Exact Representation Methodology of 3D Annotation Information in Model Based Definition for Product Digitization

Yazhou Chen[1], Songqing Xu[1], Yi Gan[2] and Qihao Zhang[1]

[1]Jimei University, yazhouchen@jmu.edu.cn, 1413444860@qq.com, 2531092570@qq.com
[2]University of Shanghai for Science and Technology, Ganyi@usst.edu.cn

Corresponding author: Yazhou Chen, yazhouchen@jmu.edu.cn

Abstract. Given that a geometrical dimensioning and tolerancing (GD&T) and geometric form features cannot form a correct semantic association relationship based on the semantic analysis of 3D dimension annotations, different types of dimension semantic association objects have been defined. In this study, considering a 3D linear size as an example, two types of judging rules are proposed, and related algorithms are established. An exact representation methodology, which can automatically check whether a 3D dimension semantic-related object is correct or not, is proposed. Additionally, an algorithm for automatically converting 3D annotation-associated objects is established. A prototype system is developed based on a UG NX1847 application program interface; this system can automatically convert the points with wrong semantic association objects, edges with wrong semantic association objects, and faces with wrong semantic association objects in 3D dimension annotations. The 3D exact representation methodology proposed in this study helps improve GD&T definition quality in a model-based definition (MBD) model. It is helpful to realize an MBD model for machine-readable and machine-processable data.

Keywords: 3D Dimension Annotation, Model Based Definition, Exact Representation, Model Quality, Automatic Convert.
DOI: https://doi.org/10.14733/cadaps.2023.856-871

1 INTRODUCTION

Model-based definition (MBD) is an advanced digital definition methodology wherein product geometric and non-geometric information can be integrated into a 3D computer-aided design (CAD) model [1,2]. Geometric information in an MBD model refers to a geometric form feature information, such as a extrude feature or a rotate feature [3]; the non-geometric information in an MBD model generally refers to product manufacturing information (PMI). An MBD design model must be reused in computer-aided engineering (CAE) [4], computer-aided manufacturing (CAM) [5], computer-aided process planning (CAPP) [6], computer-aided tolerancing (CAT), and coordinate measuring machine (CMM) systems [7,8]. To improve the reuse efficiency of an MBD model, the quality of an MBD model must be improved. The MBD model quality is determined by the geometric feature modeling quality and the PMI annotation quality.

Numerous researchers focus on the geometric feature modeling quality. Amadori believes that to improve the quality of a geometric model, the flexibility and robustness of the geometric model must be considered. Note that flexibility refers to the ability to meet the configuration requirements within a specific range of products, whereas robustness refers to the ability to make no mistakes or collapse during design changes to a geometric model [9]. Gebhard believes that the quality of a geometric model can be improved in three aspects, namely editable, obvious, and reusable [3]. Company thinks the quality of a geometric model can be improved in five aspects: valid, complete, consistent, concise, and effective [10-13]; herein, valid implies that a CAD model does not contain any error information or warning information, complete implies that a CAD model has achieved a customer's design requirement, consistent implies that a CAD model will not crash when editing, concise implies that a CAD model contains no irrelevant or duplicate information, and effective implies that a CAD model can express design intent.

Few researchers have focused on the annotation quality of PMI in an MBD model. Ruemler et al. proposed two forms of PMI in an MBD model: graphical presentation and semantic representation [14-17]. They highlighted that a graphical presentation PMI is suitable for human reading, but is difficult to be processed automatically by a computer; semantic representation PMI is suitable to be automatically processed by a computer; and an MBD model data automatic consumption can be promoted. Lipman reported that a PMI annotation semantic limitation and a graphical PMI annotation limitation exist. The semantic quality of a PMI annotation can be improved in three aspects: structure, parameters, and geometry; moreover, the graphical quality of a PMI annotation can be improved from six aspects: visibility, layout, location, orientation, lines, and text [18,19]. Agovic reported that PMI must be associated with a processing feature or a surface feature in a 3D CAD model because CAM/CMM software cannot further process PMI, which is associated with an auxiliary geometry element or an edge automatically [20,21].

In contrast, studies detailing how the PMI definition quality in an MBD model can be improved are limited. PMI includes geometrical dimensioning and tolerancing (GD&T), surface textures, process notes, material specifications, and weld symbols [14]. According to ISO 14405-1 [22] and ISO 14405-2 [23], dimensions can be subdivided between "linear size" dimensions and dimensions other than linear size (i.e., distance), and similarly for angular dimensions according ISO 14405-3 [24]. Geometrical type tolerances can be subdivided into form tolerances, orientation tolerances, location tolerances, and run-out tolerances [25,26]. Furthermore, the form tolerances can be subdivided into four types, such as straightness, flatness, roundness and cylindricity. The orientation tolerances can be subdivided into three types, such as parallelism, perpendicularity and angularity. Owing to the numerous PMI annotation types, this study focused only on a 3D dimension annotation. A 3D dimension annotation exact representation methodology was proposed to improve the definition quality of PMI in an MBD model.

2 DESIGNING A 3D DIMENSION INFORMATION EXACT REPRESENTATION METHODOLOGY

2.1 Connotation of a 3D Dimension Information Exact Representation

Realizing a 3D dimension information exact representation implies that no invalid 3D dimension annotations, isolated 3D dimension annotations, and repeated 3D dimension annotations exist in an MBD model. The format and the content of an added 3D dimension annotation must comply with ISO 16792 [2], and there is no error or ambiguity in the associated objects of a 3D dimension annotation [23]. A bidirectional logical association between a 3D dimension annotation feature and a geometric form feature can be established, thus making it easy to look up the corresponding 3D dimension annotation from a geometric form feature and vice versa.

The purpose of establishing the exact representation methodology of a 3D dimension annotation is to improve the definition quality of an MBD model, realize the semantic representation of PMI [27,28], and improve the reuse efficiency of an MBD model.

2.2 Associated Objects Analysis of a 3D Dimension Annotation

From the perspective of type, 3D dimension annotations in an MBD model can be divided into a linear sizes, angular sizes and distance dimension [23]. From the perspective of function, 3D dimension annotations can be divided into form and location dimensions. From the perspective of property, 3D dimension annotations can be divided into a performance dimension, which plays the role of constraint, and a reference dimension, which plays the auxiliary role. From the perspective of orientation, 3D dimension annotations can be divided into a horizontal, vertical, perpendicular, and parallel dimensions.

A "linear size" dimension is used to represent the length, width, height and depth of a feature. A "linear size" dimension needs to be associated with two geometric elements, and these two geometric elements should be the surfaces of the same form feature to achieve an exact representation. A "linear distance" dimension is used to represent the distance between two geometric features. A "linear distance" dimension needs to be associated with two geometric elements which belong to two different features. These two geometric elements should be two integral features or an integral feature and a derived feature or two derived features. The derived feature should be the axis of a rotation feature, such as a cylinder or a hole, to achieve an exact representation [22].

A "linear size" dimension can be used to represent the radius, diameter, or sphere radius, which is used to determine the form of a hole feature, a cylinder feature, or a sphere feature. In this situation it must be associated with one geometric element. To achieve an exact representation of this size, the associated geometric element type should be the surface of a cylinder or sphere.

A "linear size" dimension can be used to determine the thickness of a geometric feature. In this situation it must be associated with two geometric elements. To achieve an exact representation of this size, the type of these two geometric elements should be the surface of a geometric form feature.

An "angular size" dimension plays a shaping role used to determine a geometric feature form, and must be associated with two geometric elements. To achieve an exact representation, the type of the two associated geometric elements should be the surface of the same form feature. An "angular distance" dimension is used to determine the location of two different geometric features, and must be associated with two geometric elements. One of these two geometric elements may be a derived feature, such as the axis of a cylinder or a hole, and the other one may be the surface of a geometric feature, or both the derived axis of two revolve features [24].

Other dimension associated objects, such as a chamfer, fillet, bevel, and coordinate dimensions, can be analyzed similarly.

2.3 Methodology for the 3D Dimension Information Exact Representation

There are two modes to process the error associated objects in 3D dimension annotations. Mode 1: check 3D dimension annotations and adjust the associated error objects manually; Mode 2: check and convert 3D dimension annotations through a computer program. Using mode 1 is time-consuming, laborious, and inevitably contains omissions; mode 2 has the characteristic of high efficiency and good reliability. Therefore, in this study, mode 2 was adopted.

Numerous CAD software has an automatic inspection tool for model quality check, such as Q-Checker in CATIA, Check-mate in UG NX, Design Checker in Solidworks, and ModelCheck in Creo. The test revealed that these tools could not automatically process the content proposed in this study [13]. Therefore, a methodology for a 3D dimension information exact representation was proposed, as follows.

Step1: Through the application program interface (API) of CAD software, acquire the 3D dimension annotations information, including a dimension identifier, dimension type, dimension value, dimension tolerance, dimension associated object identifier, and dimension associated object type.

Step2: Establish different types of 3D dimension annotations association rules. Some rules can be established based on the associated objects' semantic analysis.

Step3: Check the associated objects of 3D dimension annotations one by one. Using these rules established in step2 to judge whether the associated object of a 3D dimension annotation is correct or not, and the error associated object types are marked.

Step4: 3D dimension associated object conversion. Process these 3D dimensions with error associated objects through a program and convert them to associated with correct geometric elements.

3 3D DIMENSION INFORMATION EXACT REPRESENTATION

3.1 Acquiring the 3D Dimension Annotations Information

Because the author has used UG NX1847 before and is familiar with UG NX1847, the UG NX1847 software was used as the 3D part modeling platform in this study. UG NX1847 provides an API for users to extract all types of 3D dimension annotations information. An XML schema file is defined below, and it can be used to store the extracted 3D dimensions annotations information.

Dimension type name	*Value*
UF_dim_horizontal_subtype	1
UF_dim_vertical_subtype	2
UF_dim_parallel_subtype	3
UF_dim_cylindrical_subtype	4
UF_dim_perpendicular_subtype	5
UF_dim_angular_minor_subtype	6
UF_dim_angular_major_subtype	7
UF_dim_arc_length_subtype	8
UF_dim_radius_subtype	9
UF_dim_diameter_subtype	10

Table 1: Different types of dimensions in UG NX.

<Dimensions Dimensions_ID="" Dimensions_Type="">

<Dim_Value Value="" Up_limit="" Bottom_limit=""/>

<First_Associativity First_Associativity_ID="" First_Associativity_Type=""/>

<Second_Associativity Second_Associativity_ID="" Second_Associativity_Type=""/>

</Dimensions>

In the XML schema file above, Dimensions_ID indicates a unique identifier for each dimension. As UG software uses a tag value to identify each object, the Dimensions_ID value equals a UG tag value. Dimensions_Type indicates the type of a three dimension. From the type of a three dimension, it can be judged whether the object associated with it is correct or not. Different dimension types and their type value defined in UG are listed in Table 1. First_Associativity_ID indicates an identifier of the first associated object of a dimension, and Second_Associativity_ID indicates an identifier of the second associated object of a dimension. If a dimension associated object is missing and only one associated object is available, then the second associated object value is null. First_Associativity_Type indicates the type of the first associated object of a dimension, and

Second_Associativity_Type indicates the type of the second associated object of a dimension. 3D dimension possible associated object types may be a UF_face_type, a UF_edge_type, a UF_point_type, and so on.

3.2 Establishing 3D Dimension Associated Object Judgment Rules

Ambiguities or errors in 3D dimension annotations may be: (1) the type of a geometric element with which a 3D dimension associated with it is wrong, for example, a 3D linear size associated geometric element type is a point or an edge; (2) a 3D dimension associated geometric element does not belong to a corresponding geometric shape feature. For example, a 3D dimension is incorrectly associated with a datum plane at the same position as a geometric feature plane; a center-distance dimension between two holes has not been associated with the axis of the hole but is incorrectly associated with a datum axis which at the exact position of this axis.

Therefore, herein, two types of rules were proposed. The first type of rules are used to judge whether the associated geometric element type is correct or not, and the second type of rules are used to judge whether the associated geometric element belongs to a constrained geometric shape feature or not.

3.2.1 First type of rules

Rule1.1: One of the 3D linear sizes associated with objects is UF_point_type. Thus, the type of the association object is an error, set the Flag1 value to Error.

Rule1.2: One of the 3D linear sizes associated with objects is UF_edge_type. Thus, the type of the association object is an error, set the Flag1 value to Error.

Rule1.3: Both of the 3D linear sizes associated objects are UF_face_type, then the type of the association object is correct, set the Flag1 value to True.

Rule1.4: Both of the 3D linear distance dimensions associated objects are UF_draft_cyl_cntrln_subtype. Thus, the type of the association objects is correct, set the Flag1 value to True.

Rule1.5: One of the 3D linear distance dimensions associated object is UF_draft_cyl_cntrln_subtype, and another 3D linear distance dimension associated object is UF_face_type. Thus, the type of the association object is correct, set the Flag1 value to True.

3.2.2 Second type of rules

Rule2.1: For the two associated objects of the same 3D linear size or linear distance dimension, the parent feature of the first associated object was identified with the parent feature of the second associated object; subsequently, the body of these associated objects was consistent. Thus, the Flag2 value was set to True.

Rule2.2: For the two associated objects of the same 3D linear size or linear distance dimension, if the parent feature of the first associated object was not identified with the parent feature of the second associated object, then the body of these associated objects was different. Set the Flag2 value to Error.

Similar rules can be established for an angular size and angular distance dimensions.

3.3 Judging a 3D Dimension Associated Object

Based on the two types of rules mentioned above, an algorithm was designed to judge whether the associated objects of a 3D linear size are correct or not (Figure 1).

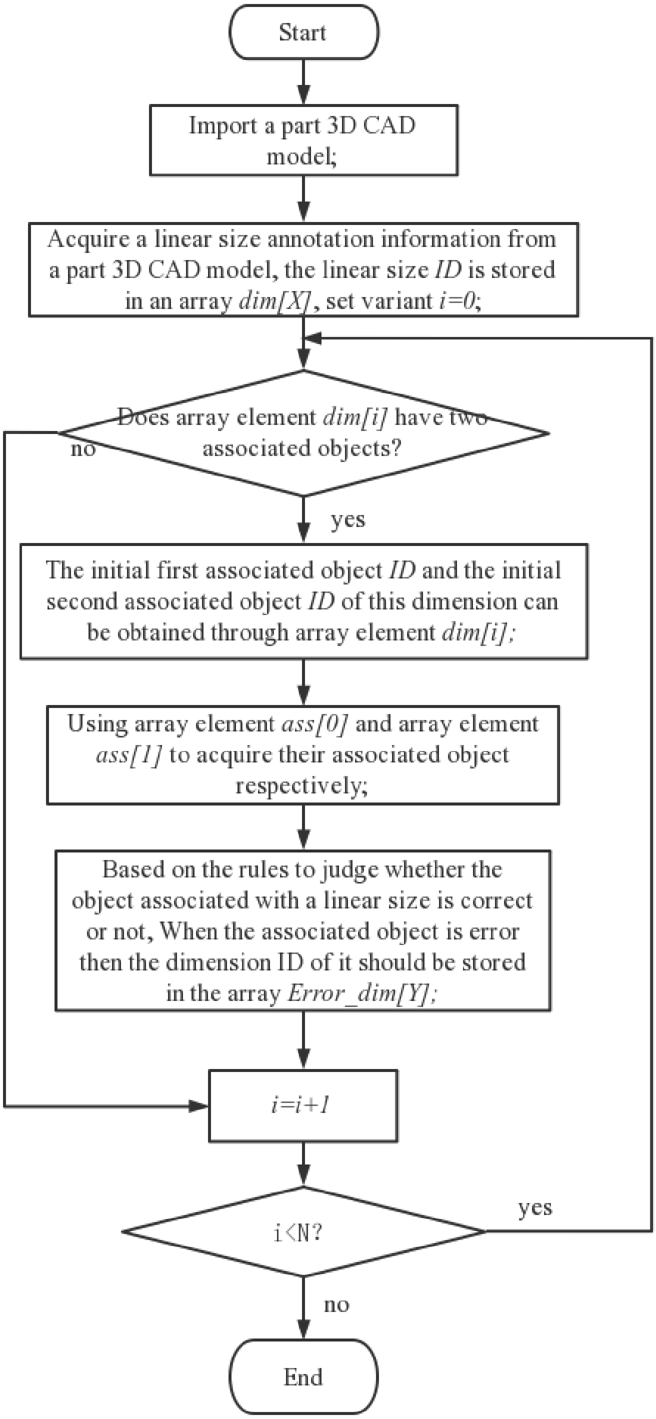


Figure 1: Flow program of a linear size associated with object judgment.

Step1: Start the algorithm.

Step2: Import a part 3D CAD model.

Step3: Acquire 3D linear size annotations information from a part CAD model according to an XML schema file, and store the linear sizes identifier in an array dim[X]. Define variant i, i∈[0, N-1], where N is the number of dimensions in array dim[X].

Step4: Does the array element dim[i] have two associated objects? If yes, then go to Step5. Otherwise, it implies that there is a missing dimension associated object, and thus it jumps to Step10.

Step5: Array ass[2] is defined. The first initial associated object identifier and the second initial associated object identifier of this dimension can be obtained through an array element dim[i]. Store the two obtained identifiers in the array element ass[0] and array element ass[1].

Step6: Use an array element ass[0] and an array element ass[1] to obtain their associated object, respectively.

Step7: Based on the two types of rules, judge whether the object associated with a linear size is correct or not. The array Error_dim[Y] is defined. When the associated object is an error, then the dimension identifier of it should be stored in the array Error_dim[Y].

Step8: Variant i is greater or equal to N-1? If so, then turn to Step10; otherwise, go to Step9.

Step9: i=i+1, turn to Step5.

Step10: End of the algorithm.

3.4 Conversion of 3D Dimension Associated Objects

The following errors may occur with 3D linear sizes associated objects. (1) 3D linear sizes associated with a point, and the point belongs to a constrained geometric shape feature; (2) 3D linear sizes associated with an edge, and the edge belongs to a constrained geometric shape feature; (3) 3D linear sizes associated to a face, and the face coincides with a surface, but this face does not belong to a constrained geometric shape feature; (4) Other error situations. In the above situations (1), (2), and (3), implied 3D linear sizes annotation clues exist. As long as a correct surface is used to replace a point, a correct surface is used to replace an edge, and a correct surface is used to replace a face, the associated object's conversion can be realized. In situations (4), converting automatically is difficult; hence, a manual conversion is still needed.

How to automatically convert the above situations (1), (2), and (3) will be discussed as follows.

3.4.1 Convert principle

Generally, a constructive solid geometry (CSG) method and a boundary representation method (B-Rep) are used together to represent a geometric shape feature in CAD software. Geometric shape feature data structure is organized into the body, surface, loop, edge, and point [3,4,29]. Using a corresponding API function (Figure 2), when a geometric body is determined, then these surfaces that constitute a body can be acquired; when a surface is determined, then these loops that constitute a surface can be acquired; when a loop is determined, then these edges that constitute a loop can be acquired when an edge is determined then these points that constitute an edge can be acquired.

Using a correct surface to replace an error-associated edge as an example, the conversion process can be described as follows. First, the edge must be determined; second, the set of faces to which the edge belongs must be found; finally, from the set of faces, a surface parallel to the other side of a dimension annotation surface is selected.

3.4.2 Conversion algorithm

The conversion algorithm is described as follows (Figure 3):

Step1: Start the algorithm.

Step2: Import an array element Error_dim[i] from an array Error_dim[Y] obtained in section3.3.

Step3: Acquire an array element Error_dim[i] initial first associated object and initial second associated object one by one. Let the initial first associated object be correctly associated with a geometric feature surface, named Face1; additionally, let the initial second associated object be associated with a geometric feature edge, named Edge2.

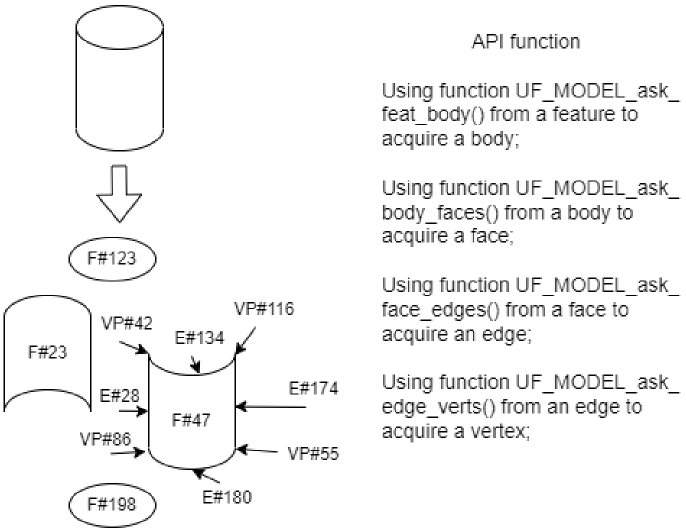


Figure 2: Geometric shape feature data relationship in UG NX.

Step4: Acquire the dimension value corresponding to an array element Error_dim[i] and named Dim12; acquire the face Face1 normal vector and named VFace1.

Step5: Using function UF_MODL_ask_edge_faces( ), acquire the set of faces to which the edge Edge2 belongs, and named array Face2_Set[Z].

Step6: Traversal the array element Face2_Set[j] in the array Face2_Set[Z]. Acquire Face2_Set[j] normal vector and named VFace2_Set[j].

Step7: If VFace2_Set[j] is parallel to VFace1, then mark Face2_Set[j].

Step8: Calculate the distance between Face1 and the marked array element Face2_Set[j], and name the distance L. If L is equal to Dim12, then the marked array element Face2_Set[j] will become a new second associated object of a 3D dimension Error_dim[i] and replace Edge2 with surface Face2_Set[j]; otherwise, go to Step10.

Step9: Set that the associated object of a 3D dimension dim[i] is correct and set the Flag1 value to True.

Step10: End of the algorithm.

Use an appropriate surface to replace both a wrong point of the associated object and a wrong face of the associated object; the process is similar and is not be described here.

4 VERIFICATION

A prototype system was developed based on UG NX1847 API. Considering an example (Figure 4(a)), through this program the 3D dimension annotations information of the part could be extracted automatically. The exacted dimension information included dimension identifier, dimension type, dimension value, allowance above nominal size, allowance below nominal size, the first associated object identifier, first associated object type, second associated object identifier, and second associated object type. Figure 4(b) revealed that when the first or the second associated object type of a linear size was UF_edge_type, an associated object error existed, highlighted in red color in Figure 4(b).

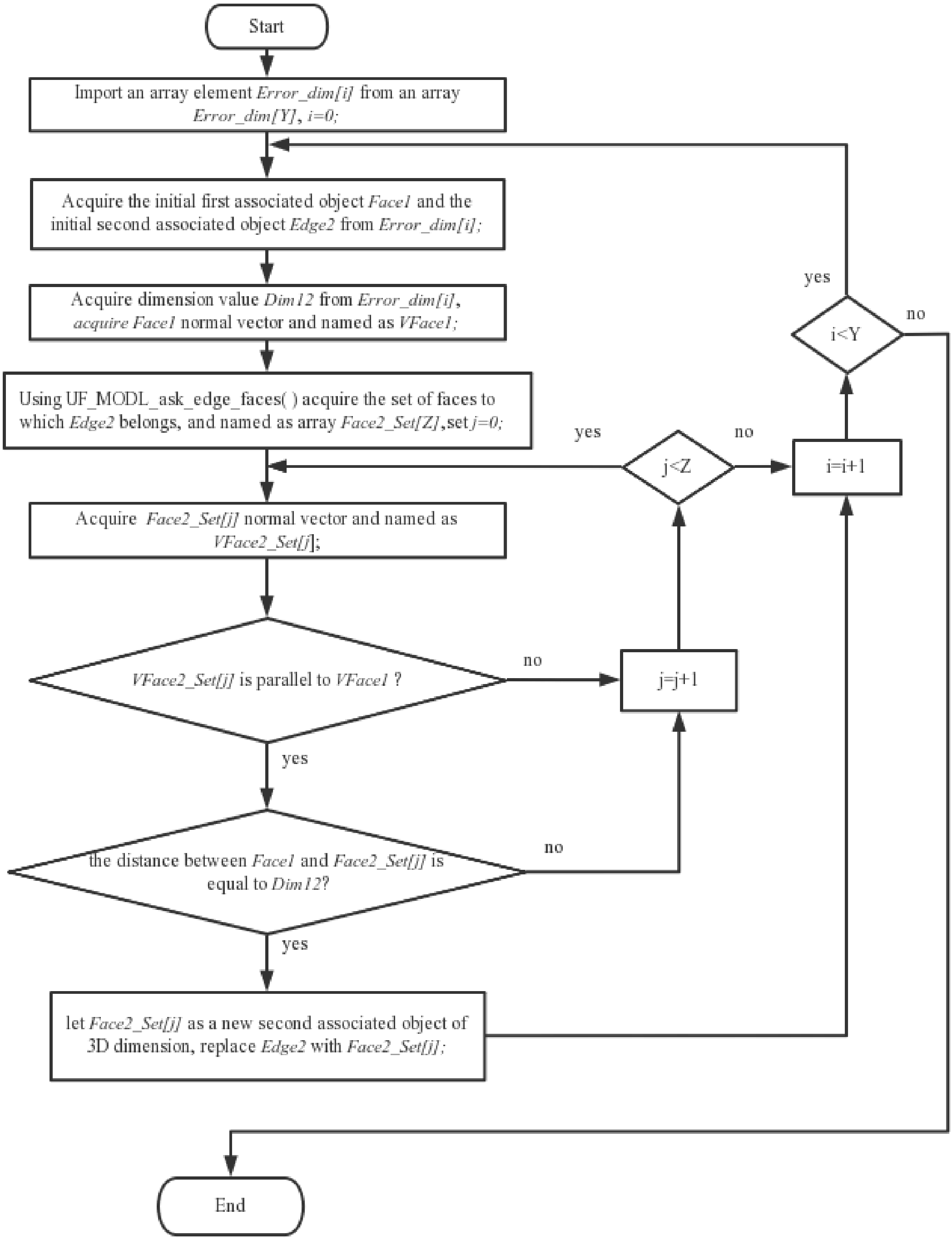
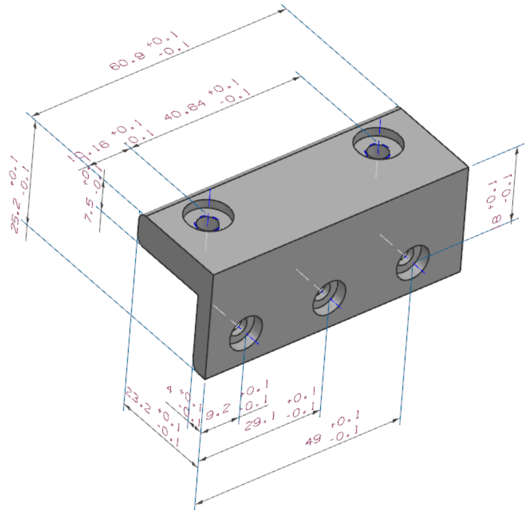
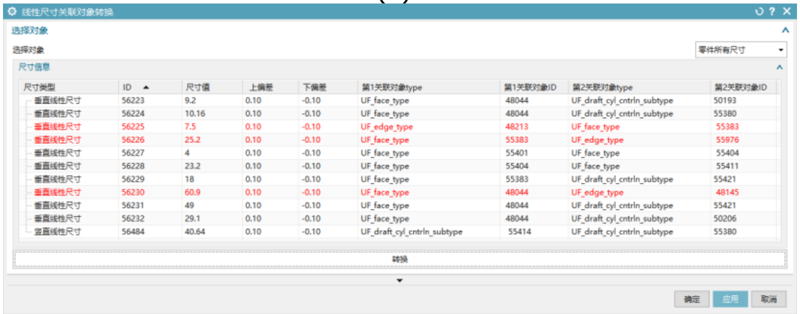


Figure 3: Flow program of a dimension associated object automatic conversion.



(a)



线性尺寸关联对象转换

选择对象

选择对象 零件所有尺寸

尺寸信息

尺寸类型	ID	尺寸值	上偏差	下偏差	第1关联对象type	第1关联对象ID	第2关联对象type	第2关联对象ID
垂直线性尺寸	56223	9.2	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	50193
垂直线性尺寸	56224	10.16	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	55380
垂直线性尺寸	56225	7.5	0.10	-0.10	UF_edge_type	48213	UF_face_type	55383
垂直线性尺寸	56226	25.2	0.10	-0.10	UF_face_type	55383	UF_edge_type	55976
垂直线性尺寸	56227	4	0.10	-0.10	UF_face_type	55401	UF_face_type	55404
垂直线性尺寸	56228	23.2	0.10	-0.10	UF_face_type	55404	UF_face_type	55411
垂直线性尺寸	56229	18	0.10	-0.10	UF_face_type	55383	UF_draft_cyl_cntrln_subtype	55421
垂直线性尺寸	56230	60.9	0.10	-0.10	UF_face_type	48044	UF_edge_type	48145
垂直线性尺寸	56231	49	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	55421
垂直线性尺寸	56232	29.1	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	50206
竖直线性尺寸	56484	40.64	0.10	-0.10	UF_draft_cyl_cntrln_subtype	55414	UF_draft_cyl_cntrln_subtype	55380

转换

确定 应用 取消

(b)

线性尺寸关联对象转换

选择对象

选择对象 零件所有尺寸

尺寸信息

尺寸类型	ID	尺寸值	上偏差	下偏差	第1关联对象type	第1关联对象ID	第2关联对象type	第2关联对象ID
垂直线性尺寸	56223	9.2	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	50193
垂直线性尺寸	56224	10.16	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	55380
垂直线性尺寸	56225	7.5	0.10	-0.10	UF_face_type	52894	UF_face_type	55383
垂直线性尺寸	56226	25.2	0.10	-0.10	UF_face_type	55383	UF_face_type	55390
垂直线性尺寸	56227	4	0.10	-0.10	UF_face_type	55401	UF_face_type	55404
垂直线性尺寸	56228	23.2	0.10	-0.10	UF_face_type	55404	UF_face_type	55411
垂直线性尺寸	56229	18	0.10	-0.10	UF_face_type	55383	UF_draft_cyl_cntrln_subtype	55421
垂直线性尺寸	56230	60.9	0.10	-0.10	UF_face_type	48044	UF_face_type	54541
垂直线性尺寸	56231	49	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	55421
垂直线性尺寸	56232	29.1	0.10	-0.10	UF_face_type	48044	UF_draft_cyl_cntrln_subtype	50206
竖直线性尺寸	56484	40.64	0.10	-0.10	UF_draft_cyl_cntrln_subtype	55414	UF_draft_cyl_cntrln_subtype	55380

转换

确定 应用 取消

(c)

Figure 4: Example verification of: (a) sample part, (b) 3D dimension information before conversion, (c) 3D dimension information after conversion.

A linear size with an error associated object could be automatically converted by this program. The conversion results are shown in Figure 4(c). As mentioned above, owing to the various associated errors in 3D dimension annotations, the program developed in this study cannot automatically convert the associated objects for all the 3D dimension annotations with errors. Essentially, some manual conversion is still required.

5 RESULTS AND DISCUSSION

5.1 Discussion 1

There are numerous possibilities when a part geometric shape feature modeling is finished, and 3D dimension annotations are added. Considering the width dimension annotation D of a part in Figure 5 as an example (given the author's familiarity with UG NX1847, the 3D modeling software used for this part was selected as UG NX1847), the possibilities are as follows. (a) The objects associated with dimension D are Face1 and Face2; (b) those associated with dimension D are Face1 and Edge2; (c) those associated with dimension D are Edge1 and Edge2; (d) those associated with dimension D are Edge1 and Point2; (e) those associated with dimension D are Point1 and Point2; (f) those associated with dimension D are DTM1 and Face2 (DTM1 is a datum plane at the exact position of Face1). When using the above style, such as (b)–(f), for 3D dimension annotations, no distinct visual differences were observed, and designers could understand. However, ambiguities and errors still existed in these 3D annotations because the distance between points cannot be equivalent to the distance between surfaces, the distance between a point and a line cannot be equivalent to the distance between two surfaces, and because it cannot correctly and precisely express a designer's intent [12]. Dimension D in style (f) of Figure 5 is ambiguous because it can represent the distance between Face2 and DTM1 or can position Face2 with respect to DTM1, but the designer intends to use dimension D to define the width of a part.

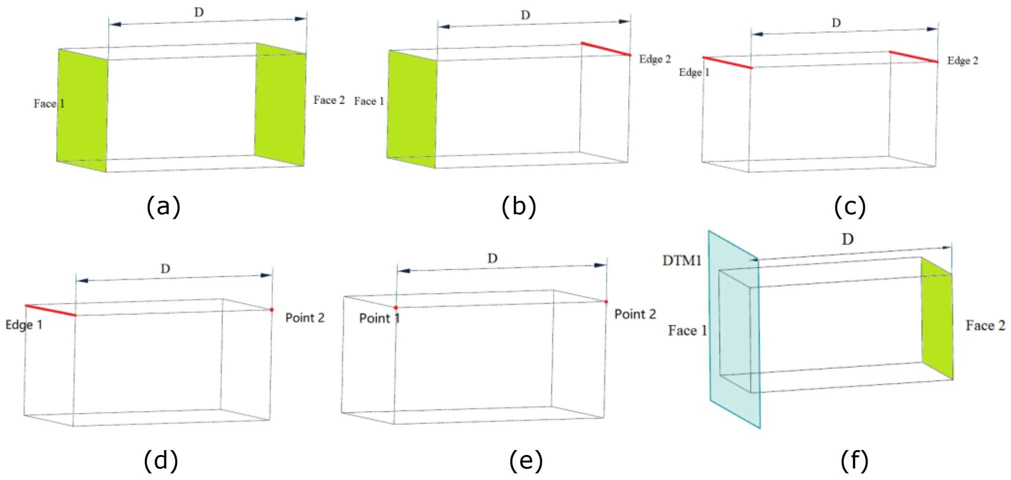


Figure 5: Analysis of associated objects in 3D dimension annotations: (a) two associated objects are surfaces, (b) two associated objects are a surface and an edge, (c) two associated objects are edges, (d) two associated objects are an edge and a point, (e) two associated objects are points, and (f) two associated objects are different faces.

Using the methodology proposed in this study, the 3D dimension annotations having an error or an ambiguity associated with objects, similar to the form of (b)–(f) in Figure 5, can be automatically converted. The definition quality of PMI in an MBD model can be improved. An MBD design model, which has achieved PMI exact representation, can facilitate subsequent CAM/CAPP/CAT/CMM system's efficient utilization [29,30]. For example, an MBD design model in Figure 6 can be easily converted to an MBD manufacturing model by offset Face2 with a process allowance ΔD. The MBD design model, which has not achieved a PMI exact representation, is challenging to be automatically consumed.

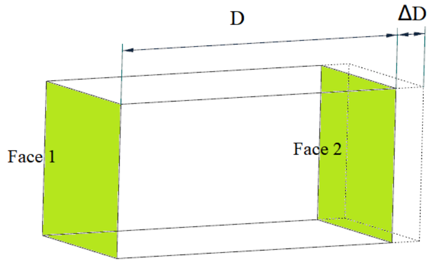
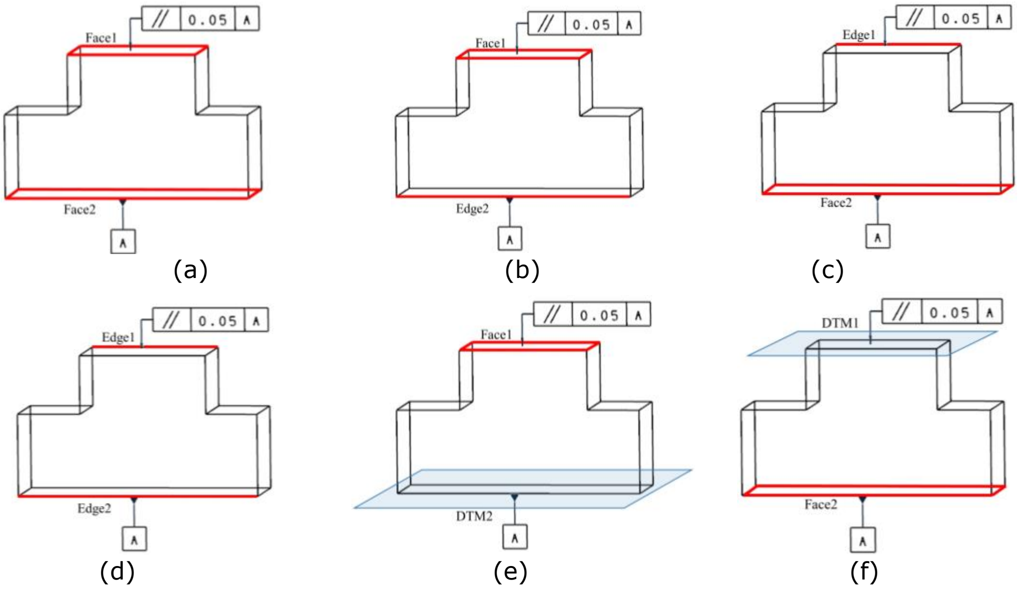
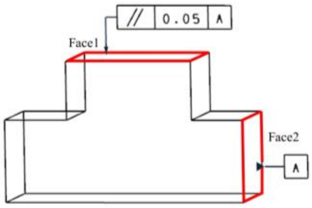


Figure 6: Conversion of an MBD design model.

5.2 Discussion 2

The methodology proposed in this study can be extended to realize exact 3D geometrical tolerance annotations representation. As Figure 7 shows, when a tolerance indicator is used to define a parallelism tolerance related to datum A, the possibilities of a tolerance indicator and datum A are as follows. (a) The object associated with a tolerance indicator is Face1 and the object associated with datum A is Face2; (b) the object associated with a tolerance indicator is Face1 and the object associated with datum A is Edge2; (c) the object associated with a tolerance indicator is Edge1 and the object associated with datum A is Face2; (d) the object associated with a tolerance indicator is Edge1 and the object associated with datum A is Edge2; (e) the object associated with a tolerance indicator is Face1 and the object associated with datum A is DTM2 (DTM2 is a auxiliary datum plane at the exact position of Face2) ; (f) the object associated with a tolerance indicator is DTM1 (DTM1 is a auxiliary datum plane at the exact position of Face1) and the object associated with datum A is Face2; (g) the surface which datum A associated with is not parallel to the surface which a tolerance indicator associated with. When using the above style, such as in Figure 7 (b) - (g), ambiguities and errors existed in these 3D tolerance annotations. Based on the methodology proposed in this study, similar rules and algorithms can be established to realize exact 3D geometrical tolerance annotations representation as in Figure 7 (b)–(f).





(g)

Figure 7: Analysis of associated objects in 3D geometrical tolerances annotations: (a) two associated objects are surfaces, (b) two associated objects are a surface and an edge, (c) two associated objects are an edge and a surface, (d) two associated objects are two edges, (e) two associated objects are a surface and DTM2, (f) two associated objects are DTM1 and a surface, and (g) datum A associated surface is not parallel to the surface which a tolerance indicator is associated with.

5.3 Discussion 3

The methodology proposed in this study is an ex-post check and convert method. Owing to the variety and complexity of 3D annotations, not all types of 3D annotations can be converted by this methodology. A real-time 3D dimension exact representation methodology that can assist a designer when a 3D dimension annotation is simultaneously added must be developed. The new real-time methodology will not reduce a CAD software's flexibility when 3D annotations are added, which is a further research direction.

Owing to the numerous types and contents of PMI, only a 3D dimension exact representation methodology has been discussed in this study. To achieve an exact representation of 3D PMI annotations, the content and format of 3D PMI annotations must be correct to avoid invalid 3D PMI annotations and illegal 3D PMI annotations [2]. For example, when a roundness and cylindricity tolerance are added to the same cylindrical surface, if the roundness tolerance value is greater than the cylindricity tolerance value then it is illegal. Further research will focus on these problems.

6 CONCLUSION

Given that a GD&T and a form feature cannot form a correct semantic association relationship when traditional 3D annotation methodology is adopted, in this study, a 3D annotation information exact representation methodology is proposed which is helpful in improving the definition quality of an MBD model and can facilitate an MBD model automatic processing.

Owing to the numerous types and contents of GD&T in an MBD model, this study focused only on a 3D dimension exact representation. Based on the semantic analysis of 3D dimension annotations, the object types associated with different dimensions were defined. Considering a 3D linear size as an example, possible ambiguities or errors in 3D linear size annotations have been analyzed and the judgment rules were proposed which can be used to judge whether a linear size associated object is correct or not.

Automatic conversion algorithms for 3D linear size associated objects, which can automatically convert an incorrect semantic associated point, edge, or face to a correct one, were established. Based on these rules and algorithms a prototype system was developed. The prototype system running result showed that the rules and algorithms proposed in this paper are valid and efficient.

Similar rules and algorithms can be proposed for other types of GD&T, which can expand the application scope of this methodology. Note that UG NX1847 was selected solely given the author's familiarity with it; no sponsorship was accepted for it. The methodology proposed in this study can be applied to other CAD software similarly.

Owing to the variety and complexity of 3D annotations, not all types of 3D annotations can be automatically converted by the methodology proposed in this study. In addition, the methodology proposed in this study is an ex-post check and convert methodology, so a real-time 3D dimension exact representation methodology that can assist a designer when a 3D dimension annotation is simultaneously added must be developed, and this is a future research direction.

In order to apply the methodology proposed in this study, when a different CAD software platform is adopted a different program needs to be developed through their corresponding API, which means the methodology proposed in this study lacks universality. As heterogeneous CAX systems often use an ISO STEP format file for information exchange, the question of how to realize the 3D annotations information exact representation in a STEP242 format file is another future research direction.

Yazhou Chen, https://orcid.org/0000-0003-0113-6666
Songqing Xu, https://orcid.org/0000-0003-1393-2563
Yi Gan, https://orcid.org/0000-0001-9943-6565
Qihao Zhang, https://orcid.org/0000-0003-3963-6738

ACKNOWLEDGMENTS

The project was supported by the Fujian Provincial Guiding Science and Technology Project (2018H0024) and Xiamen City Science and Technology Project (2022CXY03005).

REFERENCES

[1] Alemanni, M.; Destefanis, F.; Vezzetti, E.: Model-based definition design in the product lifecycle management scenario, International Journal of Advanced Manufacturing Technology, 52, 2011, 1-14. https://doi.org/10.1007/s00170-010-2699-y

[2] ISO 16792:2021, Technical product documentation - Digital product definition data practices, International Organization for Standardization, Geneva, Switzerland, 2021.

[3] Cheng, Z.; Ma, Y.: A functional feature modeling method. Advanced Engineering Informatics, 33, 2017, 1-15. http://dx.doi.org/10.1016/j.aei.2017.04.003

[4] Hallmann, M.; Goetz, S.; Schleich, B.: Mapping of GD&T information and PMI between 3D product models in the STEP and STL format, Computer-Aided Design,115,2019,293-306.https://doi.org/10.1016/j.cad.2019.06.006

[5] Srivastava, D.; Komma, V. R.: Development of STEP AP224 extractor for interfacing feature based CAPP to STEP-NC (AP238), International Journal of Automation and 1Computing, 6(5), 2019, 655-670.https://doi.org/10.1007/s11633-018-1144-9

[6] Liu, X.; Li, X.; Xing, J.; Chen, H.; Ni, Z.: Integrating modeling mechanism for three-dimensional casting process model based on MBD, International Journal of Advanced Manufacturing Technology, 94, 2018, 3145-3162.https://doi.org/10.1007/s00170-016-9479-2

[7] Kwon, S.; Monnier, L. V.; Barbau, R.; Bernstein, W. Z.: Enriching standards-based digital thread by fusing as-designed and as inspected data using knowledge graphs, Advanced Engineering Informatics, 46(10), 2020, 101102.https://doi.org/10.1016/j.aei.2020.101102

[8] Yuchu, Q.; Wenlong, L.; Qunfeen, Q.; Li, T.; Huang, M.; Scott, P. J.; Jiang, X.: Explicitly representing the semantics of composite positional tolerance for patterns of holes,International Journal of Advanced Manufacturing Technology, 90, 2017, 2121-2137. https://doi.org/10.1007/s00170-016-9457-8

[9] Amadori, K.; Tarkian, M.; Olvander, J.; Krus, P.: Flexible and robust CAD models for design automation, Advanced Engineering Informatics, 26, 2012, 180-195. https://doi.org/10.1016/j.aei.2012.01.004

[10] Company, P.; Contero, M.; Otey, J.; Plumed, R.: Approach for developing coordinated rubrics to convey quality criteria in MCAD training, Computer-Aided Design, 63, 2015, 101-117. https://doi.org/10.1016/j.cad.2014.10.001

[11] Otey, J.; Company, P.; Contero, M.; Camba, J. D.: Assessment of parametric assembly models based on CAD quality dimensions, Computer-Aided Design& Applications, 16(4), 2019, 628-653. https://doi.org/10.14733/cadaps.2019.628-653

[12] Company, P.; Naya, F.; Contero, M.; Camba, J. D.: On the role of geometric constraints to support design intentcommunication and model reusability, Computer-Aided Design& Applications, 17(1), 2020, 61-76. https://doi.org/10.14733/cadaps.2020.61-76

[13] González-Lluch, C.; Company, P.; Contero, M.; Camba, J. D.; Plumed, R.: A survey on 3D CAD model quality assurance and testing tools, Computer-Aided Design, 83, 2017, 64-79. http://dx.doi.org/10.1016/j.cad.2016.10.003

[14] Ruemler, S. P.; Zimmerman, K. E.; Hartman, N. W.; Hedberg, T. Jr.; Barnard Feeny, A.: Promoting Model-Based definition to establish a complete product definition, Journal of Manufacturing Science and Engineering, Transactions of the ASME, 139(5), 2017, 51-58. http://dx.doi.org/10.1115/1.4034625

[15] Cax-If Recommended Practices for the Representation and Presentation of Product Manufacturing Information(PMI) (AP242). https://www.cax-if.org/documents/rec_pracs_pmi_v40.pdf.

[16] Xiao, J.; Anewer, N.; Durupt, A.; Duigou, L. J.; Eynard, B.: Information exchange standards for design, tolerancing and Additive Manufacturing: a research review, International Journal on Interactive Design and Manufacturing, 12, 2018, 495-504. https://doi.org/10.1007/s12008-017-0401-4

[17] Yang, W.; Zhao, Q.; Yan, X.; Chen, Z.: A system framework of model quality analysis for product model in collaborative manufacturing, International Journal of Advanced Manufacturing Technology, 117, 2021, 1351–1374. https://doi.org/10.1007/s00170-021-07622-1

[18] Lipman, R.; Lubell, J.: Conformance checking of PMI representation in CAD model STEP data, Computer Aided Design, 66, 2015, 14-23. http://dx.doi.org/10.1016/j.cad.2015.04.002

[19] Lipman, R.; Filliben, J.: Testing implementations of geometric dimensioning and tolerancing in CAD Software, Computer-Aided Design & Applications, 17(6), 2020, 1241-1265.https://doi.org/10.14733/cadaps.2020.1241-1265

[20] Gao, J.; Zheng, D. T.; Gindy, N.; Clark, D.: Extraction/conversion of geometric dimensions and tolerances for machining features, International Journal of Advanced Manufacturing Technology, 26, 2005, 405-414. https://doi.org/10.1007/s00170-004-2195-3

[21] Agovic, A.; Trautner, T.; Bleicher, F.: Digital transformation-implementation of drawingless manufacturing: A Case Study, Procedia CIRP, 107, 2022, 1479-1484. https://doi.org/10.1016/j.procir.2022.05.178

[22] ISO 14405-1:2016, Geometrical product specifications (GPS) - Dimensional tolerancing - Part 1: Linear sizes, International Organization for Standardization, Geneva, Switzerland, 2016.

[23] ISO 14405-2:2018, Geometrical product specifications (GPS) - Dimensional tolerancing - Part 2: Dimensions other than linear or angular sizes, International Organization for Standardization, Geneva, Switzerland, 2018.

[24] ISO 14405-3:2016, Geometrical product specifications (GPS) - Dimensional tolerancing - Part 3: Angular sizes, International Organization for Standardization, Geneva, Switzerland, 2016.

[25] ISO 1101:2017, Geometrical product specifications (GPS) - Geometrical tolerancing - Tolerances of form, orientation, location and run-out, International Organization for Standardization, Geneva, Switzerland, 2017.

[26] Stefano Tornincasa, Technical Drawing for Product Design - Mastering ISO GPS and ASME GD&T, Springer Tracts in Mechanical Engineering, 2021. https://doi.org/10.1007/978-3-030-60854-5

[27] Sarigecili, M. I.; Roy, U.; Rachuri, S.: Interpreting the semantics of GD&T specifications of a product for tolerance analysis, Computer-Aided Design, 47, 2014, 72-84. http://dx.doi.org/10.1016/j.cad.2013.09.002

[28] Nzetchou, S.; Durupt, A.; Remy, S.; Eynard, B.: Semantic enrichment approach for low-level CAD models managed in PLM context: Literature review and research prospect, Computer in Industry, 135, 2022, 103575. https://doi.org/10.1016/j.compind.2021.103575

[29] Zhi, Li.; Xiaowu, Z.; Wang, W. M.; Huang, G.; Tian, Z.; Huang, S.: An ontology-based product design framework for manufacturability verification and knowledge reuse, International Journal

of Advanced Manufacturing Technology, 99, 2018, 2121-2135.https://doi.org/10.1007/s00170-018-2099-2
[30] Peng, Z.; Huang, M.; Zhong, Y.; Chen, L.; Liu, G.: A new method for interoperability and conformance checking of product manufacturing information, Computers & Electrical Engineering, 85, 2020, 106650. https://doi.org/10.1016/j.compeleceng.2020.106650