular interest were significant model configurations, which describe a model configuration in a certain context that is significant in respect to action constraints, which in turn are associated with individual actions. These significant model configurations can be related through a mapping to concrete constraints limiting the actions possible in a particular situation.

Third, to implement the approach and integrate it into the current MCAD course, besides traditional lectures and tutorials, various modeling exercises are provided, which are individually designed for different learning aspects (see Figure 1). Results of the exercises are collected and assessed, to identify shortcomings and errors, which usually remain hidden from students due to their limited domain knowledge and expertise. Representative examples of the assessed exercises are later used to discuss, during lectures and also online through the MCAD course web site, issues relating to critical modeling situations, model deficiencies and how to prevent them. In certain cases, if feasible, they are also used to demonstrate how to initiate a recovery.

3.2. Knowledge Concepts

Within the framework developed, two forms of knowledge are used, namely positive knowledge and negative knowledge. In this sub-section, details of their concepts and characteristic attributes along with their relationships to elements of geometric models as well as modeling tasks are presented and discussed.

3.2.1. Concept of positive knowledge

Currently, the traditional approach to MCAD teaching is based mainly on the use of positive knowledge in the form of lecture-based courses employing tutorials and practical examples, along with definitions of guidelines and best practice. One advantage of such an approach is the immediate focus on the functional aspect of commands provided by a system and on specific modeling strategies usually applied to example cases resulting in success. Positive knowledge conveyed in this manner can be conceptualized in relation to engineering processes and tasks, which in turn are associated with goals and sub-goals within the context outlined earlier as follows. Each creation and editing of a surface-based CAD model can be abstracted as a set of geometric modeling tasks with individual elements denoted by *Ti* and defined as listed below.

- *T1*: analysis of the shape to be modeled
- *T2*: definition of a patch layout

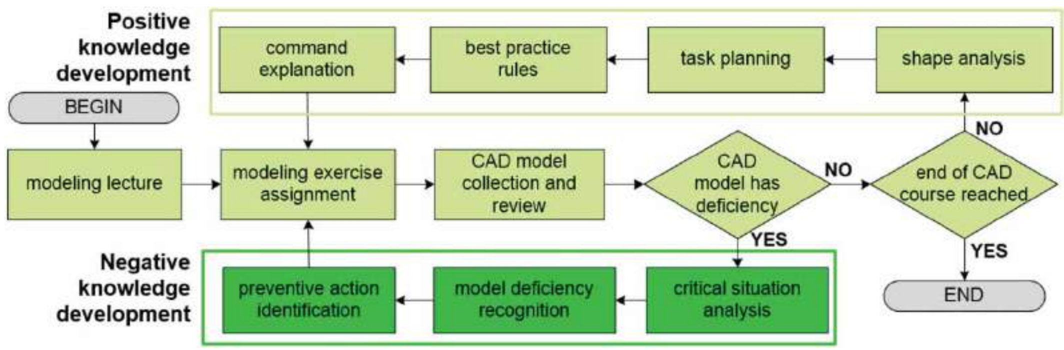


Figure 1. Overview of the newly developed CAD course structure.

- *T3*: modeling of curves required as input for patches
- *T4*: modeling of the main patches
- *T5*: modeling of trim, blending and round patches

Outcomes of surface modeling tasks that relate to commonly accepted modeling practice resulting in surface models with characteristics such as smooth surfaces, and fair curvature and adjacency properties among patches, can be compiled into a set of basic overall rules denoted by *Ri* and defined as shown below.

- *R1*: the patch layout should use the minimum number of patches possible
- *R2*: continuity and symmetry across center lines need to be ensured
- *R3*: each patch should, if possible, be a rectangular shape that needs to be defined by four boundary curves
- *R4*: each patch should be as simple as possible
- *R5*: the curves on the opposite sides of a patch should have the same number of control points and the same degree
- *R6*: curves should be single span (with the number of control points equal to the degree plus 1) and must have as few control points as possible

However, due to the context of hybrid geometric modeling, this set of basic rules needs to be extended. The surface model must now consist of a geometric structure, which is sufficient to guarantee that it can be converted consistently into a corresponding solid model. Therefore:

- *R7*: the modeling system tolerance needs to be defined carefully
- *R8*: rounds with very small radius must be avoided
- *R9*: curves and patches must not be smaller than the modeling system tolerance value

At this point, perhaps it needs to be made explicit that *T1* to *T5* and *R1* to *R6* as described are tasks and rules, which are paradigmatic of the domain of surface modeling. We should recall that the individual process steps within hybrid geometric modeling relate to an actual engineering application domain, namely modeling the skin, conversion into a solid, separation of the solid into components, and shelling of and adding features to the components. The above definitions relate to the first step of the process, and that step must be accomplished in a proper manner in order to create a CAD model that can be used in subsequent steps.

Based on the structural characteristics of model elements relating to topology and geometry and denoted by *Sn*, the set of basic modeling rules as outlined above can be grouped as shown below.

- *S1*: overall model structure (*R1,R7*)
- *S2*: model entity adjacency (*R2,R8*)
- *S3*: model entity intrinsic characteristics (*R3,R4, R5,R6,R9*)

A grouping for *S'n* similar in nature, related to tasks and their projected outcomes, can be made as shown below.

- *S'1*: overall model structure (*T1,T2*)
- *S'2*: model entity adjacency (*T5*)
- *S'3*: model entity intrinsic characteristics (*T3,T4*)

Note that the grouping of rules and tasks as shown does not represent a fixed relationship in a mathematical sense for every modeling situation possible, but is only a heuristic representative of the most common examples usually encountered.

However, this "positive" what-to-do approach lacks the knowledge and know-how required when facing unexpected or unusual situations. Those are not included in the smooth modeling path that is usually shown during modeling exercises, but they are particularly critical from the point of view of defining usable hybrid models. In order to help novices develop awareness of issues deriving from shortcomings and mistakes related to sub-optimal goal settings and inappropriate modeling strategies, and gain know-how on avoiding getting stuck in undesirable situations during problem-solving, we propose to integrate the traditional "positive" teaching approach with a newly developed approach, which is based on the use of negative knowledge, that is, knowing what not to do in a certain situation, and what assumptions are wrong with regard to a certain problem, and learning from errors.

3.2.2. Concept of negative knowledge

The basic design approach for negative knowledge as employed within the framework was to aim for more similarity, which means reducing variety. This objective was achieved by formulating negative knowledge as an element of strategic knowledge constraining actions within critical situations that would otherwise lead to errors and mistakes. In other words, the objective was to restrict actions that induce situations best avoided. In the context of hybrid geometric modeling within MCAD education and negative expertise, as discussed earlier in this paper, this translates into the goal of supporting the development of know-how and competency aimed at creating CAD models containing fewer undesirable structural elements. This can be achieved by systematically reducing model shortcomings introduced by errors and mistakes usually committed by novices, but never by domain experts.

Negative knowledge can be conceptualized, from a theoretical point of view, through relationships with desirable situations, which in turn can be indicated by what is considered a good model (configuration) within a given context employing normative knowledge (see discussions in the next sub-section). Here, desirable situations represent a reduced set of all possible situations (desirable/undesirable). In this scenario the nature of similarity of desirable situations is determined by reducing variety (cf. [15]), which in turn is realized by avoiding undesirable situations by means of restricting actions that have a high tendency (according to what we know and believe to be true) to lead to them. Hence, negative knowledge in terms of knowing what not to do in a certain situation can be conceptualized as a form of action constraint. It limits the variety of situations, and consequently their number, by preventing actions that might result in constellations (model configurations) considered not good, i.e. situations deemed undesirable. This concept now features both a quantitative and a qualitative method of determining similarity as an overall defining structural property of situations, which are considered desirable.

A situation can be abstracted as a set of relations associated with particular sets of model configurations, action constraints, anticipated failures, and individual goals and sub-goals. This concept of a situation is, in turn, defined by the model and the context (cf. [15]). For instance, a concrete situation is determined by the actual model configuration in a given context under a specific goal, and assumptions of possible failures if action constraints that are known for this situation are ignored. Properties that define the quality of the configuration, i.e. whether the model is good or not, are related to the normative knowledge of an application domain. Of particular interest are significant model configurations that relate to a context that is significant in respect to action constraints, which in turn are associated with individual actions. These significant model configurations can be related through a mapping to concrete constraints limiting the actions possible in a particular situation. Such constraints provide a concept that takes into account the portion of negative knowledge, mostly tacit in nature, which relates to action constraints spanning various different types of situations. Note that within the work presented in this paper, however, situations are related only to hybrid geometric modeling. Within the framework, several different types of situations are considered. They are classified according to whether they are related to the course of modeling actions, the model configuration, or the goal structure. The last category is organized as a hierarchy of an overall goal and individual sub-goals.

3.3. Concept transformation of normative knowledge

To translate the approach and framework as outlined earlier into pedagogical practice within a given context, normative knowledge needs to be established of concepts that characterize the shortcomings of CAD models, considered within an application domain. This represents a modus operandi that is consistent in nature with negative knowledge. However, it is different from traditional approaches with positive knowledge, where the focus is on efforts to characterize what is considered good, such as good design practice and geometric models that are good enough to be used in various tasks within computer-aided product development. For that purpose, by taking into account hybrid geometric modeling, the concept of geometric model deficiency has been developed. This concept can be seen as one important element supporting definition and evaluation of what is to be avoided in respect to particular situations and contexts. Of course, in a different context, for example in computer-aided design for environmentally-conscious products or computer-aided architectural design, such a concept would have a completely different emphasis.

Geometric model deficiency is used as a qualitative measure to help express certain characteristics of situations during modeling. These characteristics usually lead to models being poorly structured and are thus best avoided. In other words, deficiency represents the loss of one or several characteristics of a geometric entity meaningful in a certain application domain. Not only are those characteristics an important defining property of individual entities at a common level or dimension, but they may also become a significant element as an input for the definition of a higher dimensional entity. Currently, geometric model deficiencies as used within the newly re-designed CAD course, are sub-divided into four groups and denoted by *Dn* as shown below.

- *D1*: self-intersecting patch
- *D2*: peak and bump
- *D3*: degenerated patch
- *D4*: open boundary

Those geometric deficiencies are structured in regard to single entities and compound entities, with the former relating to curves, patches, and solids, while the latter relate to polycurves and polysurfaces. A detailed description along with definitions of all deficiencies as introduced, can be found in Appendix A.

Since individual geometric deficiencies as introduced above are used within the framework application as an instrument for supporting learning of domain-specific knowledge and development of negative expertise, these concepts are integrated with further knowledge elements relating their definitions to domain-specific problems, critical situations regarding a specific goal, preventive measures, and references to corresponding elements of positive domain-specific knowledge. Related information on domain-specific problems is employed to convey knowledge on why a particular geometric deficiency should be avoided and which kind of anticipated negative effects such a geometric deficiency is most likely to have on both the CAD model itself and further modeling operations. Information relating to particular goals and situations is used to convey knowledge on how to relate a particular geometric deficiency to an individual goal and model configuration that are considered critical, and thus signify when to become aware of and subsequently to look out for the deficiency. Information on preventive measures is used to convey knowledge on which actions are best avoided and which actions are most appropriate either to continue or to backtrack in case of recovering (if possible) from a mistake which has been committed earlier. References to corresponding elements of positive domain-specific knowledge are used to support the development of understanding of and insight on how actual outcomes in the form of successful results, and also errors and mistakes, are related to basic modeling operations and recommended rules of best practice.

Correspondence relationships between modeling rules, as elements of the positive knowledge domain, and geometric model deficiencies, as elements of the negative knowledge domain, in respect to individual modeling tasks, can be graphically represented as a lattice-based mapping onto the surfaces of a cube stretching in three directions as shown in Figure 2. The coloring of individual cells in the lattice structure is aimed at supporting an efficient and effective way of visually encoding the type of relationship represented in respect to the grouping *S* and *S'* respectively. Color encoding is based on the standard additive RGB color model where the combination of two of the three standard additive primary colors red, green, and blue in equal proportions produces an additive secondary color, namely cyan, magenta, or yellow. The color of individual cells is determined by the additive RGB color model related to the pre-defined colors of cells (see again the description of *S* and *S'* in the previous section) in the first column of both the task space and the rule space of the lattice structure.

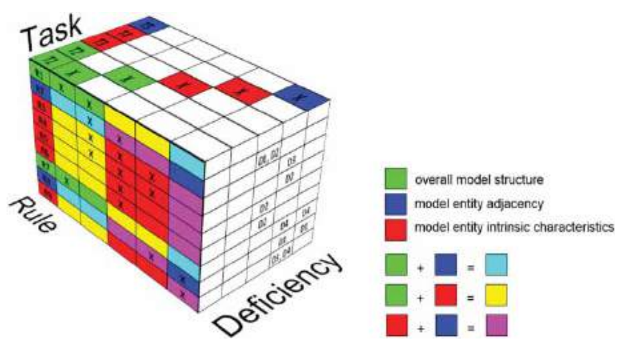


Figure 2. Overview of relationships between modeling tasks, modeling rules, and geometric model deficiencies.

Essential components for the definition of a good modeling strategy are expertise to correctly evaluate a given situation and the capacity to reason ahead. The three-dimensional graphical representation of the relationships between modeling tasks, modeling rules, and model deficiencies as discussed earlier, allows for a visual and more transparent way of explaining the complex correlations between actions performed during a certain task and the negative impact their outcome may have on tasks later in the process, even if those actions did not seem to violate any best practice rules at the time they were executed. For example, taking a look at task *T2* (definition of a patch layout), one can immediately recognize that rules from *R1* to *R4* are related to this task (indicated by the X mark in the second column of the task/rule space as shown in Figure 2. However, while rule *R1* explicitly refers to the patch layout (which is coincidental to the goal of task *T2*), rules *R2*, *R3*, and *R4* have only an implicit reference to the patch layout, as they actually refer to geometric properties required for either adjacent patches (*R2*) or single patches (*R3* and *R4*). The space visually representing geometric model deficiency in the visual representation in Figure 2 shows that no deficiency will explicitly be inflicted on the geometric model before task *T3*. Additionally it is shown that while performing task *T2* one should be aware of the rules that are actually related to entities (curves and patches) that have not yet been modeled. These circumstances are due to rules *R3* and *R4,* which are both related to task *T2* and the fact that violating these rules will introduce deficiencies *D2* and *D4* later on during task *T4*. This brief example reflects just one tiny portion of the know-how and capacity experienced CAD engineers possess and represents in nature the kind of competency that novices need to develop.

4. Implementation and example

4.1. Overview

A structural overview on how concepts of negative knowledge and geometric model deficiency, as introduced and developed, are used to implement the approach and integrate it into the current course work for CAD education is shown in Figure 3. Besides traditional lectures and tutorials, various modeling exercises are provided, and these are individually designed for different learning aspects. Results of the exercises are collected and assessed, to identify shortcomings and errors, which usually remain hidden from students due to their limited domain knowledge and expertise. Results are then used for feedback and reflective discussions on critical situations overlooked and errors committed. As the approach is scalable to adjust to the student body profile, which varies in each semester, individual knowledge and skill development cycles, as outlined in Figure 1, can be adjusted. Currently, individual cycles are designed for the duration of one week in regard to a course unit, and then repeated five times

4.2. Administration and concrete measures

To implement the approach and integrate it into the current MCAD course, besides traditional lectures and tutorials, various modeling exercises are provided. To set up an affordable and functionally adequate modeling

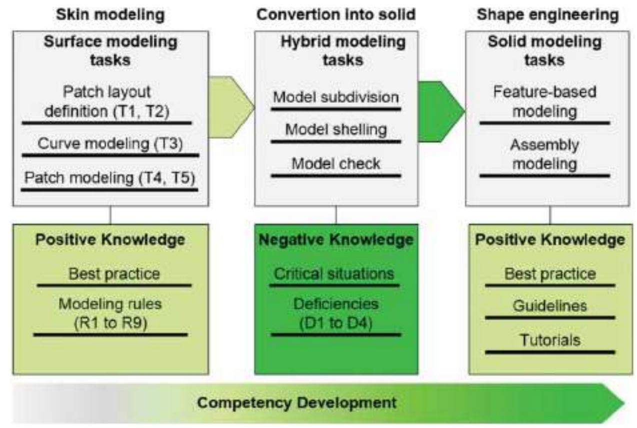


Figure 3. Overview of teaching methods and hybrid modeling skills related to competency development.

environment for the CAD course, two commercially available CAD systems in the mid range, namely *Solid Edge* from Siemens AG and *Rhinoceros 3D* from Robert McNeel & Associates, are deployed. This modeling environment structure serves two main purposes. First, it takes into account aspects related to both NURBS-based surface modeling and the exchange of CAD models between different system platforms, as is commonly required in practice. Second, it provides a surface modeling tool that allows for a relaxed approach to modeling since the geometric model quality is controlled entirely by the user. Hence, any kind of surface can be generated, including self-intersecting surfaces and surfaces of poor geometric quality. Note that this system characteristic is one key feature which, from a tool and modeling environment point of view, explicitly supports the implementation of learning by error and development of negative expertise as discussed elsewhere in this paper.

In order to manage the interaction between faculty and students, including the administration of procedures and deadlines for the distribution and collection of exercise material, a web site for the course has been developed within the e-learning platform of the institution's engineering faculty using *Moodle*, an open source learning management system (LMS). Note that the LMS is not used to create any domain subject related contents (see also discussions in [19, 20]). This LMS is also employed to implement and administer a computer-aided questionnaire, which is sub-divided into three parts and associated with the overall course structure regarding both chronological order and domain subject contents. Online participation in the questionnaire by students is both anonymous and voluntary. Information obtained in such a manner is aimed at supporting further insight into and understanding of the workings and implications of the newly-developed approach. It also supports current efforts to improve the framework developed and to provide constructive input for the design of next generation modeling examples and laboratory exercises, which are to be included in the course for the next academic year.

4.3. Example compilation

In the following it will be demonstrated how to translate and implement concepts and framework elements of the newly developed approach by presenting and discussing a selection of compiled examples that relate to the modeling of an actual product from the industrial engineering context, namely the housing of a telephone handset (cf. Figure 4), which serves as one exercise example among several that are employed in the currently modified CAD course curriculum.

Figure 4. Details of part and product geometry of the exterior of the modeling example.

Competency development within the new CAD course curriculum requires students to master modeling activities across three different modeling domains, namely surface modeling, hybrid modeling and solid modeling, but also to ensure that model interoperability across these modeling domains remains valid. Here individual modeling tasks associated with the different geometric modeling domains are as follows.

- surface modeling: modeling of the skin of the object
- hybrid modeling: converting the model into a solid; separating the solid into single components; shelling of the components
- solid modeling: completing the model by adding some functional features; setting up the assembly model

A traditional approach to such a kind of CAD exercise is usually based on positive knowledge only. In other words, the focus is on demonstrating and discussing the right steps that will safely lead from beginning to end. However, this approach will miss the chance to show the students the different types of drawbacks, shortcomings, and errors that may occur if something goes wrong. The approach presented in this paper is based on the integrated use of positive knowledge and negative knowledge, each taking turns in the role of dominant knowledge form as seems most appropriate. Positive knowledge, conveyed in the form of sequences of main tasks and control rules, is aimed at providing to novices a foundation to develop basic domain knowledge and skills at a level deemed appropriate for beginners. Negative knowledge, conveyed in the form of information and know-how on the identification of critical situations and related geometric model deficiencies, is aimed at the development of awareness and understanding regarding negative implications and subsequent impacts an inferior or inappropriate surface modeling strategy may have on the overall modeling process. This scheme also contributes to learning from errors and the development of negative expertise, which in turn and in like manner supports competency development. In what follows, a summarized overview consisting of three parts is presented on how positive knowledge and negative knowledge are communicated and used within the new CAD course. Note that, due to limits regarding the length of the manuscript, presentation and discussion of selected examples will be confined to the surface and hybrid modeling phases, which best represent typical cases of reference for the material used.

Part I. The role of positive knowledge

As recommended by best practice, before starting on the modeling process, an analysis of the shape needs to be performed (cf. task *T1*), in order to identify the main characteristics that will impact both the definition of the modeling strategy and how the modeling process is conducted. In the case of the modeling example, the overall shape of the handset exterior is characterized by its lateral and top silhouette curves together with some section curves. Additionally, the region where the receiver will be placed can be identified as a revolution surface that is a tangent to the rest of the shape, which needs to be blended with it. Next, task *T2* requires the definition of the patch layout. Within the given scenario it is quite natural to define a patch layout that is based on the silhouette and cross-section curves mentioned above. At this stage, students can verify that the proposed solution seems to meet the requirements imposed by rule *R1* as the layout that is based on the shape analysis looks simple and straightforward (see Figure 5). Of course, a more experienced user will immediately recognize that the proposed solution may lead to a critical situation, but this is not the case for a novice. Referring to the correspondence relationships introduced in Section 3.3, it becomes evident that in order to understand the nature of the impact the proposed layout will have on the overall modeling process, at the same time novices ought to be aware of the deficiency that will most likely be introduced during task *T4*, due to violating rule *R3*.

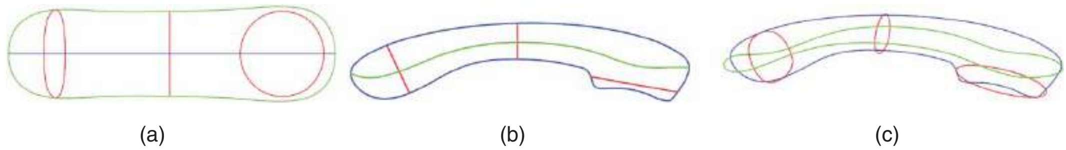


Figure 5. Silhouette curves and surface formation. From left to right: (a) bottom view, (b) side view, (c) isometric view.

After completing the previously mentioned preliminary tasks, students will face task *T3*, which is aimed at defining the curves that will then be used as input for the patch modeling commands, taking into account rules associated with curve modeling, such as rule *R6*, which recommends that we limit the number of control points. Notice that at this stage the modeling rules for curves do not provide any hint which will alert students to the previously incompetently defined patch layout. Once the curves have been modeled, task *T4* can be started, which is aimed at the modeling of the main patches. An appropriate and straightforward method seems to be to use the previously defined curves for modeling the central section of the handset. However, when we reach the region of the handset geometry where the microphone is to be located, problems will be encountered as follows. According to rule *R3* individual patches within a designed patch layout should tend to a rectangular shape. However, this requirement cannot be met with the curves that have been modeled in regard to the previously defined patch layout, as can be visually verified by taking a look at the current local model structure depicted in Figure 6.

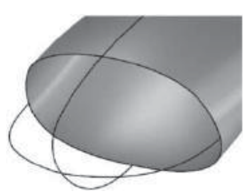


Figure 6. Surface formation with a patch layout containing non-rectangular patches.

To avoid defining a new patch layout from scratch, most probably novices will face this problematic situation by trying to model these non-rectangular patches and then determine, according to the outcome, if the result can be considered acceptable or not. Note that this approach is feasible only in the case of using a modeling system which adopts a *relaxed* approach, that is, which allows the user to implement patches based on any kind of layout design. Otherwise, systems will prevent the implementation of such degenerated patches. This reflects on the decision outlined elsewhere in this paper to employ the *Rhinoceros* modeling system as a geometric surface modeling tool that affords a relaxed approach. Depending on the modeling command and the sequence of curves eventually used as input, results with an appropriate adjacency and continuity of patches can be obtained. However, in all cases, a degenerated boundary in the form of a collapsed point is created. This situation is illustrated in Figure 7 by patch isocurves converging and eventually collapsing in one point at different locations.

Part II: The role of negative knowledge

Due to a lack of (negative) expertise and competency, awareness and correct interpretation, as well as understanding, of the previously described quite precarious situation are absent. Hence, the severity of this critical modeling situation and its implications for further modeling tasks remain overlooked by novices. They will now continue in the modeling process by trying to import the previously created surface model into a solid modeling tool. Depending on the severity of the deficiencies introduced into the model during the previously mentioned modeling operations, students may face several different

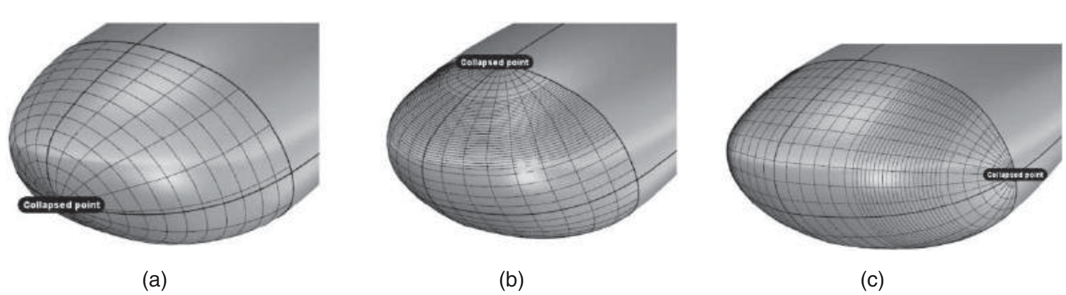
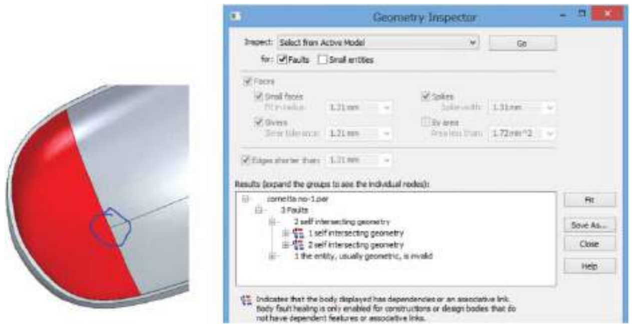


Figure 7. Surface geometry and patch layouts supplemented with isocurves. From left to right: (a) patch layout with collapsed point on the tip, (b) patch layout with collapsed points on top and bottom, (c) patch layout with collapsed points on the left side and right side.

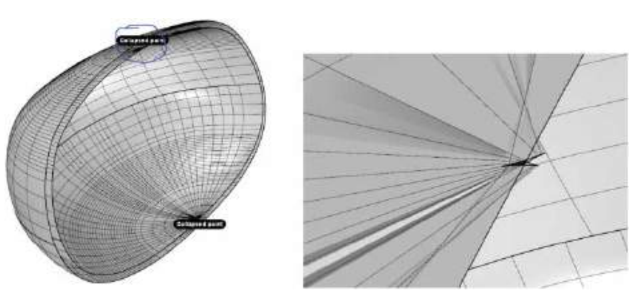
kinds of issues. These will vary from an inability to convert the surface model into a solid model, to a failure during the application of the shelling command, or the identification of self-intersecting surfaces during a model integrity check (as shown in Figure 8).



(a) (b)

Figure 8. Deficiency analysis within the solid modeling tool. From left to right: (a) region subject to geometric analysis with location of deficiency indicated on the upper part, (b) system generated message of analysis results.

All issues just outlined can actually be related to the fact that the offset of a patch with a collapsed point will generate a self-intersection in the vicinity of the collapsed point. For supporting reflective analysis of the modeling mistake committed and for educational purposes, the model deficiency in the form of self-intersecting surfaces is made explicitly visible by going back to the surface modeling tool and by making the offset of the critical patch as shown in Figure 9.



(a) (b)

Figure 9. Deficiency analysis within the surface modeling tool. From left to right: (a) offset of originally created surfaces in the region of deficiency subject to geometric analysis, (b) section of an enlarged view of this geometry in the region close to the upper collapsed point.

At this stage, an explicit relationship has been established between the modeling issue, the model deficiency, and the corresponding modeling rule, in which, according to ideal cases, the modeling rule should have prevented the occurrence of such a deficiency. What has been demonstrated so far will not only help to explain why the model is unusable for subsequent engineering tasks, but also contribute to the development of student competency, due to an improved awareness of the underlying reasons behind the definition of the modeling rules.

Through such first-hand experience during laboratory exercises on quite difficult issues such as the relationship between modeling rules and geometric model deficiencies, support is provided for the development of awareness. This awareness involves understanding *when*, *why*, and *how* all the recommendations and rules of best practice need to be followed and interpreted.

Part III. Back to the role of positive knowledge and expertise

After one of the negative knowledge cycles has been completed within an exercise (see again Figure 1 and Figure 3), interim modeling results of the same modeling example can be used to show alternative solutions that can be put in place at different levels. For example, one feasible solution among several is to locally change the patch inflicted with the deficiency. This goal can be achieved, for example, by trimming the region around the collapsed point and closing the area with a new patch, as shown in Figure 10.

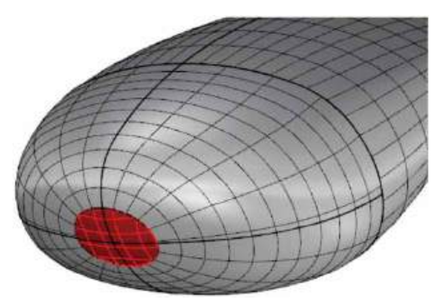


Figure 10. Local geometry and modified patch.

An alternative solution, although more drastic in nature, requires the complete re-design of the patch layout, starting all over again from scratch by employing an

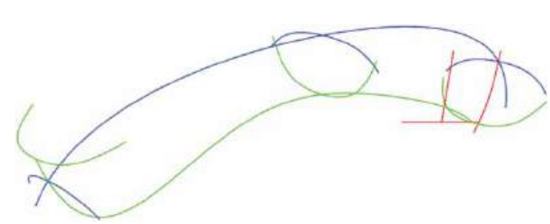


Figure 11. Surface formation under a different interpretation of the shape analysis.

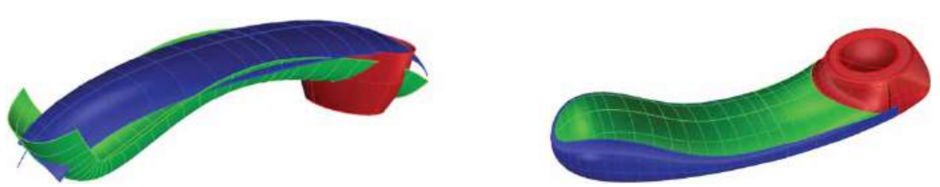


Figure 12. Patch layout under a different interpretation of the shape analysis.

entirely different interpretation of the shape analysis, as shown in Figure 11.

Perhaps, at a first glance, this new layout appears to be less intuitive and more complex. In comparison to the previous layout it also seems to meet rule *R1* to a lesser degree. However, benefiting from what was previously learned in the exercise, students are now in a better position to recognize that this layout is actually much more appropriate and less prone to errors than the previous layout. For the implementation of this solution, some aspects of the modeling procedure will be more difficult than in the previous case, due to the need to deal with trim and blend in order to create the final result, as shown in Figure 12. However, this fact should not cause us to misevaluate this solution. It is simply a hurdle that is prevalent in such cases for novices and which needs to be overcome by furthering competency development.

Another trial of importing the surface model into a solid modeler can then be carried out, and that will hopefully allow the students to continue the modeling tasks until the final result can be reached, as depicted in Figure 13.

Figure 13. Details of part and product geometry of the modeling example. From left to right: (a) solid model of the exterior, (b) upper and lower housing shell.

5. Conclusions and future work

Within the contributions reported in this paper, the framework, central concepts, and issues of current implementation of a novel approach have been outlined and discussed. This approach is aimed at supporting competency development within hybrid geometric modeling for the wider application context of product design. It is based on the integration of traditional teaching methods with an educational approach based on negative knowledge. Within this new approach, development of situational awareness is targeted in respect to critical modeling situations and potential failures, which may occur if inappropriate strategies and/or actions are chosen.

A compiled selection of examples was given to illustrate central concepts of the framework and how they relate to and interact with domain knowledge application and the development of skills and expertise. As became evident during both theoretical analysis and first empirical work, current efforts to integrate traditional teaching methods with an educational approach based on negative knowledge are indeed capable of supporting capacity development in the form of explicit knowledge about what is not a geometric model sufficiently structured for subsequent engineering tasks related to product development processes, what not to do during strategy formation and action translation, and a domain-specific situation awareness related to a knowledge-based error anticipation.

First promising results were reflected in, among other things, students showing an overall better understanding on issues related to the usability of CAD models and an increased capability to recognize critical modeling situations and thus prevent mistakes typically made by novices. Also, an overall qualitative increase in planning and performance was observed during problem-solving course work and laboratory exercises, and this was achieved, in part, by identifying and subsequently avoiding sub-optimal and inappropriate modeling strategies. To further understanding and insight on both a theoretical and a practical basis regarding the design and implementation of the novel educational approach as outlined, current efforts in the collection of empirical data will be continued. Presentation of detailed results including statistics of the examination and analysis of empirical data collected until the end of the current MCAD course, is planned for a forthcoming publication.

ORCID

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

References

[1] Bauer, J.; Mulder, R. H.: Modelling Learning from Errors in Daily Work, Learning in Health and Social Care, 6(3), 2007, 121–133. http://dx.doi.org/10.1111/j.1473-6861.2007.00150.x

[2] Case, J. M.: Problematizing Curriculum: Contemporary Debate in Engineering Education, in: M. Young and J. Muller (eds.) Knowledge, Expertise and the Professions, Routledge, Oxon, UK, 2014, 143–156.

[3] Dankwort, C. W.; Weidlich, R.; Guenther, B.; Blaurock, J. E.: Engineers' CAx Education-It's Not Only CAD, Computer-Aided Design, 36(14), 2004, 1439–1450. http://dx.doi.org/10.1016/j.cad.2004.02.011

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

[5] Gartmeier, M.; Schüttelkopf, E. M.: Tracing Outcomes of Learning from Errors on the Level of Knowledge, in: Bauer, J. and Harteis, C. (eds.), Human Fallibility: The Ambiguity of Errors for Work and Learning, Springer, Dordrecht, The Netherlands, 2012, 33–51. http://dx.doi.org/10.1007/978-90-481-3941-5_3

[6] Harteis, C.; Bauer, J.; Gruber, H.: The Culture of Learning from Mistakes: How Employees Handle Mistakes in Everyday Work, International Journal of Education Research, 47(4), 2008, 223–231. http://dx.doi.org/10.1016/j.ijer.2008.07.003

[7] Heid, H.: Erfordernis und Problematik einer Unterscheidung zwischen Verhalten und Verhaltensdisposition, in: Beck, K.; Müller, W.; Deißinger, T. and Zimmermann, M. (eds.), Berufserziehung im Umbruch, Deutscher Studienverlag, Weinheim, Germany, 1996, 79–85.

[8] Hunt, E.: Expertise, Talent, and Social Encouragement, in: K. A. Ericsson, N. Charness, P. J. Feltovich and R. R. Hoffman (eds.) The Cambridge Handbook of Expertise and Expert Performance, Cambridge University Press, New York, NY, USA, 2006, 31–38. http://dx.doi.org/10.1017/CBO9780511816796.003

[9] Kolodner, J.: Towards an Understanding of the Role of Experience in the Evolution from Novice to Expert, International Journal of Man-Machine Studies, 19(5), 1983, 497–518. http://dx.doi.org/10.1016/S0020-7373(83)80068-6

[10] Minsky, M.: Negative Expertise, International Journal of Expert Systems, 7(1), 1994, 13–19.

[11] Minsky, M.: The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Human Mind, Simon & Schuster, New York, NY, USA, 2006.

[12] Muller, J.: Forms of Knowledge and Curriculum Coherence, Journal of Education and Work, 22(3), 2009, 205–226. http://dx.doi.org/10.1080/13639080902957905

[13] Oser, F.; Spychiger, M.: Lernen ist schmerzhaft: Zur Theorie des negativen Wissens und zur Praxis der Fehlerkultur, Beltz, Weinheim, Germany, 2005.

[14] Oser, F. K.; Näpflin, C.; Hofer, C.; Aerni, P.: Towards a Theory of Negative Knowledge (NK): Almost-Mistakes as Drivers of Episodic Memory Applications, in: Bauer, J. and Harteis, C. (eds.), Human Fallibility: The Ambiguity of Errors for Work and Learning, Springer, Dordrecht, The Netherlands, 2012, 53–70. http://dx.doi.org/10.1007/978-90-481-3941-5_4

[15] Otto, H. E.; Mandorli, F.: Negative Knowledge and Eco-Design, in: Proceedings of the International Conference on EcoBalance, November 20–23, Yokohama, Japan, Paper-No. P-133, 2012.

[16] Parviainen, J.; Eriksson, M.: Negative Knowledge, Expertise and Organisations, International Journal of Management Concepts and Philosophy, 2(2), 2006, 140–153. http://dx.doi.org/10.1504/IJMCP.2006.010265

[17] Posner, M.: What Is It to be an Expert?, in: Chi, M. T. H.; Glaser, R. and Farr, M. J. (eds.) The Nature of Expertise, Erlbaum, Hillsdale, NJ, USA, 1988, xxix – xxxvi.

[18] Wang, G. G.: Definition and Review of Virtual Prototyping, Journal of Computing and Information Science in Engineering, 2(3), 2002, 232–236. http://dx.doi.org/10.1115//1.1526508

[19] Walker, R.; Voce, J.; Nicholls, J.; Swift, E.; Ahmed, J.; Horrigan, S.; Vincent, P.: 2014 Survey of Technology Enhanced Learning for Higher Education in the UK, Universities and Colleges Information Systems Association (UCISA), University of Oxford, Oxford, UK, 2014.

[20] Watson, W. R.; Watson, S. L.: An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become?, TechTrends, 51(2), 2007, 28–34. http://dx.doi.org/10.1007/s11528-007-0023-y

[21] Winch, C.: Dimensions of Expertise: A Conceptual Explanation of Vocational Knowledge, Continuum, London, UK, 2010.

[22] Winch, C.: Know-How and Knowledge in the Professional Curriculum, in: Young, M. and Muller, J. (eds.), Knowledge, Expertise and the Professions, Routledge, New York, NY, USA, 2014, 47–60.

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

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

APPENDIX A

All the descriptions of the geometric model deficiencies denoted by *D1, D2, D3*, and *D4* as given in this appendix are structured in regard to single entities and compound entities, with the former relating to curves, patches, and solids, while the latter relate to polycurves and polysurfaces.

D1	Self-intersecting patch
Definition	A surface folded to the point to intersect itself (ref. figure a).
Type	Entity deficiency
Related problems	A self-intersecting surface cannot be the boundary of a solid.
Critical situations	**To make a sweep**: a sweep along a curve with minimum radius r and cross-sections with dimensions $\geq$ r (ref. figure b). **To make an offset**: an offset of distance $\geq$ r of surfaces having curvature radius $<$ r (ref. figure c).
Preventive action	Carefully check curvature values of curves and surfaces before applying sweep and offset commands (or commands based on sweep and offset, like shelling and thickness).
Ref. to positive knowledge	R8

D2	Peak and bump
Definition	A peak is a wrinkling of small dimensions; a bump is a relief due to a local and small fold (ref. figure d).
Type	Entity deficiency
Related problems	These small imperfections can compromise the model usability due to their small curvature radius, which can cause self-intersections during the offset operations that are part of the shelling and thickness commands.
Critical situations	**To select input curves:** input curves with an unnecessarily high number of control points can easily generate peaks or bumps. This is due to the local deformation imposed by the eventual constraint of adjacent continuity (ref. figure e). Curves across center lines must have an appropriate continuity.
Preventive action	Check and eventually reconstruct the curves that are derived from other entities, like isocurves, mid curves, and projected curves. Do not use polycurves. Instead use single curves as input for patch modeling commands whenever possible.
Ref. to positive knowledge	R2, R4, R5, R6

D3	Degenerated patch
Definition	A patch with one boundary or two boundaries collapsed into a point (ref. figures f and g).
Type	Entity deficiency
Related problems	Patches with collapsed boundary have the normal vector undetermined in the collapsing point. The indetermination of the normal vector makes the offset operation critical. The higher the distance to be offset, the more critical the defect (ref. figure h).
Critical situations	**To model a patch:** a poor patch layout will most likely lead to the application of commands and input curves that will not result in a four-border patch.
Preventive action	Carefully define the patch layout and plan a modeling strategy that will prevent the need for creating patches with fewer than 4 boundary curves (ref. figures i and j). In any case, do not use commands and input curves that will create patches with a boundary degenerating into a point.
Ref. to positive knowledge	R3

D4	Open boundary
Definition	A surface made of joined adjacent patches with adjacent boundaries that are not properly "sewed" (ref. figures k and l).
Type	Entity relationship deficiency
Related problems	A surface with open boundary cannot be the boundary of a solid model (i.e. it cannot be converted into a sheet solid or bulk solid).
Critical situations	**To model sewed patches:** in order for patches to be sewed, the distance between adjacent boundaries of the patches must be within the modeling system tolerance value. The selection of the type of command and related settings (command inputs and options) is critical to obtain properly sewed patches.
Preventive action	Check the modeling system tolerance value. Do not use commands and settings that will produce an adjacency distance that is greater than the modeling system tolerance value. Do not use commands that necessitate sewing unless the boundary distance is within the system tolerance.
Ref. to positive knowledge	R7, R9

Reference Figures: