COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 1, 108–123
http://dx.doi.org/10.1080/16864360.2015.1059201

Explicit 3D functional dimensioning to support design intent representation and robust model alteration

Ferruccio Mandorli[a], Harald E. Otto[a] and Roberto Raffaeli[b]

[a]Polytechnic University of Marche ; [b]eCampus Online University

ABSTRACT
Design intent representation is a well-known issue in the MCAD domain, and is related to the readability, alterability, and usability of CAD models. The recent widespread introduction of functionality and commands in modern CAD systems, aimed at facilitating explicit modeling, introduces not only a new modeling paradigm supplementary to the feature-based approach, but also a new perspective on how the design intent can be captured and represented. Taking into account the traditional method of communicating design intent with functional dimensioning in mechanical drawings, in this paper a novel approach is presented, aimed at translating this traditional design intent representation from 2D into 3D. Objectives are directed towards the specification and implementation of dimensioning correspondence mapping and the identification as well as examination of shortcomings in current systems. This should help direct future improvements aimed at supporting 3D dimensioning within 3D explicit modeling systems.

KEYWORDS
explicit modeling; 3D driving dimensions; 2D/3D dimensioning mapping

1. Introduction

In recent years the rapidly increasing demand for and use of 3D geometric models for analysis, simulation, assessment, documentation, etc., has further intensified. Nowadays, virtual prototypes play a key role in supporting design and decision-making, while considerably reducing product development time and limiting the demand for and costs of building physical prototypes. In other words, virtual prototypes are fully integrated in the iterative decision-making activities related to the design and product development process.

Within this context, the efficient and robust alteration of 3D CAD models is a well-known issue. Two aspects are central to virtual model alteration. First, models need be easy to alter by means of using an intuitive user interaction. Second, the original design intent has to be preserved throughout the process of altering the model. In the past decade, efforts to solve this issue have been devoted to the development of feature-based modeling systems. This trend is also reflected in the modeling functionality and data structures provided by the majority of current commercially available MCAD systems.

The concept of feature was introduced in the early 1980 s as a means to cluster and represent engineering information about a product and relate such information to the geometric representation of the product itself. The aim of the feature-based methodology was to support a more efficient creation, modification and use of the product model, throughout its complete lifecycle, by adding "semantics" to the simple geometric representation of the product, and then to make explicit that part of the information that is implicitly related to the geometric properties of a component. In particular, the role of features-in-design was to relate functional meaning to the different shape elements of a product component, most specifically in the embodiment phase of design. An overview of the development of so-called *feature technology*, from the early 1980s to the mid 1990s, including its historical development from the first definitions of feature taxonomy to the problems related to feature recognition, feature mapping and design-by-feature, can be found in [8,21,26].

Nowadays, the concept of feature plays a major role in the so-called *feature-based modeling systems*, where features are actually used to record the modeling procedure that leads to the final shape of a component. Some features, such as holes, ribs, loops, and keyholes, are able to precisely convey functional meaning, while others, such as cutouts and extrusions, have a more generic and imprecise relation to function (especially if we recall that features always need to be related to the specific design context). A feature-based model should be easier

CONTACT Ferruccio Mandorli f.mandorli@univpm.it

to interpret and a feature-based modeling system should better support model creation and alteration by making explicit the relationship between geometry and functions, i.e. the design intent. Software procedures have been developed to drive feature-based modeling, while model templates, part families, assembly variants and spreadsheets linked to the model parameters have been successfully used in the development of different types of knowledge-based engineering and design automation application, and these solutions have often been applied to deal with the problems of mass customization [5,10].

In order to address the problem of feature-based model alteration, two main approaches have emerged in recent years: on the one hand, there is the methodological approach, based on the definition of best practices, guide lines, rubrics and, more in general, modeling strategies, able to lead to "well designed" feature-based models; on the other hand, there is an extension of the modeling system functionality, aimed at providing a faster and easier local modification of the model shape, independent of its feature-based representation. An extension of the latter has led to the introduction of a new modeling approach, referred to as both *explicit modeling* and *direct modeling*. This modeling approach is based on the definition of 2D regions the user can interactively manipulate, to add volume to or remove volume from the model shape. Geometric constraints between model entities, as well as dimension constraints, are directly linked to the 3D model and they determine (drive) the way the model can be altered. When we consider both design intent representation and the way a CAD model can be altered, in terms of both dimensions and shape, some new opportunities unfold for the explicit modeling approach, and we can perceive possible benefits compared with feature-based modeling. If we compare the way a feature-based system and an explicit modeling system deal with the problem of robust model alteration, we may propose the following statements. A feature-based system allows for the capture of design intent by using modeling features. However, if the feature-based model is not well designed, the model alteration can become tricky because the feature tree (i.e. the model history) no longer reflects the design intent. On the other hand, an explicit modeling system makes it easier to alter the model shape, but it does not seem able to support the explicit representation of the design intent that should drive a robust model alteration.

Although there have been considerable research efforts (which are still ongoing) related to establishing a better definition of the strategic knowledge required to support a robust feature-based model definition and alteration, no work (as far as the authors are aware at the time of writing) has been done in the field of assessing how far explicit modeling systems are actually able to capture and preserve the design intent and support robust model alteration. Now the objective of the approach presented is to identify a methodological approach to explicit modeling which, by taking advantage of the new functionality provided by the explicit modeling systems, will be able to support design intent definition and robust model alteration. The proposed approach is based on the following assumptions. Firstly, there is a focus on the embodiment phase of design (which is actually the phase that most requires highly interactive modeling tasks). Secondly, by design intent we mean the way elementary function can be achieved by the definition of specific shapes and dimensions. Thirdly, in the physical world dimensions are always related to tolerances. Fourthly, there is an explicit relationship between elementary functions and the way dimensions are specified on a component.

The proposed approach aims at the development and implementation of a framework which is capable of introducing and establishing *explicit 3D functional dimensioning* as a three-dimensional counterpart to traditional 2D functional dimensioning as used in mechanical drawing. This will offer a structured and unambiguous method of design intent formation and representation based on functional dimensioning in the domain of 3D models and modern MCAD systems. Just as a designer is able to recognize the design intent by looking at a component represented on a 2D mechanical drawing, there is no reason to believe that that designer will not be able to recognize the design intent on a 3D model, assuming that the 3D model contains the same information as that contained in the 2D drawing. It seems evident that if the information about the component shape is equally represented on a 3D model and on a correct 2D drawing, then the 2D drawing will also contain a lot of additional information that needs to be mapped onto the 3D model, and the most important of this information is the functional dimensions.

Within the research as presented in this paper, individual objectives were formed in the following three directions. First, development of dimensioning correspondence mapping, which is central to the implementation of explicit 3D functional dimensioning. Second, theoretical and empirical work to determine how far an actual implementation can be carried out, using functionality common to most commercially available MCAD systems supporting 3D explicit modeling. Third, forming suggestions for the improvement of current MCAD system user interface structures and functionality to further support the application of 3D explicit functional dimensioning.

2. Design intent, functional dimensioning and model alteration

As of now, there is no a unique definition of the term *design intent* (see discussions in [2, 3, 4, 19, 20], and for an overview see also [7, 18, 22]). Because it is a vague and complex concept, most, if not all, technical standards related to design intent do not provide a definition either. When we take a look at the history of CAD technology, we find that the concept of design intent was introduced in this field with the aim of making clear that a design support system (CAD) should be able to represent and manage not only geometry but also some of the higher level information, which from an engineering point of view, is related to shape. Going back to the early 1990 s, Henderson was defining design intent as "the purpose or underlying rationale behind an object" ([11], p. 387). Note that Henderson distinguishes between function, which he describes as "the behavior of an object", defined as "an operation of energy, material, or signal", and design intent, about which he states, "The intent differs from functionality in that the intent justifies a design decision whereas the functionality just tells what the design does." In recent years, there has appeared to be an increasing tendency to relate the concept of design intent to the concept of function. For example, Whitney states that, "Every assembly has a purpose or function. Every assembly is a collection of connected parts that is designed to achieve that function. The function is the *what* and the design is the *how*. The term design intent refers to the *how*, and includes the spatial arrangements of the parts, their sizes and shapes, and their geometric relationships to each other." ([27], p. 315).

Zhang and Luo [29] point out that CAD illustrates design intent through its history, features, parameters, and constraints. The original reasons for having CAD models capable of representing the design intent were two-fold. On one hand, such models encouraged the development of systems that were capable of automatic or semi-automatic reasoning about the design objects at a higher level of abstraction than the pure geometry. This was, and still is, reflected in efforts to develop process-planning systems, manufacturing cost estimation systems, etc. In addition, there was a strong demand for making digital models easier to alter, while preserving the related functionality, due to recognized benefits related to the reuse of computerized models from previous projects in future projects.

Over the years, increased distribution and use of parametric feature-based CAD systems has in practice somewhat modified the design intent definition to the point that today there is a considerable overlap between the concept of design intent and the concept of model alterability (see [6, 9, 23, 28] and related discussions in [1, 24]). For example, Rynne and Gaughran [25] define design intent as " ... the term used to describe how the model should be created and how it should behave when it is changed. It is not just about the size and shape of features, but includes tolerances, consideration of manufacturing processes, relationship between features, dimensions, and the use of equations." Additionally, Rynne states that, "Design intent is built into the model according to how dimensions and relations are established. Changes to a model will yield a different result for each different design intent." (cf. URL: www3.ul.ie/~rynnet/designintent-solidworks.php). Also, explanations of design intent within handbooks for CAD systems, such as for the system Pro/Engineer, state that, " ... features can be related to one another in a number of different ways. These relationships govern how the model will behave when changed. Design intent is the careful control of these relationships so that they correctly govern the intended behavior of the model. With good design intent, models can be updated almost effortlessly." (cf. Pro/Engineer wikibook at URL: en.wikibooks.org/wiki/Pro_Engineer/Design_ Intent).

Within this context, as the direction of some elements of design intent definition is partly driven by the developments of modern CAD systems, the question of whether and how far feature-based CAD systems are actually capable of representing design intent and supporting robust model alteration seems to have reached a cross-roads. The result is a recently rising trend towards integrating explicit modeling with CAD systems. Since the paradigm of direct modeling is independent of the concept of modeling features, the newly arising question is likely to re-shape itself into whether and to what extent explicit modeling systems are capable of capturing and representing the design intent. In order to provide an answer to this question, one might start off by taking into account two basic issues as follows. First, we should consider the traditional way in which mechanical engineers document and convey the design of components, namely 2D mechanical drawings. In other words, by looking at the mechanical drawing of a component, the engineer is able to recognize how and by which means functions are achieved, due to the shape of the represented component and the specified dimensions and tolerances. Second, we should consider the core characteristics of the explicit modeling paradigm. In other words, the possibility of explicitly adding driving dimensions (that is dimensions that can be used to alter the model shape through modification of the dimension values) to the 3D shape without any constraints related to the modeling sequence, gives the user the opportunity of approaching

3D dimensioning by employing almost the same criteria and mental processes as are used for 2D dimensioning in mechanical drawings.

We need to gain insight on whether explicit modeling systems are capable of representing design intent at least to the same extent that 2D mechanical drawings can do so, and whether design engineers will be able to recognize the design intent on the 3D model, even in the absence of explicit features and a feature history tree. To discover this, one has to examine first whether the semantics that designers can express, by using the syntax provided by the standards for 2D dimensioning, can be mapped from the 2D drawings to the 3D models, eventually by means of using a different syntax. The need to consider both syntax and semantics arises because the mechanical drawing is actually a graphic language based on commonly accepted rules and criteria, most of which are now specified by international standards. Those standards, together with standards on geometric product specification, introduce ideas, principles, and definitions, which highlight characteristics of the distinctions among different types of dimensions. Functional dimensioning represents one out of several methods of determining the dimensions and tolerances to be added to mechanical drawings.

The tenet central to this method is to specify these unambiguously and to communicate the functional meaning of the geometric elements that compose the shape of a part or component. One of the central problems being addressed within the framework developed and presented in this paper is conceptual insight into, and correct identification of, the elements and entities used in 2D drawings to define dimensions, and their correct mapping into corresponding 3D elements capable of supporting definition of functional dimensioning in 3D (see also Figure 1). This is an issue that is not as trivial as it might appear at first glance, because some types of elements and entities are represented in a different way in 2D and 3D. For example, an axis might be represented explicitly in 2D, but not in 3D and vice-versa. And elements that are of a different type might be represented in the same way in one of the representation schema. For example, a boundary edge and a silhouette edge might be represented in the same way in a 2D drawing.

3. Approach and framework

Concepts, structures, and mappings, as defined within the framework and described in detail in this section, aim at providing a means to investigate and enable novel ways of placing functional dimensions of mechanical components directly as 3D driving dimensions within the MCAD systems that provide explicit modeling functionality. Such functional dimensions will usually be the functional 2D dimensions in mechanical drawings. Design and formulation of the framework and concepts are oriented on the set of international standards for Geometric Product Specification and Technical Drawings and best practices, which provide definitions, general principles, and strategies for 2D dimensioning and its consistent representation within technical documentation such as line drawings for mechanical engineering.

3.1. Basic idea and conceptual outline

When an engineer looks at the mechanical drawing of a component, that engineer is able to recognize the associated design intent, due to the combination of the shape of the represented component and the specified dimensions and tolerances. This fact aroused our interest when we were analyzing the different aspects that are related to dimensioning on a 2D drawing. Analysis of how 2D functional dimensioning is defined on a mechanical drawing has to start with the standards which define the exact details of dimensioning by providing concepts and rules for syntax and semantics.

The standard that defines the syntactic rules for dimensioning is ISO 129 - Technical Drawings - Indication of Dimensions and Tolerances [13]. This standard provides definitions and rules on how the graphical symbols must be used in order to place dimensions on a

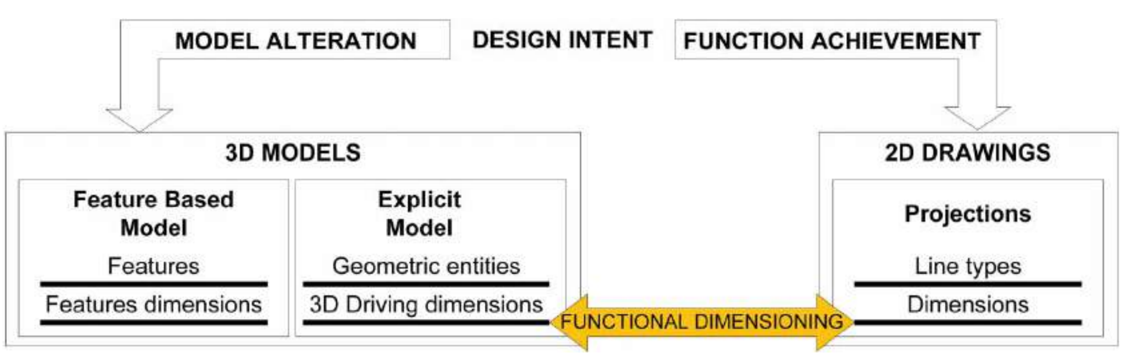


Figure 1. Design intent and model alteration and the domains of 2D/3D representation.

technical drawing. The standard that defines the semantic rules for dimensioning, in particular with reference to mechanical applications, is ISO 14405 - Geometric Product Specification (GPS) - Dimensional Tolerancing. This standard consists of two parts: Part 1: Linear Sizes, and Part 2: Dimensions other than Linear Sizes. Part 1 gives definitions and rules for the specification and interpretation of size dimensions. Size dimensions as defined by this standard, for example, can be the diameter of cylindrical elements and the distance between two opposite and symmetric parallel surfaces (cf. [15]). Part 2 provides definitions and rules for the specification and interpretation of the other types of dimensions (cf. [16]). The reason why the standard distinguishes between size dimensions and dimensions other than size, is that size dimensions can be directly related to assembly problems, so they usually require a more precise definition and interpretation of related tolerances. Note that from the syntactic point of view in respect to our inquiry on the subject, there is no difference between dimensions of size and dimensions other than size regarding their representation and placing in a mechanical drawing. The current classification of different types of dimensions, other than size dimensions, as provided in Part 2 of the ISO 14405 standard, is shown in Table 1.

Our goal is to develop novel ways of directly placing functional 2D dimensions representing design intent into mechanical drawings, as 3D driving dimensions within the MCAD systems that provide explicit modeling functionality, and to do this in a structured, consistent and standardized manner. To achieve this, it is a prerequisite that we gain insight into and an understanding of the nature and requirements of the different modeling situations that a 3D explicit modeling system has to manage in respect to the different types of dimensions that can be found in mechanical drawings. Table 1 provides a good starting point for a number of reasons. First and foremost, the dimensions are classified in a manner independent from their representation space and its dimension (2D or 3D). Second, the table includes all the possible types of dimensions that may occur in a mechanical drawing, including size dimensions, which can be classified as one feature / integral and two feature / integral-integral. Third, the classification provides references to integral and derived entities (cf. [17]), which in turn can be related to the explicit and implicit entities that are part of the 3D geometric model of the MCAD system.

3.2. Functional dimensioning and entity structure representation

Efforts to relate the different types of functional dimensions, as defined by the standard, to their corresponding 2D representation, while developing structures to define their corresponding 3D representation, have resulted in combinations of entity representation relationships being ordered and classified as shown in Table 2 and Table 3. In Table 2 all types of relationships as identified for linear dimensions are listed, while their counterparts for

Table 2. Classification of entity representation relationships for linear dimensions.

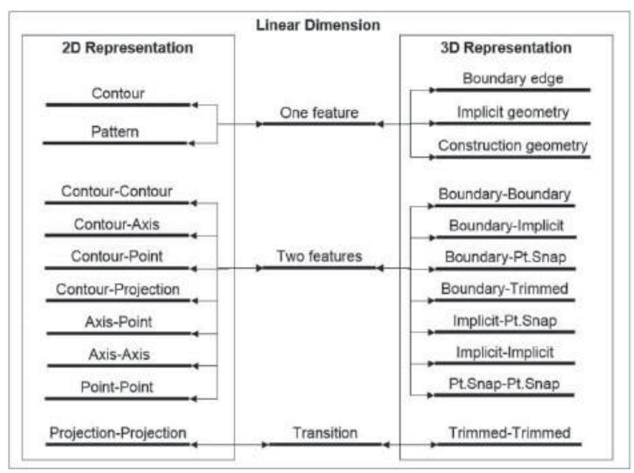


Table 1. Overview of concepts used to define different types of dimensions reflecting on the structure of the classification as reported in the GPS standards.

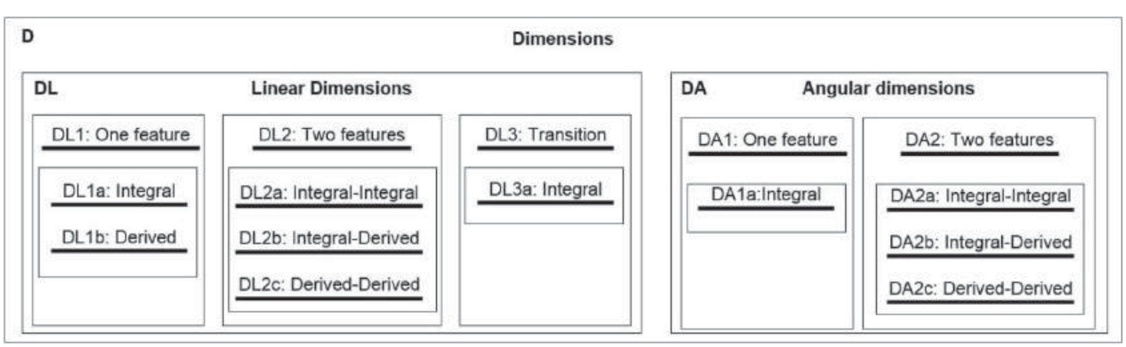
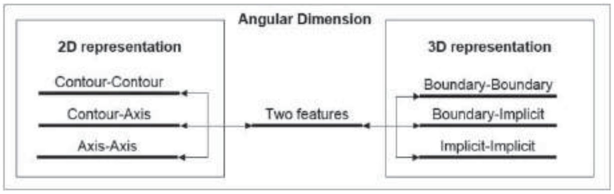


Table 3. Classification of entity representation relationships for angular dimensions.



angular dimensions are listed in Table 3. The classification considers all the different types of geometric entities required to define the various types of dimensions taking into account all relevant cases for both 2D and 3D. Note that both tables explicitly reflect the first hierarchical level as reported in the standard table (see again Table 1 and [16]).

The concrete requirements of valid combinations in respect to how a particular functional dimension is represented using corresponding entities from both 2D and 3D can be determined from the classification of entity representation relationships as given in the tables. For instance, in order to represent in 2D the linear dimension of a single integral feature, a single contour element is required, while in 3D a boundary edge or an implicit geometry, such as a silhouette edge, is required. Note that the classification as listed in the two tables is all-embracing as it takes into account all possible combinations and their respective entity relationships. However, in industry and engineering practice, it may be that not all of the cases listed actually occur, and this is due to the current state of the art of the explicit modeling which supports the MCAD systems.

For example, a single integral feature represented by a contour in 2D is, as of now, hardly ever represented by using construction geometry in 3D. In order to document and specify which combinations can be related to real dimensioning situations in practice, the results of empirical work as described elsewhere in this paper were compiled into a dimensioning correspondence mapping as shown in the Appendix. Here each considered representation of a functional 2D dimension, as listed, has been related to a real dimensioning case and to its corresponding 3D representation. This has led to the formation of actual entity correspondence relationships and their mapping, as presented in the next sub-section.

3.3. Mapping of entity correspondence relationships

We aim to implement and translate into practice an approach for systematic and standardized 3D dimensioning employing explicit modeling within objectives and scope relating to design intent preservation and consistent/robust model alteration as presented earlier. In order to achieve this, structures and elements are required for representing the nominal shape of objects and related functional dimensions in the three-dimensional domain. Also needed is a correspondence that maps functionality between elements of the 2D dimensioning domain and the 3D dimensioning domain, as shown in Figure 2 (see again also Table 1 and Table 2). In the framework developed, concepts, structures, and correspondence relationships are designed as follows. Geometric features being defined and used within traditional 2D dimensioning as outlined above are related to the concept of explicit 3D entities, which in turn corresponds to actual elements of the geometric model of a MCAD system. That, nowadays, can be considered in most cases to be a boundary-representation (Brep) based model.

Explicit 3D entities are comprised of explicit topological entities and related geometric information pertaining to the Brep representation schema. In addition to explicit 3D entities, geometric constraints need to be considered as a means of providing the mechanisms required to maintain consistency between associated driving dimensions and resulting CAD model geometry. Geometric constraints are also required to provide functionality for consistent model alterations during CAD model exchange and re-design.

Elements for 3D dimensioning are represented using the concepts of explicit 3D entities (as previously outlined) and implicit 3D entities. Implicit 3D entities are comprised of topological entities and related geometric information as used within a Brep-based model. However, due to their structural properties, the actual elements of this entity domain are not an explicit part of the set of topological and geometric model entities used to represent the three-dimensional shape of an object. To provide functionality for placing 3D dimensions correctly and consistently within a MCAD system that is supporting explicit modeling, so-called *dimensioning correspondence mapping* (DCM) has been developed and implemented. This maps individual 2D dimensioning elements to 3D dimensioning elements. The DCM considers linear dimensions and angular dimensions as defined in [13, 14]. In accordance with ISO 14405–2, for the case of linear dimensions, correspondence mappings are provided for dimensions comprised of one feature and two features. Linear dimensions for transitions are also supported. For linear dimensions with one feature the mapping is based on relating specific entities from the 2D entity representation domain corresponding to a contour outline to specific entities from the 3D entity representation domain corresponding to either a boundary edge or a silhouette edge. For linear

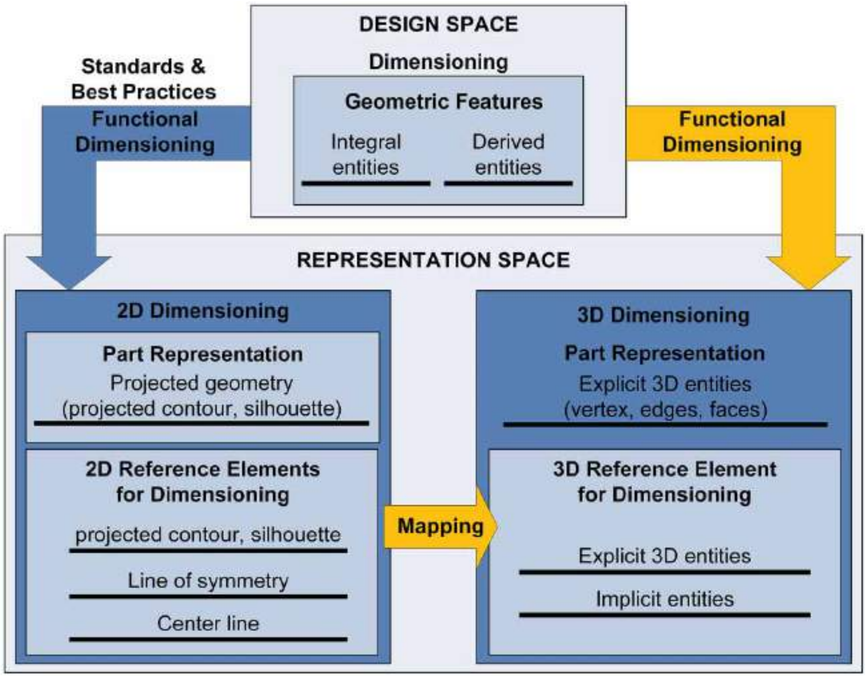


Figure 2. Overview of entity concepts, relationships, and their mapping.

dimensions with two features the mapping is based on relating geometric point spaces of spatial intersections of specific entities from the 2D entity representation domain corresponding, for example, to a contour outline or to an axis of a geometric feature, to geometric point spaces of spatial intersections of specific entities from the 3D entity representation domain, corresponding, for example, to a boundary edge or to a snap point. For angular dimensions and dimensioning of transitions, the mapping is based on combinations of the previously mentioned two mapping scenarios, additionally considering correspondences to projections of contour outlines for geometric point spaces of spatial intersections of specific entities from the 2D entity representation domain and correspondences to trimmed (implicit) edges for geometric point spaces of spatial intersections of specific entities from the 3D entity representation domain.

4. Example of implementation and use

4.1. Overview

To verify that the classification method, as proposed within the framework developed, is actually able to deal with real dimensioning situations, and in order to check to what extent present commercially available explicit modeling enabled MCAD systems are capable of supporting real 3D functional dimensioning, we asked collaborating partners from industry to provide actual data and information used in practice, in order to examine real test cases. The test case example reported in this paper relates to data and information provided by a partner company that produces milling machines for woodworking and marble working. The test case example drafted is for the spindle of a double-headed machine. The company has a division entirely dedicated to the design of spindles, which are one of the most critical components of a milling machine, because the effectiveness and precision of the machine relies on the performances and quality of this device.

For designing this kind of device, the main design parameters are the spindle power and the rotation speed. Among the various design constraints, the operative precision and the milling part accessibility are considered to be central. In order to balance these central design constraints, the designer has to find the best compromise between overall dimensions of the assembly and the component thickness. Smaller dimensions will result in reduced mass, higher acceleration, and better accessibility to the part to be milled, while decreasing the risk of obstructions and collisions. On the other hand, larger dimensions will result in increased mass, improved robustness, and reductions in vibration and distortion, while increasing the overall operative precision. Another aspect to be taken into account while designing this type of device concerns the configurability of the solution. Variants of the spindle may be required in order to operate in different power ranges and at different rotation speeds. In this case, the design process includes a review of previous solutions, so that the components can be adapted to suit the new required variant. The possible changes mainly refer to the length of the bearing seats, the diameter of the shafts, with particular reference to the sections where couplings with seals and other parts are required, the number and dimensions of fixing holes in the flanges, and gaps for the passage of the lubrication grease. During this review/redesign activity, it is crucial not to lose the functional requirements of the components, while also maintaining the original design intent.

4.2. Application context and settings

In the design solution used as a test example, the assembly includes a gear box, which splits the driving force onto two separate shafts, by means of spur gears. The gears are screwed at the extremities of the rotating shafts, which are guided by oblique ball bearings. Lubrication is provided by grease, which is contained by lip seals on rotating parts and O-rings on static couplings. The entire system is supported by boxes milled from blocks and turned flanges. The parts are screwed in order to allow easy disassembly for maintenance and part replacement. Dimensions and tolerances are selected in order to ensure correct couplings, sufficient clearances between the parts with relative motion, and adequate force transmission. The required operational precision is guaranteed by the optimal rigidity of each single component and the related quality of the production process, in terms of manufacturing tolerances and heat treatments.

The partner company has provided the 3D models and the related 2D mechanical drawings of the whole spindle assembly, as well as all the main components. The assembly is made up of about 175 components. Most of the components have an axial-symmetric shape. Hence, a significant number of dimensions are diameter dimensions and dimensions related to height and location of cylindrical shapes. Due to the relatively large number of screws in the assembly, a considerable number of various-sized holes with or without thread can also be found on individual components, resulting in many dimensions for thread diameter and length, as well as location dimensions, with most of these being related to the hole patterns.

About 90% of the dimensions are linear dimensions. Angular dimensions are mainly used to locate holes on circular patterns and for the chamfers. Among the various components of the spindle assembly, four have been selected for the analysis presented in this paper, namely the body, the bushing, the gear box and the hollow shaft as shown in Figure 3. To gain insight into and understanding of the main design requirements and constraints, and determine their relationships with the functional dimensioning, all 2D technical drawings provided by the industrial research collaboration partner have been inspected and analyzed. This process was supported by several interviews that were conducted with designers from the company's spindle design division who were familiar with the product design. In the next step, using the framework and classification developed, geometric elements and their respective representations in 2D and

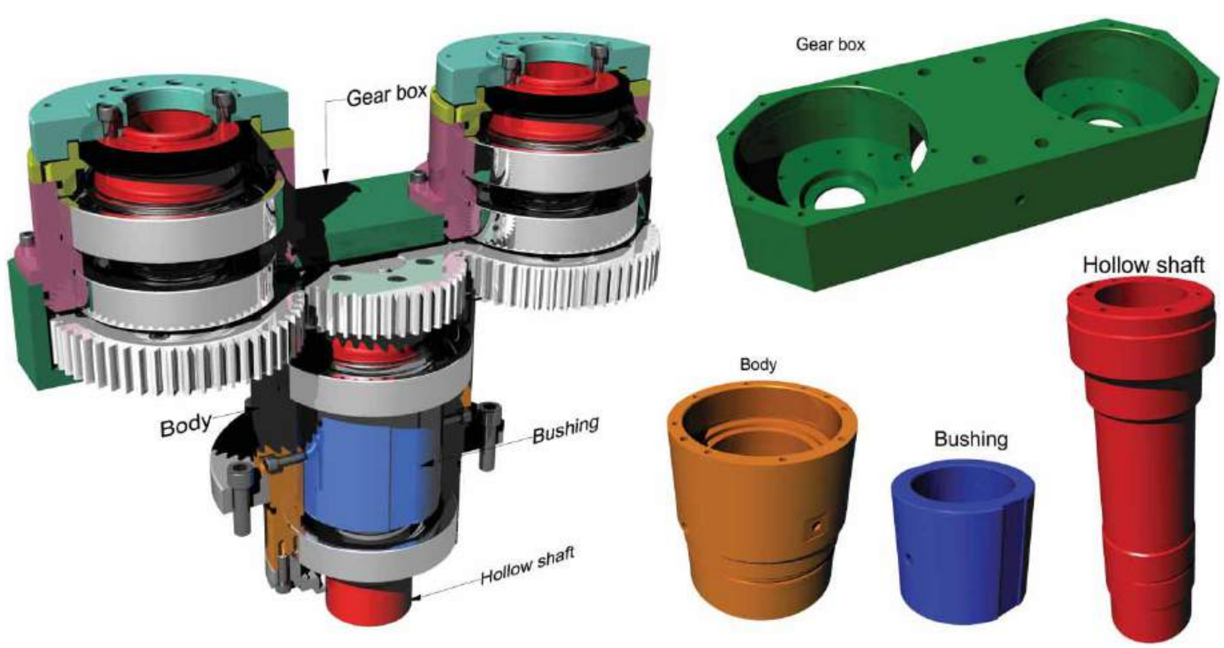


Figure 3. Examples of the double head milling machine spindle. From left to right: (a) 3D section of the spindle assembly, (b) individual assembly components referred to as gear box, body, bushing, and hollow shaft.

3D were identified in respect to the specification domain and associated with design intent and mappings of functional dimensions from 2D to 3D.

4.3. Evaluation of individual test cases

In order to better explain the previously described process, we will now present and evaluate three different design situations relating to the selection of different components, as described earlier. The examples refer to the grease slots between the hollow shaft and the bushing, the assembly of the grease nipple, and the interface between the gear box and the upper body. The exact locations of the components within the assembly, as related to each individual case, are shown in Figure 4.

Example 1. Grease slots on the bushing part

From a functional point of view, the depth of the slot (see Figure 5(a)) is a critical dimension, since it controls the extent of the cross-sectional area for the lubrication (grease) passage. The designer would determine such depth carefully in order to allow for adequate lubrication. This dimension should be maintained even if the diameter of the bushing is modified.

If the designer wishes to show this design intent on the 2D drawing, he/she will explicitly add the slot depth dimension on an appropriate view of the component (see dimension 2.75 mm in Figure 5(b)). If we make reference to the hierarchy of dimensions, as described in Table 1 in Part 2 of the ISO 14405 standard (cf. [16], Table 1), we find that this dimension is a linear dimension between two features of the integral-derived type. The integral feature is the straight segment representing one of the edges of the bottom face of the slot. The derived feature is the intersection point between the projection of the bushing part symmetry axis and the projection of the external cylindrical surface of the bushing part, which has been trimmed by the slot. In our proposed classification, this situation falls into the L2b.BT1 case (see also the table in the Appendix).

Consider specifying directly in 3D this type of dimension, as one can easily select the slot edge that represents one of the two reference elements to define this dimension (see Figure 5(c)). The other point required is not explicitly represented in the 3D model. Note that similar situations occur during the dimensioning of the key housing, and these situations are akin to case L2b.BI as listed in the Appendix.

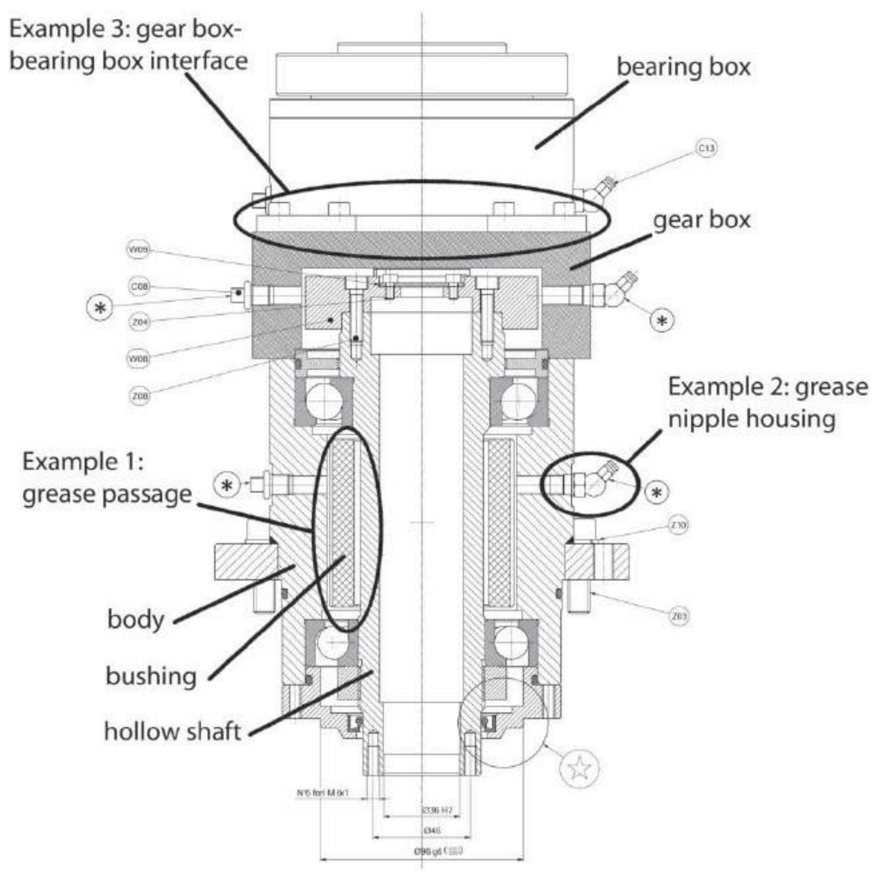


Figure 4. Annotated cross-section of the spindle along the bottom main shaft with indication of the locations of the three examples discussed in the evaluation.

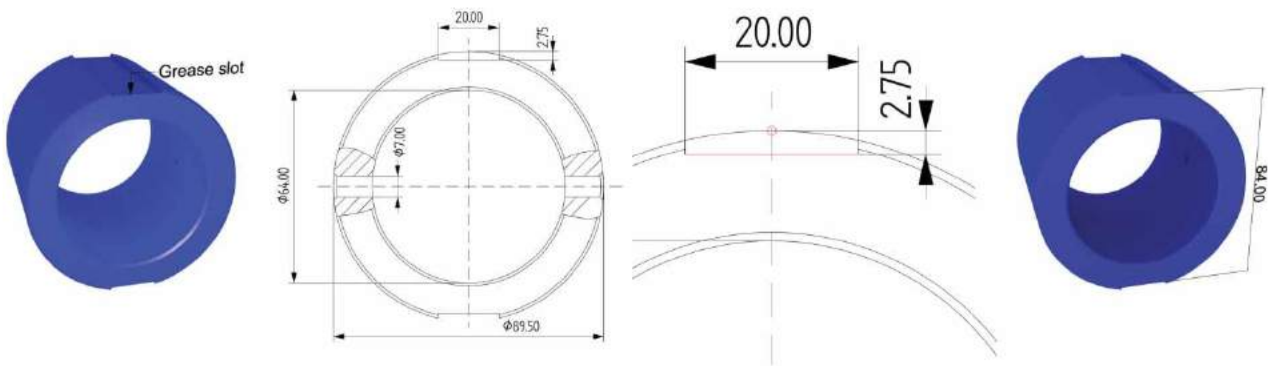


Figure 5. Examples of the grease slots on the bushing part. From left to right: (a) 3D section of the spindle assembly, (b) dimensioning as used within the original technical drawing, (c) enlarged section of the 2D dimensioning as used within the original technical drawing, (d) dimensioning as used within explicit modeling in 3D CAD.

At the present stage of 3D explicit modeling system development, the only possible solution to changing the slot depth is to position a dimension between the edges of the bottom face of the two symmetric slots, as shown in Figure 5(d). However, dimensioning in this manner is not equivalent to the original 2D dimensioning-based approach from either the functional point of view or the design intent representation aspect. This is because there is a missing point, which has been implicitly defined by the cylindrical surface and the symmetry axis. In fact, though, a computational algorithm capable of making explicit the required point could easily be implemented.

Example 2. Dimensioning and positioning of threaded insert for grease fitting installation

For permanent installation of a standard hydraulic grease fitting (Alemite fitting) in the form of a grease nipple, a threaded insert, constructed as a rectangular housing with a threaded hole at its center, is required on the spindle shaft housing, which is the external lower cylindrical part of the spindle body (cf. Figure 6(a)). From a functional point of view, several dimensions are significant for the correct dimensioning and positioning of the threaded insert. The depth of the recess should ensure a surface that is sufficiently flat to accommodate the grease nipple.

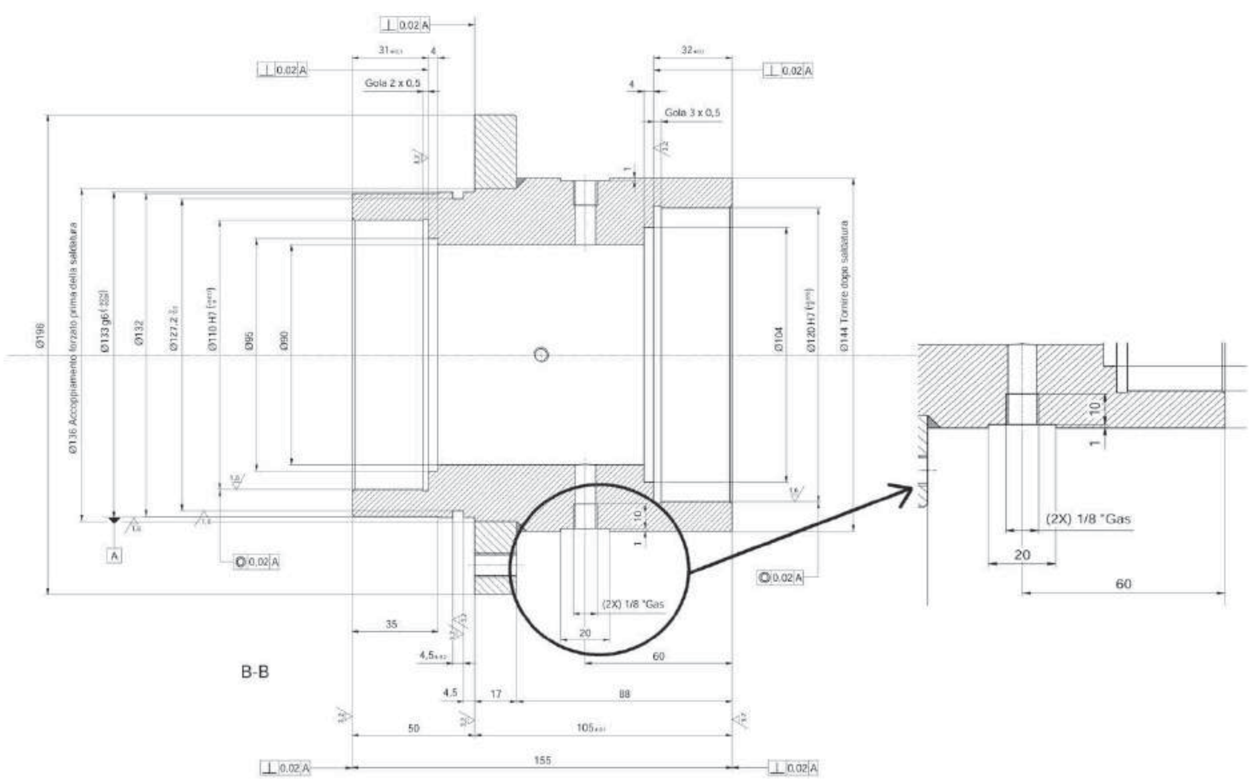


Figure 6. Original technical drawing of the spindle shaft housing. From left to right: (a) cross-section of the spindle shaft housing with complete dimensioning, (b) enlarged section of the original drawing showing details related to dimensioning of the threaded insert.

This represents a condition, which depends on the width and the depth of the housing. Such dimensions in turn are linked to the contour and implicit edges. Note that the depth of the thread should allow the nipple to be screwed in place flush with the insert. In case of design variations, such depth should be editable. The position of the hole should be fixed from the body end to the implicit axis of the hole.

As can be seen in Figure 6(b), five values are required for dimensioning of the threaded insert. Two of them refer to the thread of the threaded hole ((2X) 1/8″ GAS (1/8″: one-eighth of an inch) is the code containing the value for the thread diameter, 10 mm is the length of the thread). Another two values refer to the housing size (20 mm height, 1 mm depth) and one value refers to the location of the hole (60 mm).

First of all, the thread dimensions need to be considered in a manner separate from all the other dimensions as outlined hereafter. In both 2D and 3D representation, threaded holes are represented in a simplified way. In 2D, threaded holes need to be dimensioned by specifying the thread diameter and length. Note that for certain types of threads, the thread diameter is specified with a coding that does not correspond with the actual dimension of the thread diameter in an explicit and obvious manner. In 2D, the geometric elements used as reference for the dimensioning are the geometric entities used for the simplified representation of the thread. In 3D, threaded holes are usually represented as cylindrical or conical surfaces that are slicked on the hole surface. Hence, 3D dimensioning of a thread can then be related to the problem of dimensioning a cylinder or a cone. Currently, most, if not all, 3D explicit modeling systems consider the thread as an attribute of the hole, where changes in the thread dimensions can only be achieved by modification of the hole diameter. An approach more consistent with the engineering methodology of dimensioning would be to consider the hole as an attribute of the thread, because the thread dimensions are actually the dimensions that drive the dimensions of the hole (cf. entries L1a.C1 and L2a.CC1 in the Appendix).

Now we should consider the location dimension. In 2D this dimension is represented as a so-called *two feature* distance between the hole axis and the line representing the projection of the top face of the body (see again Figure 6(b)). As the top face and its boundary are explicitly represented as geometric entities in the 3D CAD model, they can be selected directly as a reference. However, the axis of the hole is not explicitly represented as a geometric entity in the 3D CAD model. In such cases, the MCAD system should provide functionality which allows the axis to be made explicit, rather than just providing a picking function to infer the axis from the cylindrical surface (this situation is similar to entry L2b.BI1 as listed in the Appendix).

Finally, consideration must be given to the two dimensions of the housing for the threaded insert. These specify its height and depth. In 2D, the height is defined by a so-called *one feature dimension*, which is related to the height of the housing face. In 3D one can use an appropriate boundary edge of the face to place the dimension (this dimensioning situation is similar to the dimensioning exemplified by entry L1a.B1 as listed in the Appendix). In the case of the depth, the 2D dimension is defined by the two feature distance between the planar face of the housing and the cylindrical face of the outer part of the body, which in turn, in the 2D view, corresponds to a silhouette edge. Generally, in such a dimensioning scenario, current 3D MCAD systems do not permit the user to make a reference to silhouette edges, because they cannot be explicitly represented on the CAD model. However, in most cases, such systems do allow the user to select a polar point on a circumference, which provides a workaround in some cases, but not in the one presented here, where there is no way to place the dimension correctly (see also entry L2a.BB1 as listed in the Appendix).

Example 3. Interface units screwed between the gear box and the upper body

In the spindle assembly, two bearing boxes are screwed on top of the gear box. Screw holes are present in the bearing boxes and in the gear box. The holes in the bearing boxes are simple through holes as the ones in the gear box are blind threaded holes. The holes are located along a circular pattern, with a non-uniform angular positioning. The centers of the hole patterns are located along the axis of the mating cylindrical surfaces on the gear and bearing boxes (see Figure 7(a)).

In the 2D drawings (cf. Figure 7(b)), the position of the holes is defined by using dimensioned reference geometry: the diameter of a circumference that defines the distance of the holes from the center of the pattern (106 mm dimension in Figure 7(b)); and the angular distance between the lines connecting the center of the pattern and the centers of two holes (45° and 30° dimensions as depicted in Figure 7(b)).

Due to the reference geometry, which in this case is not an explicit element of the component shape, in respect to the hierarchy of dimensions introduced by the standard, we consider the diameter of the pattern to be a linear one feature derived dimension. The pattern diameter dimension falls into the L1b.C1 case as listed in the table in the Appendix. By following the same approach, we can consider the angular location of the holes to be an angular two feature derived-derived dimension.

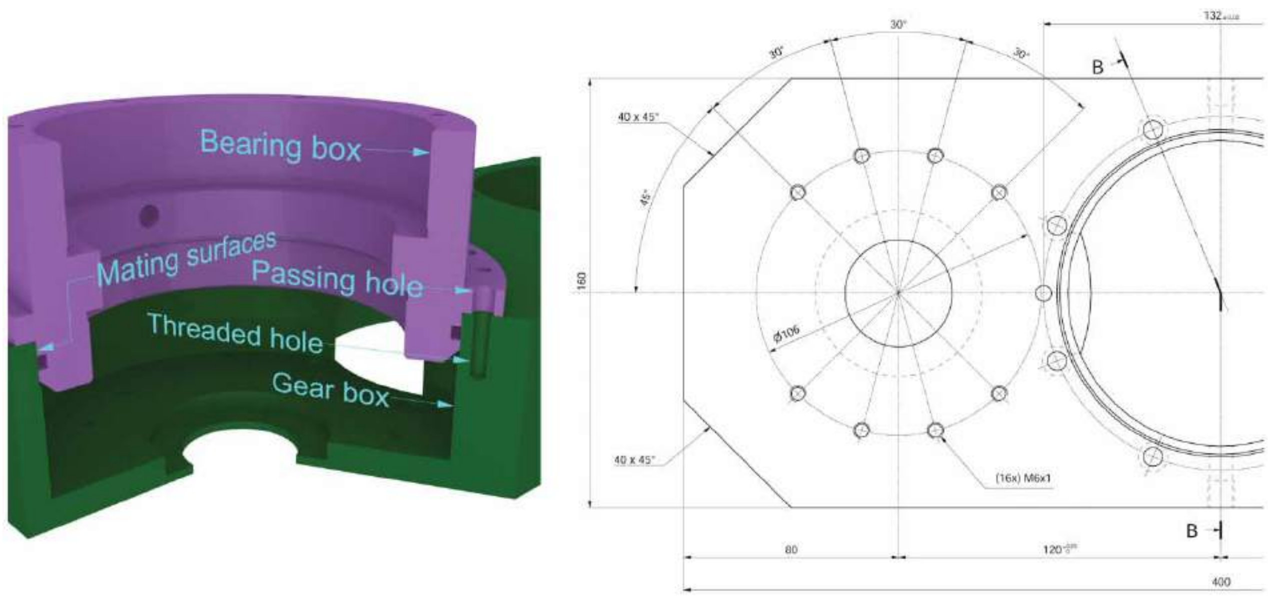


Figure 7. Location of the connection holes between the gearbox and the upper body. The definition of the position of the holes depends on several implicit geometries and constructions. From left to right: (a) 3D geometry of the bearing box sub-assembly with cross-section, (b) section of the original drawing showing the left part of the gear box.

If we look at our representation table, it is easy to say that this situation refers to an axis-axis condition in 2D and a derived-derived condition in 3D. However, due to the particular nature of the shapes that have been analyzed up to now, at the moment entries related to cases of angular dimensions have not been developed within our classification framework. Nevertheless, it is quite evident that, since the required entities for dimensioning are derived entities, an efficient framework for direct 3D dimensioning needs to provide algorithms to identify and make explicit the geometric elements required as reference elements for dimensioning.

5. Conclusions

In this paper a novel approach has been presented and discussed, aimed at translating traditional design intent representation from 2D into 3D. The route taken to formulate a framework for defining and implementing explicit 3D functional dimensioning started with an analysis of the concepts, criteria, and entities related to functional dimensioning as used traditionally by engineers in mechanical drawings and specified in the standards for Technical Product Documentation. Issues of correctly identifying the elements used in 2D drawings to define dimensions and their correct mapping into corresponding 3D elements capable of supporting the dimensioning definition in 3D were among the central problems addressed, to conceptualize and implement explicit 3D functional dimensioning in a syntactically consistent and semantically sound manner. As became evident during both theoretical analysis and empirical work, at present the dimensioning concepts, as defined by the standards related Geometric Product Specification, and the explicit modeling functionality available within the MCAD system and used as an enabling technology to translate and implement 3D dimensioning, are neither sufficiently structured nor coherent enough to allow for sound and complete 3D functional dimensioning, as traditionally applied in mechanical engineering.

In particular, it could be shown that several shortcomings of the explicit modeling functionality as provided within most of today's MCAD systems are related to either an incomplete access to, or even the total absence of, geometric elements such as the referenced or implicit entities that are required to support 3D functional dimensioning. As could be demonstrated in some cases, though, modifying or adjusting the definition and implementation of the dimensioning correspondence mapping, and the related sets of 3D elements for dimensioning, can overcome the shortcomings arising from the absence of proper entity access or the actual absence of elements from the model. However, a more appropriate and sound solution in this context lies in the efforts to improve and extend current user interface functionality of commercially available MCAD systems in respect to their explicit modeling capabilities. This represents a precondition, necessary for adjusting the current inadequate levels of usability and affordance to current needs in practice, in order to provide a sound and consistent alternative to feature-based modeling in respect to design intent representation and preservation. This approach will undoubtedly gain further in importance with today's increasing demands for cross-platform exchangeable and re-usable CAD models.

ORCID

Ferruccio Mandorli http://orcid.org/0000-0003-4864-5265
Harald E. Otto http://orcid.org/0000-0002-4580-0429
Roberto Raffaeli http://orcid.org/0000-0003-0301-454X

References

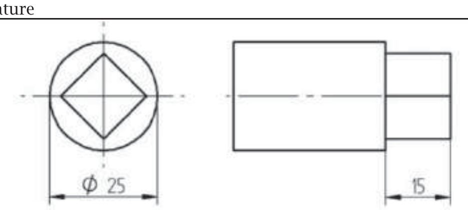
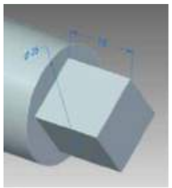
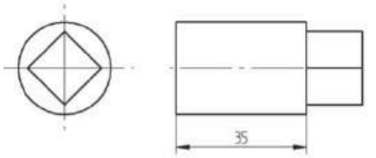
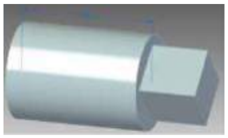
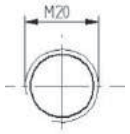
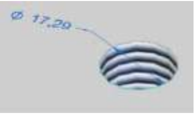
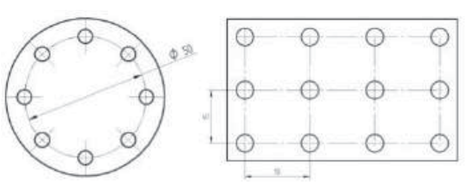
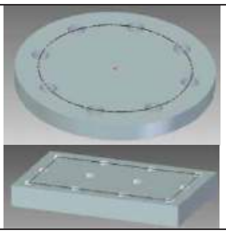
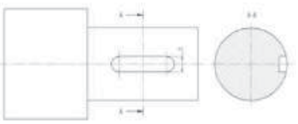
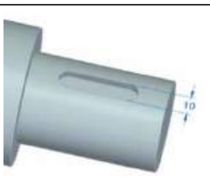
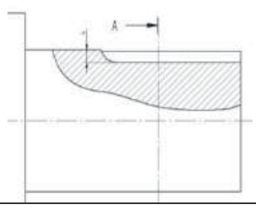
[1] Aberdeen Group (ed.): The Transition from 2D Drafting to 3D Modeling: Benchmark Report – Improving Engineering Efficiency, Aberdeen Group, Boston, MA, USA, 2006.
[2] Alducin-Quintero, G.; Rojo, A.; Plata, F.; Hernandez, A.; Contero, M: 3D Model Annotation as a Tool for Improving Design Intent Communication: A Case Study on its Impact in the Engineering Change Process, In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, August 12–15, Chicago, IL, USA, 2012, Vol. 2, 349–356. http://dx.doi.org/10.1115/DETC2012-70872
[3] Anderl, R.; Mendgen, R.: Analysing and Optimising Constraint-Structures in Complex Parametric CAD Models, In B. Bruderlin and D. Roller (eds.) Geometric Constraint Solving and Applications, Springer, Berlin, Germany, 1998, 58–81. http://dx.doi.org/10.1007/978-3-642-58898-3_4
[4] Ault, H. K.: Using Geometric Constraints to Capture Design Intent, Journal for Geometry and Graphics, 3(1), 1999, 39–45.
[5] Berti, S.; Germani, M.; Mandorli, F.; Otto, H. E.: Design of Product Families: An Example within a Small and Medium Sized Enterprise, In Proceedings of 13th International Conference on Engineering Design, August 21–23, Glasgow, UK, 2001, 507–514.
[6] Bodein, Y.; Rose, B.; Caillaud, E.: Explicit Reference Modelling Methodology in Parametric CAD System, Computers in Industry, 65(1), 2014, 136–147. http://dx.doi.org/10.1016/j.compind.2013.08.004
[7] Bracewell, R.; Wallace, K.; Moss, M.; Knott, D.: Capturing Design Rationale, Computer-Aided Design, 41(3), 2009, 173–1786. http://dx.doi.org/10.1016/j.cad.2008.10.005
[8] Butterfield, W.; Green, M.; Scott, D.; Stoker, W.: Part Features for Process Planning, Technical Report R-85-PPP-03, CAM-I Inc., Arlington, TX, USA, 1985.
[9] Camba, J.; Contero, M.; Johnson, M.; Company, P.: Extended 3D Annotations as a New Mechanism to Explicitly Communicate Geometric Design Intent and Increase CAD Model Reusability, Computer-Aided Design, 57, 2014, 61–73. http://dx.doi.org/10.1016/j.cad.2014.07.001
[10] Colombo, G.; Mandorli, F.: Evolution in Mechanical Design Automation and Engineering Knowledge Management, In M. Bordegoni and C. Rizzi (eds.) Innovation in Product Design: From CAD to Virtual Prototyping, Springer, London, UK, 2011, 55–78. http://dx.doi.org/10.1007/978-0-85729-775-4_4
[11] Henderson M. R.: Representing Functionality and Design Intent in Product Models, In Proceedings of the ACM Symposium on Solid Modelling and Applications, May 19–21, Montreal, Quebec, Canada, 1993, 387–396. http://dx.doi.org/10.1145/164360.164485
[12] ISO 10209:2012 - Technical Product Documentation - Vocabulary - Terms Relating to Technical Drawings, Product Definition and Related Documentation.
[13] ISO 129-1:2004 – Technical Drawings – Indication of Dimensions and Tolerances – Part 1: General Principles.
[14] ISO 129-2:2013 – Technical Product Documentation – Indication of Dimensions and Tolerances – Part 2: Dimensioning of Mechanical Engineering Drawings.
[15] ISO 14405-1:2010 - Geometric Product Specification (GPS) – Dimensional Tolerancing - Part 1: Linear Sizes.
[16] ISO 14405-2:2011 - Geometric Product Specification (GPS) – Dimensional Tolerancing - Part 2: Dimensions other than Linear Sizes.
[17] ISO 14660-1:1999 - Geometric Product Specification (GPS) – Geometrical Features - Part 1: General Terms and Definitions.
[18] Iyer, G. R.; Mills, J.J.: Design Intent in 2D CAD: Definition and Survey, Computer-Aided Design and Applications, 3(1–4), 2006, 259–267. http://dx.doi.org/10.1080/16864360.2006.10738463
[19] Johnson, M. D.; Diwakaran, R. P.: Assessing the Effect of Incentive on Computer-Aided Design Intent, In Proceedings of the ASME International Design Engineering Technical Conferences and Computers in Engineering Conference, San Diego, CA, USA, 2009, 523–532. http://dx.doi.org/10.1115/DETC2009-86644
[20] Kimura, F.; Suzuki, H.: A CAD System for Efficient Product Design Based on Design Intent, CIRP Annals Manufacturing Technology, 38(1), 1989, 149–152. http://dx.doi.org/10.1016/S0007-8506(07)62672-9
[21] Kyprianou, M.R.: Shape Classification in Computer Aided Design, Ph.D. Dissertation, University of Cambridge, Cambridge, U.K., 1980.
[22] Otey, J. M.; Company, P.; Contero, M.; Camba, J. D.: A Review of the Design Intent Concept in the Context of CAD Model Quality and Metrics, In Proceedings of the ASEE Annual Conference and Exposition, June 15–18, Indianapolis, IN, USA, 2014, Paper-No.: 9556.
[23] Peng, X.; McGary, P.; Johnson, M; Yalvac, B.; Ozturk, E.: Assessing Novice CAD Model Creation and Alteration, Computer-Aided Design and Applications, PACE (2), 2012, 9–19.
[24] Quintana, V.; Rivest, L.; Pellerin, R.; Venne, F.; Kheddouci, F.: Will Model-Based Definition Replace Engineering Drawings throughout the Product Lifecycle? A Global Perspective from Aerospace Industry, Computers in Industry, 61(5), 2010, 497–508. http://dx.doi.org/10.1016/j.compind.2010.01.005
[25] Rynne, A.; Gaughran, W.: Cognitive Modeling Strategies for Optimum Design Intent in Parametric Modeling, Computers in Education Journal, 18(3), 2008, 55–68.
[26] Shah, J.; Mäntylä, M.: Parametric and Feature Based CAD/CAM: Concepts, Techniques, and Applications, John Wiley & Sons, New York, NY, USA, 1995.
[27] Whitney, D. E.: The Role of Key Characteristics in the Design of Mechanical Assemblies, Assembly Automation, 26(4), 2006, 315–322. http://dx.doi.org/10.1108/01445150610705236

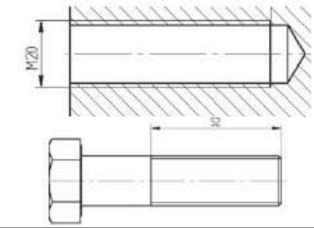
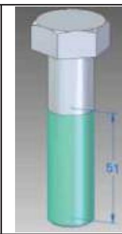
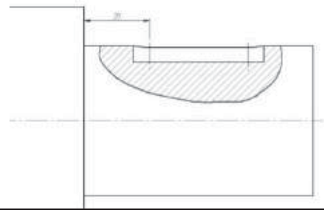
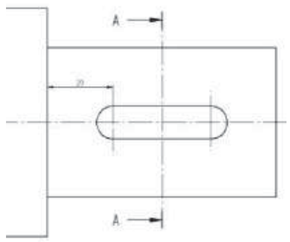
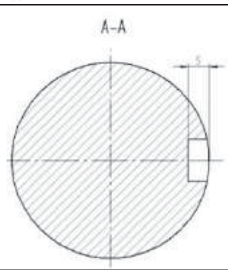
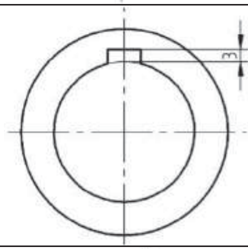
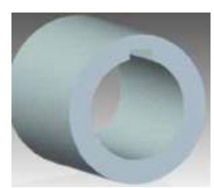
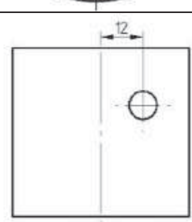
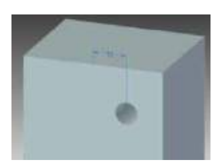
[28] Yang, J.; Han, S.: Repairing CAD Model Errors Based on the Design History, Computer-Aided Design, 38(6), 2006, 627–640. http://dx.doi.org/10.1016/j.cad.2006.02.007

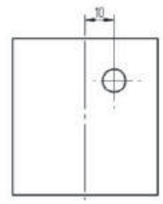
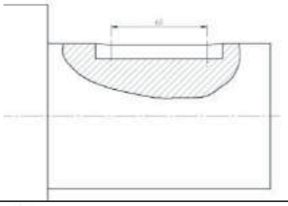
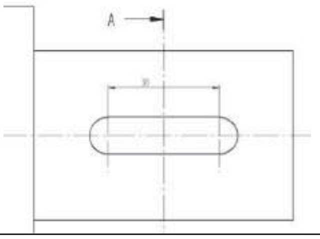
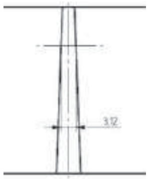
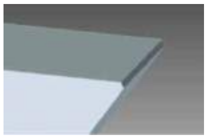
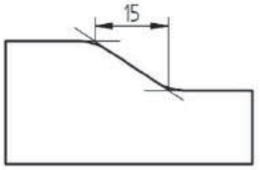
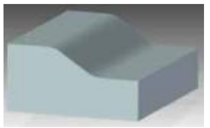
[29] Zhang Y.; Luo, X.: Design Intent Information Exchange of Feature-based CAD Models, In Proceedings of the World Congress on Computer Science and Information Engineering, March 31-April 2, Los-Angeles, CA, USA, 2009, 11–15. http://dx.doi.org/10.1109/CSIE.2009.536

Appendix

The description of the dimensioning correspondence mapping given in this appendix is structured for linear dimensions reflecting the classification given in the standard [16] within the first hierarchical level for cases of one feature, two features, and transition. It provides details on the domain (a, b) and co-domain (b, c) of this partial, non-continuous map in respect to columns as follows. From left to right: (a) entity class of the 2D representation domain (cf. Table 2, Figure 2), (b) concrete example from the domain of 2D functional dimensioning, (c) corresponding entity class of the 3D representation domain and knowledge in the form of know-how on implementing corresponding dimensioning (semantics) by using the functionality of 3D explicit modeling, (d) corresponding concrete example from the domain of 3D functional dimensioning, (e) reference index to unambiguously identify and refer to each single map specified.

DL1: One Feature				Ref.
Contour		Boundary Edge: Placed by selecting the edge.		L1a.B1
Contour		Implicit Geometry: Reference to silhouette edge is not possible. Placed using two center points.		L1a.S1
Contour		Construction Geometry: It depends on how the thread is defined in the 3D model.		L1a.C1
Pattern		Construction Geometry: Reference to construction geometry is not possible. It is impossible to place the dimension.		L1b.C1
DL2: Two Features				
Contour-Contour		Boundary - Boundary: Placed by selecting the edges.		L2a.BB1
Contour-Contour		Boundary - Implicit: One reference is a silhouette edge. Reference to the silhouette edge is not possible. Placed using two mid points.		L2a.BS1

Contour-Contour		Construction Geometry: It depends on how the thread is defined on the 3D model.		L2a.CC1
Contour-Axis		Boundary - Implicit: One of the references is an axis. Reference to the axis is not possible. Placed using two center points.		L2b.BI1
Contour-Point		Boundary - Implicit: One of the references is an axis. Reference to the axis is not possible. Placed using two center points.		L2b.BI2
Contour-Point		Boundary - Pt.Snap: One of the references is a snap point of an approximated geometry. Placed using two center points.		L2b.BI3
Contour-Projection		Boundary - Trimmed: One of the references is a portion of geometry that has been trimmed. Reference to the trimmed portion is not possible. It is impossible to place the dimension.		L2b.BT1
Axis-Point		Implicit - Pt.Snap: One of the references is a symmetry plane. Reference to the plane is not possible. Placed using a mid point and a center.		L2c.II1

Axis-Point		Implicit - Pt.Snap: One of the references is an axis. Reference to the axis is not possible. One of the references is a snap point of an approximated geometry. Placed using two center points.		L2c.II2
Axis-Axis		Implicit - Implicit: The two references are two axes. Reference to the axes is not possible. Placed using two center points.		L2c.II3
Point-Point		Pt.Snap - Pt.Snap: The two references are entities of approximated geometry. Placed using two center points.		L2c.II4
Point-Point		Pt.Snap - Pt.Snap: Reference to construction geometry. It is impossible to place the dimension.		L2c.II5
DL3: Transition				
Projection-Projection		Trimmed - Trimmed: The references are portions of geometry that have been removed. Reference to the trimmed portions is not possible. It is impossible to place the dimension.		L3a.TT1
Projection-Projection		Trimmed - Trimmed: The references are portions of geometry that have been removed. Reference to the trimmed portions is not possible. It is impossible to place the dimension.		L3a.TT2