

Towards an ontology-supported case-based reasoning approach for computer-aided tolerance specification

Yuchu Qin ^a, Wenlong Lu ^{a,*}, Qunfen Qi ^b, Xiaojun Liu ^a, Meifa Huang ^c, Paul J Scott ^b, Xiangqian Jiang ^b

^a *The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, PR China*

^b *EPSRC Centre for Innovative Manufacturing in Advanced Metrology, University of Huddersfield, Huddersfield, HD1 3DH, UK*

^c *School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, PR China*

Abstract: In this paper, an ontology-supported case-based reasoning approach for computer-aided tolerance specification is proposed. This approach firstly considers the past tolerance specification problems and their schemes as previous cases and the new tolerance specification problems as target cases and uses an ontology to represent previous and target cases. Then certain ontology-based similarity measure is used to assess the similarity between the tolerated features of target and previous cases, the similarity between the part features of target and previous cases, and the similarity between the topological relations of target and previous cases. Based on these similarities, an ontology-based similarity measure for computing the similarity between target and previous cases is designed, and an algorithm for establishing such similarity measure with high accuracy and retrieving similar previous cases for a target case with this similarity measure is presented. This algorithm shows how to linearly combine the similarity of tolerated features, the similarity of part features, and the similarity of topological relations to assess the similarity between target and previous cases to implement retrieval of previous cases under the prerequisite of ensuring the highest accuracy of the similarity measure. The paper also reports a prototype implementation of the proposed approach, provides an example to illustrate how the approach works, and evaluates the approach via theoretical and experimental comparisons.

Keywords: Computer-aided tolerance specification; Tolerance specification scheme; Tolerance specification problem; Case-based reasoning; Ontology; Similarity measure

1. Introduction

Tolerance in mechanical product development is defined as the permissible variation limit of the dimensions and geometric parameters of parts. It is the concrete reflection of product precision, the outcome of the coordination of product requirements and development costs, and an important technical indicator in the processes of product design, manufacturing, verification, assembly, and testing. Tolerance design (or tolerancing) is the design of the dimensions and tolerances of parts in the process of product development. Its purpose is to develop products which can satisfy functional and assembly requirements with minimum development costs. Since the advent of the theory of geometric dimensioning and tolerancing (GD&T), tolerance design has been broken down into three major parts that of tolerance specification, that of tolerance allocation, and that of tolerance analysis, where:

- Tolerance specification mainly considers identifying tolerated features from each part of a product, establishing a datum reference frame for each part, selecting tolerance types for each tolerated feature, and using tolerance principles under the prerequisite of satisfying the functional and assembly requirements of the product.
- Tolerance allocation focuses on determining the values of all selected tolerance types through refining the empirical values or optimizing the empirical values based on the functional and assembly requirements of the product and specific cost-tolerance functions.

* Corresponding author. *Tel.*: +86-027-87557835
E-mail address: hustwenlong@mail.hust.edu.cn (W. Lu).

- Tolerance analysis put emphasis on evaluating the accumulation of errors that are allowed by the determined tolerance values and comparing such accumulation to the functional and assembly requirements of the product.

So far, semi-automatic or automatic tolerance allocation and analysis have become a reality in a number of computer-aided design (CAD) systems, while tolerance specification is still manually carried out by mechanical designers according to specific tolerancing standards (e.g. ISO 1101 [1] and ASME Y14.5 [2]) in CAD systems [3]. In such situation, the quality of the designed tolerance specification schemes is largely depended on the judgment and experience of the designers. Different designers may opt for different tolerance specification schemes for an identical nominal geometry under the same circumstance. This may not make much difference when designing a simple product. But for a complex product (e.g. automobile, manufacturing machine, airplane), this situation will directly increase the uncertainty in product tolerance design and eventually have a negative effect on the quality of the product [4]. Besides, tolerance specification information is represented as the symbol annotations of CAD models in most of the current CAD systems. Although using such representation form is capable to establish a syntactically correct representation model of tolerance specification information, it cannot represent the semantics of tolerance specification information explicitly [5]. For this reason, tolerance specification information is not really computer-interpretable in most of the current CAD systems, which impedes the implementation of automatic tolerance specification in these systems.

To make tolerance specification information really computer-interpretable in CAD systems and to reduce the uncertainty of product tolerance design, the technologies of ontology and case-based reasoning (CBR) in the area of artificial intelligence are simultaneously introduced to computer-aided tolerance specification and an ontology-supported CBR approach for computer-aided tolerance specification is proposed in this paper. CBR [6], a powerful technology for computer reasoning and problem-solving, is well-known for providing an effective mechanism for solving new problems based on the solutions of similar past problems. It has been formalized with the purposes of computer reasoning and problem solving as a process consisting of four steps: case retrieval, reuse, revision, and retention, where case retrieval is the basis of other three steps [7]. To implement case retrieval, the first thing to do is to represent the cases. The proposed approach uses web ontology language (OWL) ontology [8], which is well-known for having a capability to represent information semantics explicitly and make information computer-interpretable, to represent the past tolerance specification problems and their schemes and considered them as previous cases and to represent each new tolerance specification problem and considered it as a target case. The represented cases have rigorous computer-interpretable semantics because of the mathematical logic-based semantics of OWL [9]. As the benefits of such representations, semantic similarity assessment can be performed on the OWL ontology, which provides an effective mechanism to implement case retrieval [10]. Based on the OWL ontology representations of previous and target cases, an ontology-based similarity measure with high accuracy is then designed to automatically retrieve similar previous cases for a target case. The tolerance specification schemes in some of these similar previous cases may provide auxiliary information in solving the new tolerance specification problem in the target case. By reusing or reusing and revising these schemes, designers can carry out tolerance specification for the new problem more easily.

The remainder of the paper is organized as follows. An overview of related work is provided in Section 2. The details of the proposed approach are explained in Section 3. Section 4 reports a prototype implementation of the approach, presents an example to illustrate how the approach works, and evaluates the approach through theoretical and experimental comparisons. Section 5 ends the paper with a conclusion.

2. Related work

Computer-aided tolerance specification is the application of computer technologies to aid in tolerance specification of mechanical products. It is preferably implemented in accordance with tolerancing standards (e.g. ISO 1101 [1], ASME Y14.5 [2]). But existing tolerancing standards do not provide a practical method to implement computer-aided tolerance specification. In current practice, tolerance specification work is manually completed by mechanical designers in CAD systems. This situation, as analyzed in the introduction, will directly increase the uncertainty in the tolerance design of a complex product and eventually have a negative effect on the quality of the product [4].

With an aim of reducing the uncertainty in product tolerance design, a number of approaches for computer-aided tolerance specification have been presented during the past two decades [11]. These approaches can be classified into the following three categories on the basis of their existing problems:

- Some step in the approach needs a great deal of know-how to complete human-computer interaction. Representative examples for this category are technologically and topologically related surface (TTRS) approach [12–15], degree of freedom (DOF) approach [16–18], and requirement decomposition approach [19–21]. The TTRS approach, which was presented by Clement et al. [12], implemented by Salomons et al. [13], and extended by Toulorge et al. [14] and Zhang et al. [15], is the first and most cited approach for computer-aided tolerance specification. This approach firstly extracts the surfaces of each part from a geometric modeler, then models each surface using an invariance class [22] and a minimum set of reference points, lines, and planes (which are called minimum geometric datum elements), and finally generates the applicable tolerance specification schemes for each surface based on the classified tolerancing cases from the attributes of the model of this surface. A significant characteristic of the TTRS approach is that it is intuitive and unambiguous. But the selection of the minimum geometric datum elements in it requires a great deal of know-how, which could lead to human errors. The DOF approach, which was proposed by Shah et al. [16], Kandikjan et al. [17], and Wu et al. [18], also works under a priori classification of tolerancing cases. In this approach, a metric relation links two atomic geometric features. Such relation must be controlled by dimension or geometric tolerances. For each link of geometric features, the priori classification firstly provides the type, number, and direction of the constrained DOFs. Then certain rules are defined to establish the mappings from the constraints on DOF, toleranced features, and constrained metric relations to tolerance types and to guide the designers to choose datums and basic dimensions. The main advantage of the DOF approach is that it has rigorous mathematical definition (its mathematical basis is rigid body kinematics). But like the TTRS approach, the input of information of the constrained DOFs also requires a great deal of know-how. The requirement decomposition approach, which was presented by Ballu and Mathieu [19] and extended by Dantan et al. [20] and Costadoat et al. [21], firstly leverages an iterative procedure to decompose all functional requirements input by the designers, then generates the influenced translations and rotations for each geometric feature according to the decomposed functional requirements, and finally selects the tolerance types and related datum for this feature on the basis of the generated influenced translations and rotations. The most prominent characteristic of this approach is that it has developed a second-order GD&T schema, while most of the existing approaches only implement first-order tolerance specification [23]. However, the approach needs explicit functional and assembly requirements as its input, which largely increases the difficulty of its use.

- The approach is incomplete for tolerance specification. Examples for this category are mirror approach [24, 25, 3] and semantic web rule language (SWRL) rules based approach [26]. The mirror approach, which was presented by Wang et al. [24] and extended by Armillotta and Semeraro [25, 3], firstly defines the planar surfaces having assembly relations with other planar surfaces as mirrors and divides mirrors into strong and weak mirrors. Strong mirrors are then selected as primary datums and weak mirrors are chosen as secondary or tertiary datums. Finally, form, location, and orientation tolerances are specified to datums and features using a rule-based procedure. The main characteristic of this approach is that it is easy to use. But for those parts without planar surface, the approach needs to be further improved. The SWRL rules based approach, which was proposed by Zhong et al. [26], aims to generate tolerance types in CAD systems automatically. This approach firstly use OWL assertions to formalize the relations in the part layer, assembly feature surface layer, and spatial relation layer of the spatial relations based assembly tolerance representation model [27]. It then uses SWRL rules to formalize the mapping relations between spatial relations and tolerance types. Finally, for an arbitrary assembly, the approach extracts its assembly requirements from the system, uses the extracted assembly requirements to instantiate the OWL assertions, and infers the recommended tolerance types for each part of the assembly through the instantiated OWL assertions and SWRL rules. Due to the intelligent inference mechanism of SWRL rules and the semantic representation mechanisms of OWL ontology, the SWRL rules based approach has advantages in having a high degree of intelligence and satisfying the requirement of representing the semantics of tolerance type information. However, the approach has not yet involved automatic generation of datums, toleranced features, and tolerance principles.
- Some step in the approach is extremely complicated. Such situation can be seen in assembly positioning constraints based approach [28–32], variational geometric constraint network (VGCN) approach [33, 34], and polychromatic sets based approach [35, 4]. The assembly positioning constraints based approach, which was presented by Anselmetti and Mawussi [28], extended by Mejbri et al. [29] and Cao et al. [32], and implemented by Anselmetti [30, 31], defines each functional requirement as a geometric tolerance on an ending geometric entity. This tolerance is decomposed into a set of geometric tolerances on individual parts according to a user-established positioning table that lists the assembly positioning constraints in order. The datum reference frame of each obtained geometric tolerance is then validated by rules that are based on the degrees of invariance of the features of individual parts. Like the requirement decomposition approach, this approach has also developed a second-order GD&T schema. But for a complex assembly, the establishment of its positioning table is extremely complicated. The VGCN approach, which was proposed by Hu et al. [33] and implemented by Hu and Peng [34], also works under a priori classification of tolerancing cases. But unlike the TTRS and DOF approaches, this approach needs explicit assembly requirements as its input. For each requirement, the approach links the involved nominal derived features using topological relations and parameters. The obtained graph is called as VGCN. Each link in a variational geometric constraint network can be assigned a geometric tolerance type according to its topological relation and the priori classification of tolerancing cases. For a simple assembly, the VGCN approach is easy to use. But for a complex assembly, it is very difficult to use because the construction of a graph-based VGCN is extremely complicated. The polychromatic sets based approach, which was presented and extended by Zhang et al. [35, 4], can be seen as an extension of the VGCN approach. This approach firstly uses some contour matrices in polychromatic sets to represent variational geometric constraints and tolerance

types. Then according to the inference relations between unified and individual colors of polychromatic sets, it can infer variational geometric constraints and tolerance types for each explicit assembly requirement. Finally, the approach constructs a tolerance network on the basis of the inferred variational geometric constraints and tolerance types. Like the issue in the VGCN approach, the construction of a tolerance network for a complex assembly in the polychromatic sets based approach is extremely complicated.

As can be seen from the above review, the existing approaches for computer-aided tolerance specification are too technical, not complete enough, or extremely complicated. How to implement effective computer-aided tolerance specification remains a challenging task.

To tackle such challenge, an ontology-supported CBR approach for computer-aided tolerance specification is proposed in the present paper. The proposed approach firstly uses OWL ontology to represent the past tolerance specification problems and their schemes as previous cases and to represent each new tolerance specification problem as a target case. It then automatically generates reference tolerance specification schemes for each new tolerance specification problem using a retrieval technique driven by ontology-based semantic similarity assessment. By reusing or reusing and revising these schemes, designers can easily carry out tolerance specification for the new problem. After that, designers can easily retain the revised tolerance specification scheme as a new previous case. Compared with the three categories of approaches above, the proposed approach has the following characteristics: (1) It directly takes the CAD model of an assembly as input and returns reference tolerance specification schemes for each tolerated feature of each part of the assembly as output. This process is fully automatic and does not need know-how. (2) The proposed approach is applicable for all shapes of parts. It has involved the generation of a complete tolerance specification scheme (i.e. tolerated feature, datum, tolerance type, and tolerance principle). (3) The proposed approach needs human-computer interaction in the steps of reuse, revision, and retention of the generated reference tolerance specification schemes. Such interaction only requires a small amount of know-how and is not complicated.

3. Ontology-supported CBR approach

Carrying out tolerance specification for an assembly is designing tolerance specification schemes for each part of this assembly. This includes identifying tolerated features from each part of the assembly, establishing a datum reference frame for each part, selecting tolerance types for each tolerated feature, and using tolerance principles under the prerequisite of satisfying the functional and assembly requirements of the assembly. As an example, the process of designing tolerance specification scheme for a shaft in Fig. 1 is as follow:

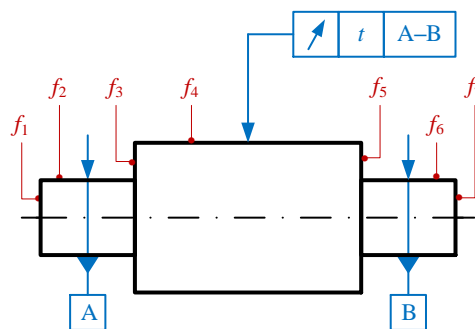


Fig. 1. An example of designing tolerance specification scheme.

- Identifying the tolerated feature: The shaft consists of seven features: planar surface f_1 ; cylindrical surface f_2 ; planar surface f_3 ; cylindrical surface f_4 ; planar surface f_5 ; cylindrical surface f_6 ; planar surface f_7 . Its main function is rotation. The run-out of the feature f_4 plays an important role in ensuring this function. Thus f_4 is selected as a feature to be tolerated.
- Establishing the datum reference frame: In general, the common axis of two cylindrical surfaces in a shaft can be taken as the datum reference frame of this shaft. The datum reference frame of the shaft in Fig. 1 is naturally established as the common axis of the cylindrical surfaces f_2 and f_6 (i.e. A–B).
- Selecting the tolerance type for each tolerated feature: The tolerated feature f_4 is a cylindrical surface, whose applicable tolerance types are straightness, roundness, cylindricity, circular run-out, and total run-out [18]. The topological relation of f_4 with respect to the datum reference frame A–B in ideal situation is parallel, whose applicable tolerance types are linear dimension tolerance, parallelism, position, symmetry, circular run-out, and total run-out [27]. Considering these two groups of applicable tolerance types and the functional requirement of the shaft synthetically, circular run-out is selected for f_4 .
- Using tolerance principles: The circular run-out tolerance cannot be used any tolerance principles since it is not specified on a feature of size [1, 2].

As can be seen from the example above, the basic elements for describing a tolerance specification problem are a tolerated feature (f_4), the part containing the tolerated feature (the shaft in Fig. 1) and all the features of this part ($f_1, f_2, f_3, f_4, f_5, f_6,$ and f_7), and the topological relation of the tolerated feature with respect to the datum reference frame of the part in an ideal situation (parallel), while the basic elements for describing a solution of a tolerance specification problem are a datum reference frame (if required) (A–B), tolerance type (circular run-out), and tolerance principle (if required). According to the tolerancing standard ISO 1101 [1] or ASME Y14.5 [2], a datum reference frame may consist of only one datum, two datums, or three datums. When it consists of two datums, the two datums are respectively called primary datum and secondary datum. When it consists of three datums, the three datums are respectively called primary datum, secondary datum, and tertiary datum. Tolerance principles can be applied on the whole tolerance, the primary datum, the secondary datum, and the tertiary datum. They are called general tolerance principle, primary datum tolerance principle, secondary datum tolerance principle, and tertiary datum tolerance principle, respectively.

Formally, let D_P be the primary datum, D_S be the secondary datum, D_T be the tertiary datum, T_T be the tolerance type, P_G be the general tolerance principle, P_P be the primary datum tolerance principle, P_S be the secondary datum tolerance principle, P_T be the tertiary datum tolerance principle, F_T be the tolerated feature, P be the part that contains the tolerated feature, F_P be a set of all the features of part P , and R_T be the topological relation of the tolerated feature with respect to the datum reference frame in an ideal situation. Then the basic elements for describing a tolerance specification problem are $D_P, D_S, D_T, F_T, P, F_P,$ and R_T , while the basic elements for describing a solution of a tolerance specification problem are $D_P, D_S, D_T, T_T, P_G, P_P, P_S, P_T, F_T, P, F_P,$ and R_T . From the perspective of CBR, $D_P, D_S, D_T, F_T, P, F_P,$ and R_T can constitute a target tolerance specification case, $D_P, D_S, D_T, T_T, P_G, P_P, P_S, P_T, F_T, P, F_P,$ and R_T can constitute a previous tolerance specification case, and the proposed ontology-supported CBR approach for computer-aided tolerance specification can be seen as an approach for retrieving similar previous cases for each target case using an ontology-based similarity assessment technique.

This section aims to describe the details of the proposed ontology-supported CBR approach. The schematic representation of the approach is shown in Fig. 2. As can be seen from this figure, the approach mainly comprises six

steps. The first step is to represent the previous tolerance specification cases using ontology. The part information (i.e. datum, features, tolerated features, and topological relations between features) of each new tolerated feature is extracted from a CAD system to construct a target tolerance specification case in the second step. The third step is to assess the similarities between tolerated features, between part features, and between topological relations. Then the similarity between a target case and a previous case can be defined as a weighted sum of these three similarities. How to determine a similarity measure with high accuracy to assess this similarity is also explained in this step. The fourth step is to use the determined similarity measure to compute the similarity between a target case and each of all previous cases in the case base. Through these four steps, some similar previous cases can be generated. The tolerance specification schemes in these previous cases can be combined with the recommended tolerance types generated by the SWRL rules based approach in [26] to assist tolerance specification in the target case through a case reuse mechanism or case reuse and revision mechanisms in the fifth step. Then the target case and its final tolerance specification schemes could be retained as a new previous case by a case retention mechanism in the last step. Because part information extraction can be easily implemented using the application program interfaces (APIs) of a CAD system and tolerance specification case retention is manually carried out by a designer, the section will not explain the second step and the last step in detail. The details of the first step, the third and fourth steps, and the fifth step are respectively explained in the following three sub-sections: the representation of tolerance specification cases, the retrieval of tolerance specification cases, and the reuse and revision of tolerance specification cases.

3.1. Representation of tolerance specification cases

3.1.1. Tolerance specification cases

A case can be seen as a data structure that consists of a description and a solution of a problem. In CBR [6], there are generally two types of cases: previous case and target case, where a previous case is a historical case that consists of a description and a solution of a problem, and a target case is a newly encountered case that only contains a description of a problem. For a tolerance specification problem, F_T , P , F_P , and R_T constitute its description, and D_P , D_S , D_T , T_T , P_G , P_P , P_S , and P_T constitute its solution (such solution is usually called a tolerance specification scheme in GD&T). Naturally, a tolerance specification case, a previous case for tolerance specification, and a target case for tolerance specification can be respectively defined as follows:

Definition 1 (Tolerance specification case). A tolerance specification case is a fifteen-tuple $C_{TS} = (N, D, A, D_P, D_S, D_T, T_T, P_G, P_P, P_S, P_T, F_T, P, F_P, R_T)$, where N is the case number, D is the last revised date of the case, and A is the author of the case.

Definition 2 (Previous case for tolerance specification). A previous case for tolerance specification is a fifteen-tuple $C_P = (N, D, A, D_P, D_S, D_T, T_T, P_G, P_P, P_S, P_T, F_T, P, F_P, R_T)$, where all tuples have the same meanings as they have in Definition 1.

Definition 3 (Target case for tolerance specification). A target case for tolerance specification is a seven-tuple $C_T = (D_P, D_S, D_T, F_T, P, F_P, R_T)$, where all tuples have the same meanings as they have in Definition 1.

Fig. 3(a) and Fig. 3(b) respectively show an example of previous case and an example of target case. Assume the number of the previous case is “PC000001”, the last revised date of the previous case is “OCT 06, 2016”, and the author of the previous case is “Yuchu Qin”. Then according to Definitions 2 and 3, the previous case in Fig. 3(a) and the target case in Fig. 3(b) can be respectively described as follows:

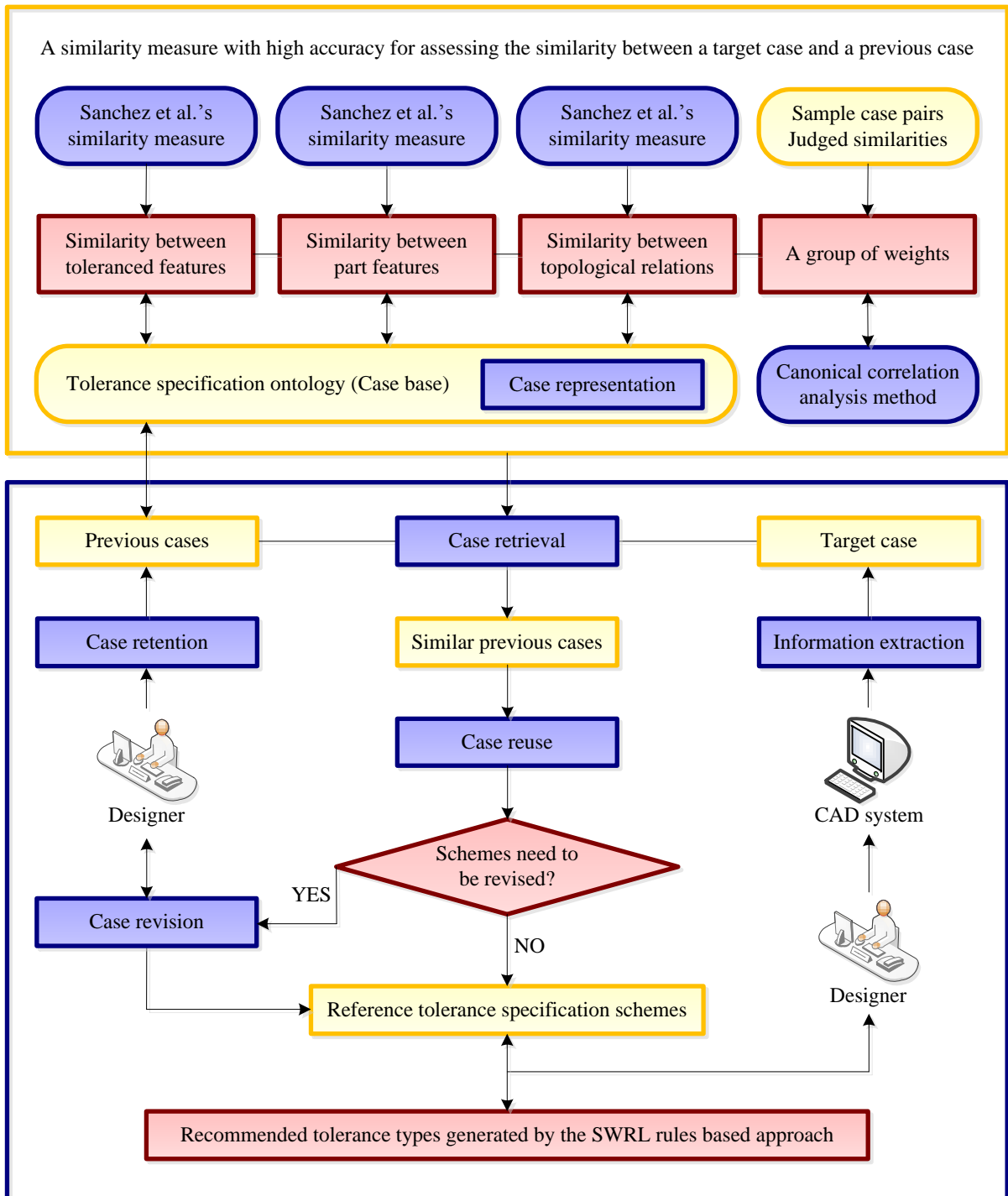


Fig. 2. Schematic representation of the ontology-supported CBR approach for computer-aided tolerance specification.

$C_S = ("PC000001", "OCT 06, 2016", "Yuchu Qin", "A-B", "", "", "Circular run-out", "", "", "", "f_{P,4}, P_P, \{f_{P,1}, f_{P,2}, f_{P,3}, f_{P,4}, f_{P,5}, f_{P,6}, f_{P,7}\}, "Parallel")$
 $C_T = ("A", "", "", f_{T,5}, P_T, \{f_{T,1}, f_{T,2}, f_{T,3}, f_{T,4}, f_{T,5}\}, "Perpendicular")$

3.1.2. Tolerance specification ontology

In artificial intelligence, ontology has been given a meaning: an explicit and formalized specification of a shared conceptualization model, where “explicit” means all concepts in the model and all of the constraints between them

have unambiguous definitions, “formalized” means the representation of the model has a rigorous mathematical basis and is directly computer-readable and computer-interpretable, “shared” means the described objects in the model are some commonly recognized concepts in a field so that the model can provide common interpretations of these concepts, and “conceptualization” means that the model is an abstraction of the objective things [8]. This meaning can be seen as a non-formalized definition of ontology. In practical applications of ontology, a formalized definition is often required. Without loss of generality, a formalized definition of tolerance specification ontology is given as follow:

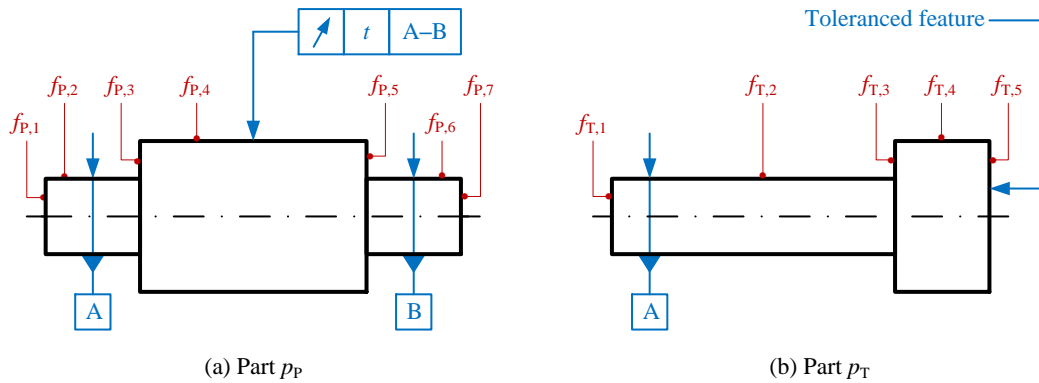


Fig. 3. An example of previous case and an example of target case.

Definition 4 (Tolerance specification ontology). The tolerance specification ontology divides the world of tolerance specification into four parts of classes, properties, individuals, and axioms, where an individual is a concrete instance in the world of tolerance specification, a property stands for a relationship between two individuals, a class is a unified name of a group of individuals that have a common property, and an axiom is a fact that holds in the world of tolerance specification. Based on these parts, the tolerance specification ontology is defined as a four-tuple $\mathcal{O}_{TS} = \{S_C, S_P, S_I, S_A\}$, where S_C is a set of classes, S_P is a set of properties, S_I is a set of individuals, and S_A is a set of axioms.

After formally defining the tolerance specification ontology, the next step is to construct this ontology. For ontology construction, there are various available methods, such as skeleton method, TOVE method, SENSUS method, METHONTOLOGY method, and seven-step method [36]. Among these methods, the most cited one is the seven-step method [37], which was presented by Stanford University, one of the most authoritative ontology research institutions in the world. This method has very detailed documentation that can be used as a reference for the construction of an ontology. In addition, an ontology constructed by the method can be easily extended and reused to multiple domains. For these reasons, the seven-step method is applied to manually construct the tolerance specification ontology. The construction process mainly consists of the following seven steps:

- Determine the domain and scope of the ontology. The tolerance specification ontology is mainly used in computer-aided tolerance specification.
- Consider reusing existing ontologies. In the field of tolerance design, some researchers like Fiorentini et al. [38], Zhong et al. [26], Lu et al. [39], Ahmed and Han [40], and Qin et al. [41] have constructed their respective ontologies that are related to tolerance for their specific purposes, where Fiorentini et al. [38] constructed an ontology for assembly tolerance representation, Zhong et al. [26] constructed an ontology for generation of assembly tolerance types, Lu et al. [39] constructed an ontology for the enrichment of the semantics of variational geometric constraint information, Ahmed and Han [40] constructed an ontology for tolerance information exchange, and

Qin et al. [41] constructed an ontology for generation of geometric tolerance zones. Among these ontologies, the ontology for generation of assembly tolerance types in [26] has a common purpose with the tolerance specification ontology in this paper. Further, the proposed ontology-supported CBR approach in the paper can be used as a complementary approach to improve the approach in [26]. Therefore, the ontology for generation of assembly tolerance types in [26] can be directly reused when constructing the tolerance specification ontology.

- Enumerate important terms in the ontology. According to the definitions of tolerance specification cases (Definitions 1, 2, and 3), important terms are case, tolerance specification case, number, last revised date, author, primary datum, secondary datum, tertiary datum, tolerance type, general tolerance principle, primary datum tolerance principle, secondary datum tolerance principle, tertiary datum tolerance principle, toleranced feature, part, part feature, topological relation, previous case, and target case.
- Define the classes and the class hierarchy. Among the enumerated terms, terms representing unary relations are case, tolerance specification case, toleranced feature, previous case, and target case. These five terms are defined as classes *Case*, *TSCase*, *TolerancedFeature*, *PreviousTSCase*, and *TargetTSCase*, respectively. Obviously, *Case* should be defined as the parent class of *TSCase*, and *TSCase* should be defined as the parent class of *PreviousTSCase* and *TargetTSCase*.
- Define the properties of classes (slots). Among the enumerated terms, terms representing binary relations are number, last revised date, author, primary datum, secondary datum, tertiary datum, tolerance type, general tolerance principle, primary datum tolerance principle, secondary datum tolerance principle, tertiary datum tolerance principle, toleranced feature, part, part feature, and topological relation. They are defined as object properties *hasNumber*, *hasLastRevisedDate*, *hasAuthor*, *hasPrimaryDatum*, *hasSecondaryDatum*, *hasTertiaryDatum*, *hasToleranceType*, *hasGTolerancePrinciple*, *hasPDTolerancePrinciple*, *hasSDTolerancePrinciple*, *hasTDTolerancePrinciple*, *hasTolerancedFeature*, *hasPart*, *hasPartFeature*, and *hasTopologicalRelation*, respectively.
- Define the facets (domain and range) of the slots. The domain and range of a property can be defined on the basis of the function of this property. For instance, the object property *hasNumber* is used to link a previous tolerance specification case to a number. Thus its domain and range are defined as the classes *PreviousTSCase* and *Number* (*Number* is a class), respectively. The domain and range of the remaining properties are defined by a similar way.
- Create instances. The authors have manually collected one hundred previous cases from the tolerancing standards ISO 1101 [1] and ASME Y14.5 [2] and several technical handbooks to instantiate the tolerance specification ontology at present. For example, the previous case in Fig. 3(a) is one of these one hundred previous cases. The following instances and assertions have been created for this previous case:

$$A_S = \{ \text{PreviousTSCase}(C_P), \text{Number}(PC000001), \text{Date}(\text{OCT } 06, 2016), \text{Author}(\text{Yuchu Qin}), \\ \text{DatumLine}(A-B), \text{CircularRunout}(cr), \text{CylindricalSurface}(f_{P,4}), \text{Part}(p_P), \\ \text{RealPlanar}(f_{P,1}), \text{RealCylindrical}(f_{P,2}), \text{RealPlanar}(f_{P,3}), \text{RealCylindrical}(f_{P,4}), \text{RealPlanar}(f_{P,5}), \\ \text{RealCylindrical}(f_{P,6}), \text{RealPlanar}(f_{P,7}), \text{ParallelRelation}(\textit{parallel}), \\ \text{hasNumber}(C_P, PC000001), \text{hasLastRevisedDate}(C_P, \text{OCT } 06, 2016), \\ \text{hasAuthor}(C_P, \text{Yuchu Qin}), \text{hasPrimaryDatum}(C_P, A-B), \text{hasSecondaryDatum}(C_P, \text{Null}), \\ \text{hasTertiaryDatum}(C_P, \text{Null}), \text{hasToleranceType}(C_P, cr), \text{hasGTolerancePrinciple}(C_P, \text{Null}), \\ \text{hasPDTolerancePrinciple}(C_P, \text{Null}), \text{hasSDTolerancePrinciple}(C_P, \text{Null}), \\ \text{hasTDTolerancePrinciple}(C_P, \text{Null}), \text{hasTolerancedFeature}(C_P, f_{P,4}), \text{hasPart}(C_P, p_P), \}$$

hasPartFeature($C_P, f_{P,1}$), hasPartFeature($C_P, f_{P,2}$), hasPartFeature($C_P, f_{P,3}$),
hasPartFeature($C_P, f_{P,4}$), hasPartFeature($C_P, f_{P,5}$), hasPartFeature($C_P, f_{P,6}$),
hasPartFeature($C_P, f_{P,7}$), hasTopologicalRelation($C_P, parallel$)}

3.2. Retrieval of tolerance specification cases

In CBR [6], case retrieval is the process of retrieving similar previous cases from case base for each given target case. The most essential step in this process is to assess the similarity of a target case and a previous case. This sub-section aims to complete this step for tolerance specification case retrieval. As can be seen from a comparison of the definitions of a previous case and a target case for tolerance specification (Definitions 2 and 3), the attributes tolerated feature, part features, and topological relation are distinctive attributes for the similarity between the previous case and the target case because the similarities of these attributes directly determine this similarity. The sub-section firstly illustrates how to quantify the similarity of tolerated features, the similarity of part features, and the similarity of topological relations. It then explains how to compute the similarity of target and previous cases. Finally, a retrieval algorithm of similar previous cases is designed based on such similarity.

3.2.1. Similarity between tolerated features

It can be seen from Definitions 1, 2, and 3 that every tolerance specification case (either a previous case or a target case) contains a tolerated feature. While in tolerance specification, different tolerated features can be applied different tolerance types. For this reason, the similarity between tolerated features can be quantified by their applicable tolerance types. Formally, let $F_{T,T}$ be the tolerated feature in a target case C_T , $F_{T,P}$ be the tolerated feature in a previous case C_P , $S(F_{T,T})$ be a set of the applicable tolerance types of $F_{T,T}$, and $S(F_{T,P})$ be a set of the applicable tolerance types of $F_{T,P}$. Then the similarity between $F_{T,T}$ and $F_{T,P}$ can be quantified by the similarity between $S(F_{T,T})$ and $S(F_{T,P})$ (i.e. $Sim(F_{T,T}, F_{T,P}) = Sim(S(F_{T,T}), S(F_{T,P}))$).

In GD&T, there are totally ten different kinds of tolerated features: point, straight line, spherical surface, cylindrical surface, planar surface, helical surface, revolute surface, prismatic surface, complex surface, and feature of size [18]. Each kind of tolerated feature may be applied one or more tolerance types. Table 1 lists the applicable tolerance types of these tolerated features.

Through searching Table 1, the elements of $S(F_{T,T})$ and $S(F_{T,P})$ can be obtained. For instance, assume $F_{T,T}$ is the tolerated feature $f_{T,5}$ in Fig. 3(b), and $F_{T,P}$ is the tolerated feature $f_{P,4}$ in Fig. 3(a). Then $S(F_{T,T}) = \{ \text{—}, \square, //, \perp, \angle, \oplus, \equiv, \nearrow, \searrow, \swarrow, \nwarrow \}$ and $S(F_{T,P}) = \{ \text{—}, \bigcirc, \oslash, \nearrow, \searrow \}$ since the tolerance specification ontology instantiated by the two examples in Fig. 3 have the assertions PlanarSurface($f_{T,5}$) and CylindricalSurface($f_{P,4}$). As can be seen from this example, the question of computing the similarity between tolerated features can be transformed into a question of computing the similarity between two sets.

For the computation of the similarity between two sets, there have been three different similarity measures whose mathematical foundation is set theory at present. The three measures are Maedche and Staab's measure [42], Rodriguez and Egenhofer's measure [43], and Sanchez et al.'s measure [44]. Since Sanchez et al. have experimentally verified that the accuracy of their measure is higher than the accuracy of the first two measures under certain standard benchmarks [44], this measure is selected to compute the similarity between the tolerated features F_T and F_P :

$$Sim_{\text{Sanchez}}(F_{T,T}, F_{T,P}) = 1 - \log_2 \left(1 + \frac{|S(F_{T,T}) \setminus S(F_{T,P})| + |S(F_{T,P}) \setminus S(F_{T,T})|}{|S(F_{T,T}) \cup S(F_{T,P})|} \right) \quad (1)$$

Table 1

Applicable tolerance types of ten different kinds of tolerated features. Notes: TT is short for tolerance type; SL is short for straight line; SS is short for spherical surface; CS is short for cylindrical surface; PS is short for planar surface; RS is short for revolute surface; TS is short for prismatic surface; XS is short for complex surface; FOS is short for feature of size; \curvearrowright stands for angle tolerance; \nearrow stands for linear dimension tolerance.

TT	PT	SL	SS	CS	PS	RS	TS	XS	FOS
—		⊙		⊙	⊙	⊙	⊙		⊙
▭					⊙				⊙
○			⊙	⊙		⊙			
↗				⊙					
∩						⊙	⊙	⊙	
∪						⊙	⊙	⊙	
//		⊙			⊙				⊙
⊥		⊙			⊙				⊙
∠		⊙			⊙				⊙
⊕	⊙	⊙			⊙				⊙
⊗	⊙	⊙							⊙
≡					⊙				⊙
\nearrow				⊙	⊙	⊙			
\nearrow				⊙	⊙				
\curvearrowright		⊙			⊙				
\curvearrowright	⊙	⊙			⊙				

For example, the similarity between the tolerated features $f_{T,5}$ and $f_{P,4}$ in Fig. 3 can be computed as follow:

$$Sim_{\text{Sanchez}}(F_T(f_{T,5}), F_T(f_{P,4})) = 1 - \log_2\{1 + [(8 + 2) \div 13]\} = 0.1769$$

3.2.2. Similarity between part features

Every tolerated feature belongs to a part which consists of a certain number of part features. According to a classification by Srinivasan [22], a part feature may be a spherical surface, a cylindrical surface, a planar surface, a helical surface, a revolute surface, a prismatic surface, or a complex surface (these seven types of surfaces are called as seven invariance classes). Naturally, the similarity between part features can be quantified by the invariance classes they belong to.

Formally, let $F_{P,T}$ be the set of part features in a target case C_T , $F_{P,P}$ be the set of part features in a previous case C_P , $S(F_{P,T})$ be a set that consists of the class assertions of all the part features in $F_{P,T}$, and $S(F_{P,P})$ be a set that consists of the class assertions of all the part features in $F_{P,P}$. Then the similarity between $F_{P,T}$ and $F_{P,P}$ can be quantified by the similarity between the two sets $S(F_{P,T})$ and $S(F_{P,P})$ (i.e. $Sim(F_{P,T}, F_{P,P}) = Sim(S(F_{P,T}), S(F_{P,P}))$). The similarity $Sim(S(F_{P,T}), S(F_{P,P}))$ can also be calculated using Sanchez et al.'s measure:

$$Sim_{\text{Sanchez}}(F_{P,T}, F_{P,P}) = 1 - \log_2 \left(1 + \frac{|S(F_{P,T}) \setminus S(F_{P,P})| + |S(F_{P,P}) \setminus S(F_{P,T})|}{|S(F_{P,T}) \cup S(F_{P,P})|} \right) \quad (2)$$

For example, assume $F_{P,T}$ is the set of the part features in Fig. 3(b), $F_{P,P}$ is the set of the part features in Fig. 3(a). Then according to the tolerance specification ontology instantiated by the two examples in Fig. 3, the two sets $S(F_{P,T})$ and $S(F_{P,P})$ are respectively obtained as follows:

$$S(\mathbf{F}_{P,T}) = \{\text{RealPlanar}(f_{T,1}), \text{RealCylindrical}(f_{T,2}), \text{RealPlanar}(f_{T,3}), \text{RealCylindrical}(f_{T,4}), \text{RealPlanar}(f_{T,5})\}$$

$$S(\mathbf{F}_{P,P}) = \{\text{RealPlanar}(f_{P,1}), \text{RealCylindrical}(f_{P,2}), \text{RealPlanar}(f_{P,3}), \text{RealCylindrical}(f_{P,4}), \text{RealPlanar}(f_{P,5}), \\ \text{RealCylindrical}(f_{P,6}), \text{RealPlanar}(f_{P,7})\}$$

As can be seen from the elements of the sets $S(\mathbf{F}_{P,T})$ and $S(\mathbf{F}_{P,P})$, $S(\mathbf{F}_{P,T})$ contains 3 planar surfaces and 2 cylindrical surfaces, while $S(\mathbf{F}_{P,P})$ contains 4 planar surfaces and 3 cylindrical surfaces. Based on this, the value of $Sim_{\text{Sanchez}}(\mathbf{F}_{P,T}, \mathbf{F}_{P,P})$ is calculated as follow:

$$Sim_{\text{Sanchez}}(\mathbf{F}_P(f_{T,5}), \mathbf{F}_P(f_{P,4})) = 1 - \log_2\{1 + [(2 + 0) \div 7]\} = 0.6374$$

3.2.3. Similarity between topological relations

As can be seen from Definitions 1, 2, and 3, every tolerance specification case (either a previous case or a target case) contains a topological relation. While in tolerance specification, different topological relations correspond to different tolerance types. For this reason, the similarity between topological relations can be quantified by their corresponding tolerance types. Formally, let $R_{T,T}$ be the topological relation in a target case C_T , $R_{T,P}$ be the topological relation in a previous case C_P , $S(R_{T,T})$ be a set of the corresponding tolerance types of $R_{T,T}$, and $S(R_{T,P})$ be a set of the corresponding tolerance types of $R_{T,P}$. Then the similarity between $R_{T,T}$ and $R_{T,P}$ can be quantified by the similarity between $S(R_{T,T})$ and $S(R_{T,P})$ (i.e. $Sim(R_{T,T}, R_{T,P}) = Sim(S(R_{T,T}), S(R_{T,P}))$).

In GD&T, there are totally seven different kinds of topological relations: coincident relation, disjoint relation, inclusive relation, parallel relation, perpendicular relation, intersecting relation, and nonuniplanar relation [27]. Each kind of topological relation may correspond to one or more tolerance types. Table 2 lists the corresponding tolerance types of these seven kinds of topological relations.

Table 2

Corresponding tolerance types of seven different kinds of topological relations. Notes: TR is short for topological relation; COI is short for coincident relation; DIS is short for disjoint relation; INC is short for inclusive relation; PAR is short for parallel relation; PER is short for perpendicular relation; INT is short for intersecting relation; NON is short for nonuniplanar relation; \sphericalangle stands for angle tolerance; \nearrow stands for linear dimension tolerance.

TR	//	\perp	\sphericalangle	\oplus	\odot	\equiv	\nearrow	\nearrow	\sphericalangle	\nearrow
COI				⊙	⊙	⊙	⊙	⊙		
DIS				⊙						⊙
INC				⊙		⊙				
PAR	⊙			⊙		⊙	⊙	⊙		⊙
PER		⊙					⊙	⊙		
INT			⊙				⊙		⊙	
NON				⊙						⊙

By searching Table 2, the elements of $S(R_{T,T})$ and $S(R_{T,P})$ can be obtained. For example, assume $R_{T,T}$ is the topological relation of the toleranced feature $f_{T,5}$ with respect to datum A in Fig. 3(b), and $R_{T,P}$ is the topological relation of the toleranced feature $f_{P,4}$ with respect to datum A–B in Fig. 3(a), Then $S(R_{T,T}) = \{\perp, \nearrow, \nearrow\}$ and $S(R_{T,P}) = \{//, \oplus, \equiv, \nearrow, \nearrow, \sphericalangle\}$ since the tolerance specification ontology instantiated by the two examples in Fig. 3 have the assertions $hasTopologicalRelation(C_T, perpendicular)$ and $hasTopologicalRelation(C_P, parallel)$. It can be seen from this example that the question of computing the similarity between topological relations can also be transformed into a

question of computing the similarity between two sets. Naturally, the transformed similarity can also be calculated using Sanchez et al.'s measure:

$$Sim_{\text{Sanchez}}(R_{T,T}, R_{T,P}) = 1 - \log_2 \left(1 + \frac{|S(R_{T,T}) \setminus S(R_{T,P})| + |S(R_{T,P}) \setminus S(R_{T,T})|}{|S(R_{T,T}) \cup S(R_{T,P})|} \right) \quad (3)$$

For instance, the similarity between the topological relation of the tolerated feature $f_{T,5}$ with respect to datum A in Fig. 3(b) ($R_{T,T}$) and the topological relation of the tolerated feature $f_{P,4}$ with respect to datum A–B in Fig. 3(a) ($R_{T,P}$) can be computed as follow:

$$Sim_{\text{Sanchez}}(R_T(f_{T,5}), R_T(f_{P,4})) = 1 - \log_2\{1 + [(1 + 4) \div 7]\} = 0.2224$$

3.2.4. Similarity between target and previous cases

In general, a case is represented by a set of attributes, some of which are distinctive attributes (the similarities between distinctive attributes directly determine the similarity between cases). Based on such representation, a local similarity measure is designed for each distinctive attribute and a global similarity measure is usually defined as a weighted sum of all local similarity measures [45].

So far, three similarity measures for respectively computing the similarities between the tolerated features of target and previous cases ($Sim_{\text{Sanchez}}(F_{T,T}, F_{T,P})$), between the part features of target and previous cases ($Sim_{\text{Sanchez}}(F_{P,T}, F_{P,P})$), and between the topological relations of target and previous cases ($Sim_{\text{Sanchez}}(R_{T,T}, R_{T,P})$) have been designed. A similarity measure for assessing the similarity between target and previous cases $Sim(C_T, C_P)$ can be naturally defined as a weighted sum of these three similarity measures:

$$Sim(C_T, C_P) = \lambda_1 Sim_{\text{Sanchez}}(F_{T,T}, F_{T,P}) + \lambda_2 Sim_{\text{Sanchez}}(F_{P,T}, F_{P,P}) + \lambda_3 Sim_{\text{Sanchez}}(R_{T,T}, R_{T,P}) \quad (4)$$

where λ_1 , λ_2 , and λ_3 are respectively the weights of $Sim_{\text{Sanchez}}(F_{T,T}, F_{T,P})$, $Sim_{\text{Sanchez}}(F_{P,T}, F_{P,P})$, and $Sim_{\text{Sanchez}}(R_{T,T}, R_{T,P})$ such that $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

As can be seen from Expression (4), the assignment of the weights λ_1 , λ_2 , and λ_3 directly affects the accuracy of $Sim(C_T, C_P)$ and the accuracy of tolerance specification case retrieval. Generally, the accuracy of a similarity measure is quantified by the Pearson correlation coefficient between the similarities of a certain number of randomly selected sample pairs computed by the similarity measure and the similarities of these sample pairs judged by a certain number of domain experts. The greater the coefficient, the higher the accuracy of the similarity measure is [44]. Therefore, the values of λ_1 , λ_2 , and λ_3 can be computed through maximizing the Pearson correlation coefficient between the similarities of sample case pairs calculated by $Sim(C_T, C_P)$ and the similarities of these sample case pairs judged by domain experts.

Formally, let $\underline{Sim}(C_T, C_P) = \underline{\lambda}_1 Sim_{\text{Sanchez}}(F_{T,T}, F_{T,P}) + \underline{\lambda}_2 Sim_{\text{Sanchez}}(F_{P,T}, F_{P,P}) + \underline{\lambda}_3 Sim_{\text{Sanchez}}(R_{T,T}, R_{T,P})$ (where $\underline{\lambda}_1$, $\underline{\lambda}_2$, and $\underline{\lambda}_3$ are arbitrary real numbers), $Sim_J(C_T, C_P)$ be the similarity of C_T and C_P judged by domain experts, vector $\mathbf{U} = [Sim_{\text{Sanchez}}(F_{T,T}, F_{T,P}), Sim_{\text{Sanchez}}(F_{P,T}, F_{P,P}), Sim_{\text{Sanchez}}(R_{T,T}, R_{T,P})]^T$, vector $\mathbf{V} = [Sim_J(C_T, C_P)]^T$, and vector $\underline{\lambda} = [\underline{\lambda}_1, \underline{\lambda}_2, \underline{\lambda}_3]^T$. Then the question of assigning λ_1 , λ_2 , and λ_3 is converted to a question of seeking a vector $\underline{\lambda}$ which can maximize the Pearson correlation coefficient between $\underline{\lambda}^T \mathbf{U}$ and \mathbf{V} :

$$pcc(\underline{\lambda}^T \mathbf{U}, \mathbf{V}) = \frac{\underline{\lambda}^T \text{cov}(\mathbf{U}, \mathbf{V})}{\sqrt{\underline{\lambda}^T \text{cov}(\mathbf{U}, \mathbf{U}) \underline{\lambda}} \sqrt{\text{cov}(\mathbf{V}, \mathbf{V})}} = \frac{\underline{\lambda}^T \Sigma_{UV}}{\sqrt{\underline{\lambda}^T \Sigma_{UU} \underline{\lambda}} \sqrt{\Sigma_{VV}}} \quad (5)$$

where $\text{cov}(\mathbf{U}, \mathbf{V})$ is the covariance of the vectors \mathbf{U} and \mathbf{V} , $\text{cov}(\mathbf{U}, \mathbf{U})$ is the covariance of the vectors \mathbf{U} and \mathbf{U} , and $\text{cov}(\mathbf{V}, \mathbf{V})$ is the covariance of the vectors \mathbf{V} and \mathbf{V} .

To solve the vector $\underline{\lambda}$ which can maximize $\text{pcc}(\underline{\lambda}^T \mathbf{U}, \mathbf{V})$, a canonical correlation analysis based method [46] is used and the solving process is as follows.

Firstly, let $\mathbf{u} = \Sigma_{UU}^{-1/2} \underline{\lambda}$ and $\mathbf{v} = \Sigma_{VV}^{-1/2} \underline{\lambda}$. Then Expression (11) can be transformed to:

$$\text{pcc}(\underline{\lambda}^T \mathbf{U}, \mathbf{V}) = \frac{\mathbf{u}^T \Sigma_{UU}^{-1/2} \Sigma_{UV} \Sigma_{VV}^{-1/2} \mathbf{v}}{\sqrt{\mathbf{u}^T \mathbf{u}} \sqrt{\mathbf{v}^T \mathbf{v}}} = \frac{\delta}{\sqrt{\mathbf{u}^T \mathbf{u}} \sqrt{\mathbf{v}^T \mathbf{v}}} \quad (6)$$

Then according to the Cauchy-Schwarz inequality, the following inequality is obtained:

$$\delta \leq \sqrt{\mathbf{u}^T \Sigma_{UU}^{-1/2} \Sigma_{UV} \Sigma_{VV}^{-1} \Sigma_{VU} \Sigma_{UU}^{-1/2} \mathbf{u}} \sqrt{\mathbf{v}^T \mathbf{v}} \quad (7)$$

where Σ_{VV} is the covariance of the vectors \mathbf{V} and \mathbf{U} .

Finally, the following inequality is obtained through combining Expressions (12) and (13):

$$\text{pcc}(\underline{\lambda}^T \mathbf{U}, \mathbf{V}) \leq \frac{\sqrt{\mathbf{u}^T \Sigma_{UU}^{-1/2} \Sigma_{UV} \Sigma_{VV}^{-1} \Sigma_{VU} \Sigma_{UU}^{-1/2} \mathbf{u}}}{\sqrt{\mathbf{u}^T \mathbf{u}}} \quad (8)$$

As can be seen from Expression (14), the maximum value of $\text{pcc}(\underline{\lambda}^T \mathbf{U}, \mathbf{V})$ is attained if and only if \mathbf{u} is the eigenvector with the maximum eigenvalue for the matrix $\Sigma_{UU}^{-1/2} \Sigma_{UV} \Sigma_{VV}^{-1} \Sigma_{VU} \Sigma_{UU}^{-1/2}$. Thus the solution is: $\underline{\lambda}$ is an eigenvector with the maximum eigenvalue for the matrix $\Sigma_{UU}^{-1} \Sigma_{UV} \Sigma_{VV}^{-1} \Sigma_{VU}$.

To solve the real weights λ_1 , λ_2 , and λ_3 , the obtained vector $\underline{\lambda}$ is normalized as follow: For all $\lambda_i < 0$ ($i = 1, 2, 3$), let $\lambda_i = 0$. Then re-solve the new vector $\underline{\lambda}$ until all $\lambda_i \geq 0$. Finally, let $\lambda_i = \lambda_i / (\lambda_1 + \lambda_2 + \lambda_3)$. A real weight vector $\lambda = [\lambda_1, \lambda_2, \lambda_3]^T$ obtained after such normalization is a vector which can maximize $\text{pcc}(\lambda^T \mathbf{U}, \mathbf{V})$ because:

$$\begin{aligned} \text{pcc}(\lambda^T \mathbf{U}, \mathbf{V}) &= \frac{\lambda^T \Sigma_{UV}}{\sqrt{\lambda^T \Sigma_{UU} \lambda} \sqrt{\Sigma_{VV}}} = \frac{[\lambda / (\lambda_1 + \lambda_2 + \lambda_3)]^T \Sigma_{UV}}{\sqrt{[\lambda / (\lambda_1 + \lambda_2 + \lambda_3)]^T \Sigma_{UU} [\lambda / (\lambda_1 + \lambda_2 + \lambda_3)]} \sqrt{\Sigma_{VV}}} \\ &= \frac{\underline{\lambda}^T \Sigma_{UV}}{\sqrt{\underline{\lambda}^T \Sigma_{UU} \underline{\lambda}} \sqrt{\Sigma_{VV}}} = \text{pcc}(\underline{\lambda}^T \mathbf{U}, \mathbf{V}) \end{aligned} \quad (9)$$

After solving the real weights λ_1 , λ_2 , and λ_3 , the measure $\text{Sim}(\mathbf{C}_T, \mathbf{C}_P)$ having the highest accuracy can be determined. This measure will be selected to implement tolerance specification case retrieval.

3.2.5. Retrieval algorithm of similar previous cases

Based on the above explanation of accurately assessing the similarity between target and previous cases, a retrieval algorithm of similar previous cases is designed as follow:

Tolerance specification case retrieval algorithm

Case base: A tolerance specification ontology that consists of a tolerance specification case base totally containing n previous cases $\mathbf{C}_{P,k}$ ($k = 1, 2, \dots, n$)

Function DetermineASimilarityMeasure

Input: N sample case pairs $(\mathbf{C}_{T,i}, \mathbf{C}_{P,i})$ ($i = 1, 2, \dots, N$) and their judged similarities $\text{Sim}_J(\mathbf{C}_{T,i}, \mathbf{C}_{P,i})$

Output: A similarity measure $\text{Sim}(\mathbf{C}_T, \mathbf{C}_P)$ that has the greatest pcc value with $\text{Sim}_J(\mathbf{C}_{T,i}, \mathbf{C}_{P,i})$

Step 1: for integer $i \leftarrow 1$ to N do

Use $Sim_{Sanchez}(F_{T,T}, F_{T,P})$ to compute the similarities between the tolerated features of $(C_{T,i}, C_{P,i})$

Use $Sim_{Sanchez}(F_{P,T}, F_{P,P})$ to compute the similarities between the part features of $(C_{T,i}, C_{P,i})$

Use $Sim_{Sanchez}(R_{T,T}, R_{T,P})$ to compute the similarities between the topological relations of $(C_{T,i}, C_{P,i})$

Step 2: Use the weight solving procedure to compute the values of λ_1 , λ_2 , and λ_3

Step 3: Output the measure $Sim(C_T, C_P) = \lambda_1 Sim_{Sanchez}(F_{T,T}, F_{T,P}) + \lambda_2 Sim_{Sanchez}(F_{P,T}, F_{P,P}) + \lambda_3 Sim_{Sanchez}(R_{T,T}, R_{T,P})$

End DetermineASimilarityMeasure

Function RetrieveSimilarPreviousCases

Input: A target case C_T ; A measure that has the greatest pcc value; A similarity threshold t ; A positive integer N_+

Output: N_+ previous cases sorted in descending order on the basis of their similarities with C_T

Step 1: Extract the datum, tolerated features, part features, and topological relations of the part in C_T

Instantiate the tolerance specification ontology according to the extracted information

for integer $k \leftarrow 1$ to n do

Use the measure that has the greatest pcc value to compute the similarities between C_T and $C_{P,k}$

Step 2: for integer $k \leftarrow 1$ to n do

Find out all of the (n_i) previous cases whose similarities with C_T are greater than or equal to t

Step 3: Sort the n_i previous cases in descending order on the basis of their similarities with C_T

Step 4: Output the top N_+ previous cases in the n_i sorted previous cases

End RetrieveSimilarPreviousCases

The designed algorithm consists of two functions. The former function is used to determine a similarity measure with high accuracy to implement tolerance specification case retrieval. It takes as input N sample case pairs and their judged similarities, and it returns as output a similarity measure for assessing the similarity between target and previous cases that has the greatest Pearson correlation coefficient with the judged similarities. For an identical case base, this function requires being executed only once. The later function is used to retrieve similar previous cases from the tolerance specification case base to assist tolerance specification for a target case. It takes as input a target case, a similarity measure that has the greatest Pearson correlation coefficient, a similarity threshold, and a positive integer, and it returns as output a specified number of previous cases sorted in descending order on the basis of their similarities with the target case. This function is executed once whenever a new target case is entered.

The complexity of the designed algorithm is analyzed as follows. For the former function, the computation amount of **Step 1** is $3N$ (each measure needs N computations), and the computation amount of **Step 2** is 1. Thus, the total computation amount of the first function is $3N + 1$. For the later function, the computation amounts of **Step 1** and **Step 2** are both n . The computation amount of **Step 3** is $n_i \log_2 n_i$ (the computation amount of the sorting procedure). Consequently, the total computation amount of this function is $2n + n_i \log_2 n_i$, and the total computation amount of the algorithm is $2n + n_i \log_2 n_i + 3N + 1$. Since generally the values of n_i and N are both far less than the value of n , the complexity of the algorithm is $O(n)$.

3.3. Reuse and revision of tolerance specification cases

In CBR [6], case reuse is the process of proposing a solution for the new problem in each given target case from the solutions in the retrieved previous cases. The most essential step in this process is to judge whether a retrieved

solution can be directly reused as a solution for the new problem. Such judgement depends on the new problem itself. For a new tolerance specification problem in a given target case, the judgement can be made according to the functional and assembly requirements of the tolerated feature in the given target case. If such requirements are identical to the functional and assembly requirements of the tolerated feature in a retrieved similar previous tolerance specification case, then the solution in this case can be directly reused to address the new tolerance specification problem and the new tolerance specification problem and its solution do not need to be retained as a new previous case. Otherwise, the solution needs to be revised (or adapted) before it is used. In this situation, case revision is required.

Revising (or adapting) the solution in a retrieved similar previous case to transform it into a solution that can be used to solve the new problem in a given target case is called case revision (or adaptation) in CBR [6]. It appears after case reuse and is followed by case retention. During the past few decades, a number of strategies for case revision have been presented. They were classified into substitution strategy, transformation strategy, and generation strategy [45]. Substitution strategy simply revises some attribute values of the retrieved solution, whereas transformation strategy will change the structure of the retrieved solution (i.e. add some attributes to the retrieved solution or remove some attributes from the retrieved solution). Generation strategy re-executes the method of generating the retrieved solutions on the new problem.

Based on the characteristics of tolerance specification problem, a combination of substitution and transformation strategies is adopted in revision of tolerance specification cases (i.e. such revision may need to modify attribute values, structure, or both of them). Like the judgement in reuse of tolerance specification cases, revision of tolerance specification cases is also implemented according to the functional and assembly requirements of the tolerated feature in the given target case. The following rules, which are summarized from the mirror approach [24], can be used to guide such implementation:

- If the tolerated feature is used to arrest leakage between two parts, then it needs to be specified tight form (straightness, flatness, roundness, cylindricity, profile any line, or profile any surface) and dimension tolerances.
- If the tolerated feature is used to maintain rotation between two parts, then it needs to be specified tight location (position, concentricity, coaxiality, or symmetry) and form tolerances.
- If the tolerated feature is used to maintain symmetry between two parts and is selected as a datum feature, then it cannot be applied maximum material requirement (MMR).
- If the tolerated feature is used to maintain gear mesh between two parts, then then it needs to be specified profile any line or profile any surface tolerance.
- If the tolerated feature is used to connect two parts, then it needs to be applied MMR.
- If the tolerated feature is used to maintain sliding between two parts, then it needs to be specified form tolerance.
- If the tolerated feature is in an interference fit, then regardless of feature size is recommended for it.
- If the tolerated feature is in a clearance fit, then MMR is recommended for it.
- If the tolerated feature is in an assembling having location function, then it is chosen as the candidate feature for secondary datum feature.
- If the tolerated feature is in an assembling having seat function, then it is chosen as the candidate feature for primary datum feature.
- If the tolerated feature is in an assembling having contact function, then it is chosen as the candidate feature for

tertiary datum feature.

- If the tolerated feature is in an assembling having alignment function, then it is chosen as the candidate feature for secondary datum feature.

Based on the above rules, revision of tolerance specification cases is carried out by the following three steps:

- Generate the recommended tolerance types for the tolerated feature using the SWRL rules based approach in [26]. This step consists of the following four sub-steps: (1) Extract the assembly constraint and topological relations between the tolerated feature and other features automatically using CAD system's APIs. (2) Instantiate the developed tolerance specification ontology according to the extracted assembly constraint relations. (3) Generate new ontology assertions via performing SWRL rules based reasoning on the instantiated tolerance specification ontology. (4) Obtain the recommended tolerance types for the tolerated feature according to the generated ontology assertions.
- List all possible tolerance specification schemes for the tolerated feature after applying the above rules.
- Combine the generated tolerance types and the listed tolerance specification schemes to make revision.

4. Implementation, example, and evaluation

This section firstly reports a prototype implementation of the proposed ontology-supported CBR approach. It then presents an example to illustrate how the approach works. Finally, the section evaluates the approach by theoretical and experimental comparisons.

4.1. Implementation

The tolerance specification ontology was developed using Protégé. Fig. 4 shows a partial visual representation of the developed ontology in Protégé. The CAD system used in the prototype implementation of the proposed ontology-supported CBR approach is NX. The tolerance specification case retrieval algorithm was implemented with the use of OWL APIs, NX Open APIs, and the Java programming language.

In the process of the implementation of the tolerance specification case retrieval algorithm, an important task is to determine a similarity measure with high accuracy for tolerance specification case retrieval. To complete this task, an experiment has been conducted. This experiment consists of two steps (Please note that such experiment needs to be conducted only once for an identical tolerance specification case base).

The first step is to obtain a certain number of sample case pairs and their judged similarities. In general, sample case pairs can be obtained through randomly selecting pairs of previous cases from case base and their judged similarities can be obtained through a statistics of the similarities judged by a certain number of domain experts. The sample case pairs in the experiment and their judged similarities were also obtained by this way. Specifically, totally 30 sample case pairs were randomly selected from the instantiated tolerance specification ontology (tolerance specification case base) that contains 100 previous cases (they can constitute 100×99 case pairs). Then the similarity of each sample case pair on a scale 0, 0.2, 0.4, 0.6, 0.8, and 1 was judged by 12 mechanical designers in the authors' institutions. The 12 judged similarities of each sample case pair were sorted in descending order and the first and last ones were removed. The mean value of the remaining 10 judged similarities was computed and considered as the judged similarity of each sample case pair. The judge similarities of the 30 sample case pairs are listed in Table 3.

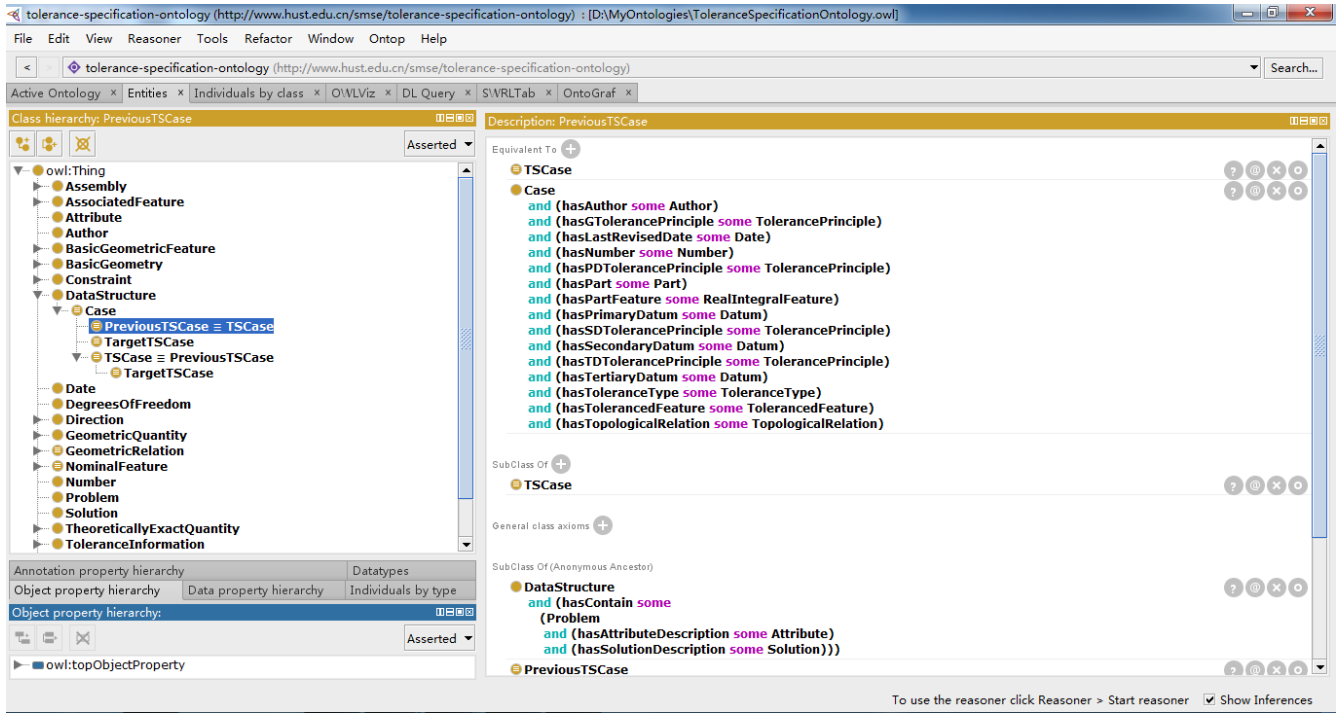


Fig. 4. A partial visual representation of the developed ontology in Protégé

Table 3

The judged similarities, the similarities between tolerated features computed by $Sim_{Sanchez}(F_{T,T}, F_{T,P})$, the similarities between part features computed by $Sim_{Sanchez}(F_{P,T}, F_{P,P})$, and the similarities between topological relations computed by $Sim_{Sanchez}(R_{T,T}, R_{T,P})$ of the 30 sample case pairs $(C_{T,i}, C_{P,i})$ ($i = 1, 2, \dots, 30$).

Sample case pair	$Sim_j(C_T, C_P)$	$Sim_{Sanchez}(F_{T,T}, F_{T,P})$	$Sim_{Sanchez}(F_{P,T}, F_{P,P})$	$Sim_{Sanchez}(R_{T,T}, R_{T,P})$
$(C_{T,1}, C_{P,1})$	0.2800	0.0000	0.4150	0.0931
$(C_{T,2}, C_{P,2})$	0.1400	0.0614	0.4854	0.1069
$(C_{T,3}, C_{P,3})$	0.2200	0.0000	0.0740	0.0931
$(C_{T,4}, C_{P,4})$	0.4400	0.4975	0.5406	0.2224
$(C_{T,5}, C_{P,5})$	0.2600	0.0614	0.1699	0.1520
$(C_{T,6}, C_{P,6})$	0.2000	0.1069	0.7776	0.1520
$(C_{T,7}, C_{P,7})$	0.8200	1.0000	0.3479	1.0000
$(C_{T,8}, C_{P,8})$	0.1600	0.0000	0.0875	0.1069
$(C_{T,9}, C_{P,9})$	0.6200	0.3479	0.5406	0.0931
$(C_{T,10}, C_{P,10})$	0.7200	0.4975	0.1375	1.0000
$(C_{T,11}, C_{P,11})$	0.8600	1.0000	0.5525	1.0000
$(C_{T,12}, C_{P,12})$	0.5200	0.2996	0.3833	0.0000
$(C_{T,13}, C_{P,13})$	0.3600	0.4975	0.2224	0.1520
$(C_{T,14}, C_{P,14})$	0.2800	0.0614	0.8301	0.0931
$(C_{T,15}, C_{P,15})$	0.9000	1.0000	0.2224	1.0000
$(C_{T,16}, C_{P,16})$	0.3400	0.0614	0.4525	0.0931
$(C_{T,17}, C_{P,17})$	0.8400	1.0000	0.3479	1.0000
$(C_{T,18}, C_{P,18})$	0.1000	0.1255	0.3219	0.0000
$(C_{T,19}, C_{P,19})$	0.1600	0.0000	0.3219	0.0000
$(C_{T,20}, C_{P,20})$	0.7600	1.0000	0.4525	1.0000

$(C_{T,21}, C_{P,21})$	0.1400	0.1069	0.0875	0.1069
$(C_{T,22}, C_{P,22})$	0.2800	0.0614	0.5305	0.2224
$(C_{T,23}, C_{P,23})$	0.1600	0.3219	0.3219	0.0931
$(C_{T,24}, C_{P,24})$	0.2600	0.0614	0.3219	0.1520
$(C_{T,25}, C_{P,25})$	0.4000	0.0000	0.4150	0.2224
$(C_{T,26}, C_{P,26})$	0.0600	0.0000	0.1699	0.0000
$(C_{T,27}, C_{P,27})$	0.3600	0.4975	0.2838	0.1520
$(C_{T,28}, C_{P,28})$	0.9400	1.0000	0.7935	1.0000
$(C_{T,29}, C_{P,29})$	0.8400	0.4975	0.3479	1.0000
$(C_{T,30}, C_{P,30})$	0.7000	0.4975	0.7370	0.2630

The second step is to determine a similarity measure with high accuracy for tolerance specification case retrieval according to the obtained sample case pairs and their judged similarities. This step was completed by the former function in the designed tolerance specification case retrieval algorithm. It firstly computed the similarities between the toleranced features of the 30 sample case pairs using $Sim_{Sanchez}(F_{T,T}, F_{T,P})$ (Expressions (1)), computed the similarities between the part features of the 30 sample case pairs using $Sim_{Sanchez}(F_{P,T}, F_{P,P})$ (Expressions (2)), and computed the similarities between the topological relations of the 30 sample case pairs using $Sim_{Sanchez}(R_{T,T}, R_{T,P})$ (Expressions (3)). The results of the computations are listed in Table 3. Then the values of the weights λ_1 , λ_2 , and λ_3 in Expression (4) were computed by the weight solving procedure. The results of the computation are: $\lambda_1 = 0.3518$, $\lambda_2 = 0.2267$, and $\lambda_3 = 0.4215$. Thus, the similarity measure with high accuracy for tolerance specification case retrieval is determined as:

$$Sim(C_T, C_P) = 0.3518Sim_{Sanchez}(F_{T,T}, F_{T,P}) + 0.2267Sim_{Sanchez}(F_{P,T}, F_{P,P}) + 0.4215Sim_{Sanchez}(R_{T,T}, R_{T,P}) \quad (10)$$

This measure can be used to accurately assess the similarity between a target case and a previous case. For example, using the measure, the similarity between the target case C_T in Fig. 3(b) and the previous case C_P in Fig. 3(a) is assessed as $Sim(C_T, C_P) = 0.3518 \times 0.1769 + 0.2267 \times 0.6374 + 0.4215 \times 0.2224 = 0.3005$.

4.2. Example

4.2.1. Retrieval of similar previous cases for a target case

The retrieval of similar previous cases for a target case is implemented by the later function in the designed tolerance specification case retrieval algorithm. An accurate retrieval of similar previous cases from the tolerance specification ontology (tolerance specification case base) to aid tolerance specification in a target case is taken as an example to illustrate how this function works. This retrieval takes as input the determined similarity measure with high accuracy for tolerance specification case retrieval, a target case shown in Fig. 5 (the part in this figure is the output shaft of the gear reducer in Fig. 6 that was designed in NX), a similarity threshold 0.8000, and a positive integer 5, and it will return as output 5 previous cases that are sorted in descending order on the basis of their similarities with the target case. The details of the process from input to output, which mainly contains three steps, are explained below.

The first step is to extract the datum, toleranced feature, part features, and topological relations of the part in the target case. Using NX Open APIs, the datum, toleranced feature, part features, and topological relations of the part in the target case in Fig. 5 were extracted from NX, and thus the target case can be represented by a seven-tuple: $C_T = (\text{"A-B"}, \text{"", ""}, A(f_5), p, \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}, f_{21}, f_{22}, f_{23}, f_{24}, f_{25}, f_{26}, f_{27}\}, \text{"Parallel"})$, where $A(f_5)$ is the axis of the cylindrical surface f_5 .

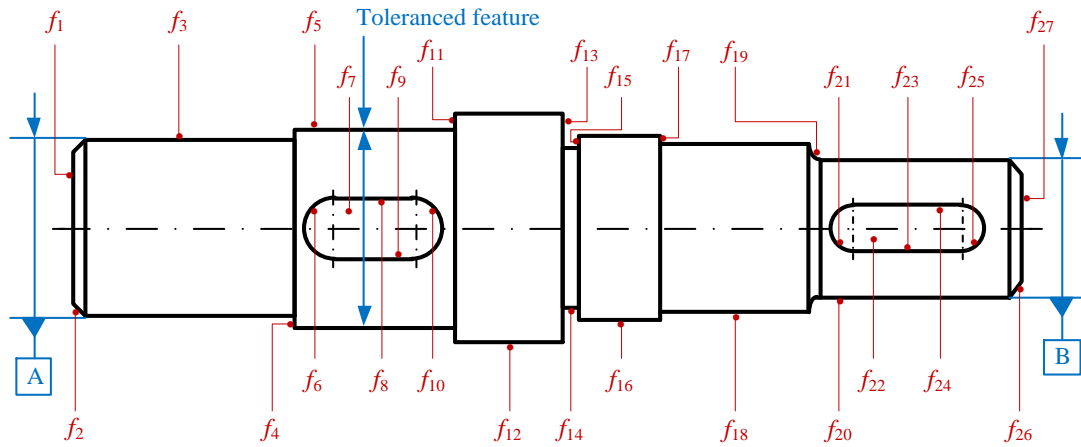


Fig. 5. A target case used to illustrate the process of tolerance specification case retrieval.

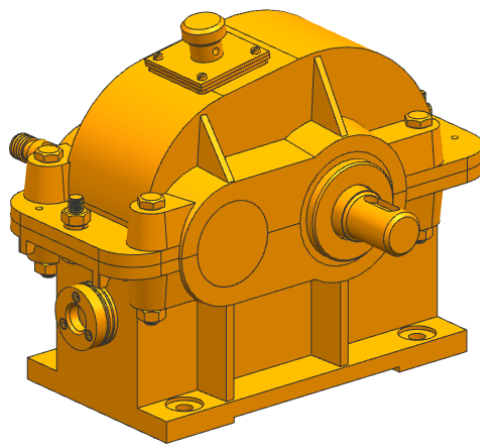


Fig. 6. A gear reducer containing the part in the target case in Fig. 5.

The second step is to instantiate the developed tolerance specification ontology according to the extracted information. Using OWL APIs, the developed tolerance specification ontology was automatically instantiated by the extracted information and so the corresponding instances and assertions were automatically created in the ontology. Fig. 7 shows the visual representation of the created assertions related to the instance C_T in Protégé

The last step is to retrieve similar previous cases for the target case. Using the determined similarity measure for tolerance specification case retrieval, the similarity between the target case in Fig. 5 and each of the one hundred previous cases in the tolerance specification ontology (tolerance specification case base) was automatically computed and all of the previous cases whose similarities with the target case are greater than or equal to 0.8000 (the input similarity threshold) were screened out. The screened out previous cases were sorted in descending order according to their similarities with the target case and the top 5 (the input positive integer) previous cases were output and are shown in Fig. 8. The details of the top one previous case are shown in Fig. 9. The tolerance specification schemes in these 5 previous cases could be used as reference tolerance specification schemes for the tolerance specification problem in the target case.

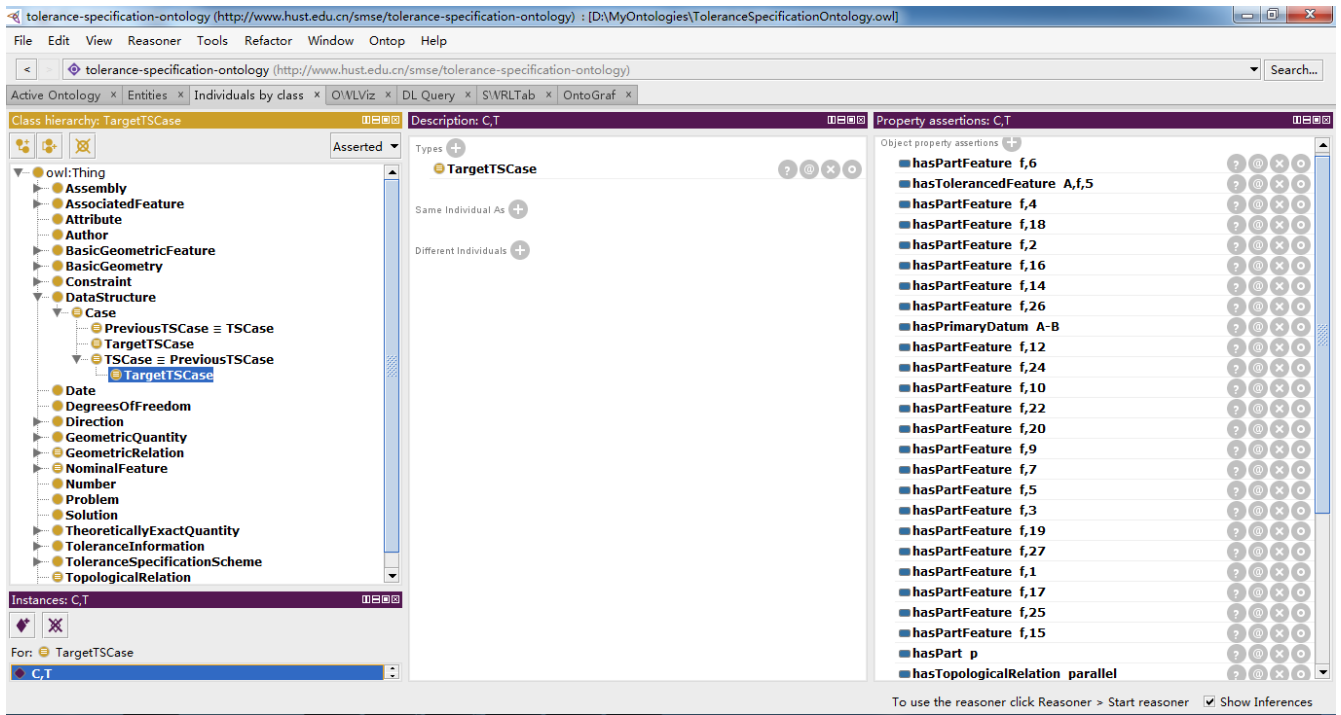


Fig. 7. The visual representation of the created assertions related to the instance C_T in Protégé

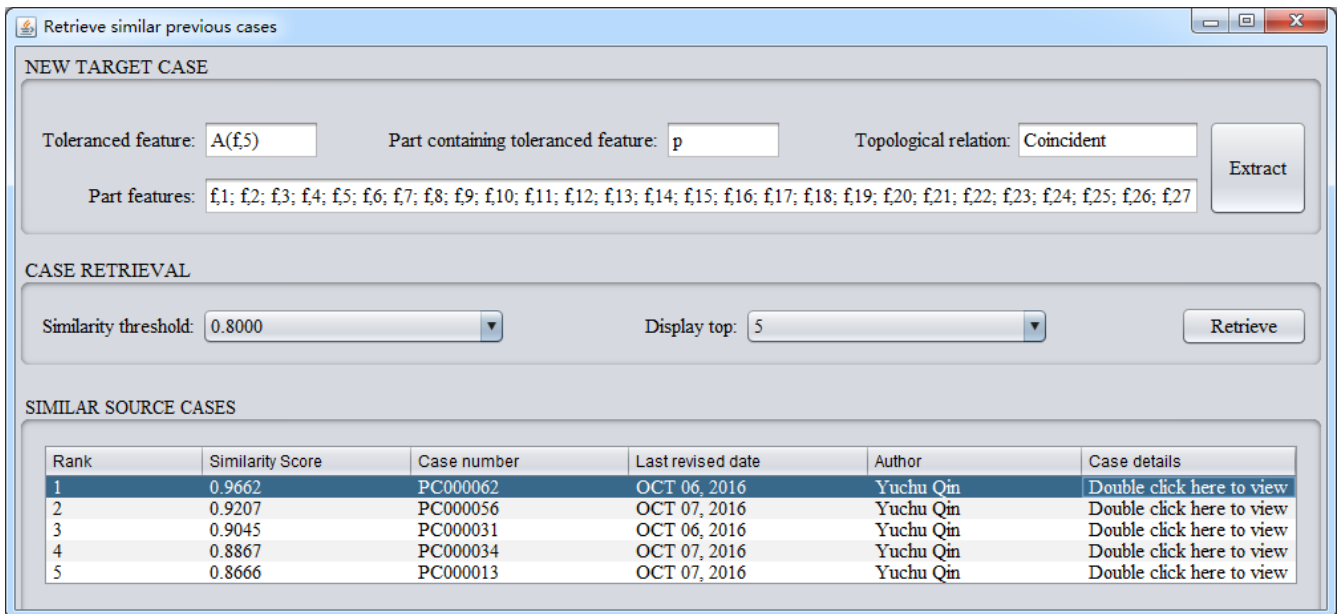


Fig. 8. The top five previous cases for tolerance specification.

4.2.2. Reuse of a previous case

The toleranced feature $A(f_5)$ in the target case in Fig. 5 is used to maintain the stability of the rotation of the whole output shaft and ensure the assemblability of the cylindrical surface f_5 . The toleranced feature $A(f_7)$ in the retrieved similar previous case PC000062 in Fig. 9 also has such functional and assembly requirements. Thus, the tolerance specification scheme in this previous case (see Fig. 9) is directly reused to solve the tolerance specification problem in the target case without revision, and a straightness with MMR and a coaxiality are specified to $A(f_5)$.

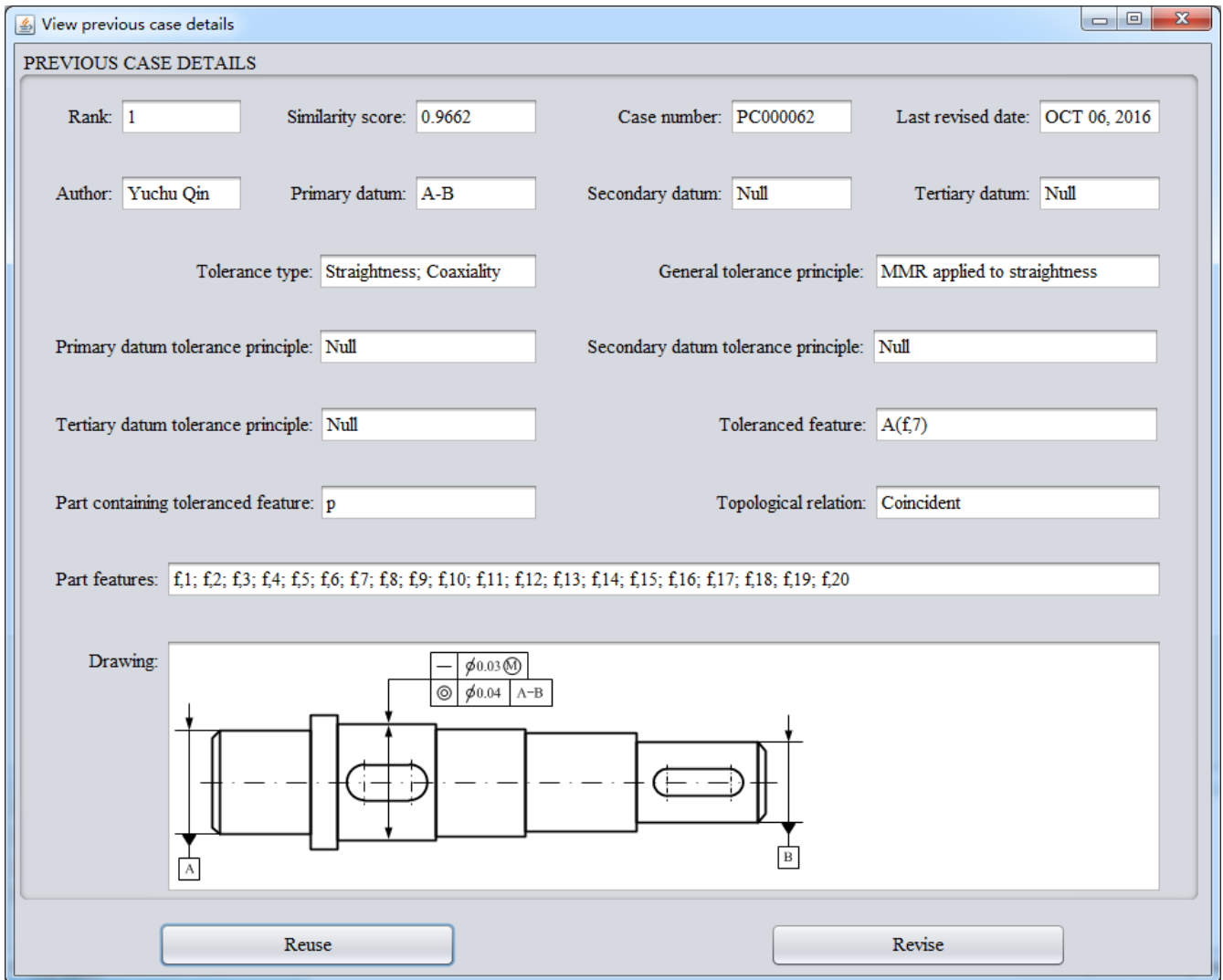


Fig. 9. The details of the top one previous case for tolerance specification.

4.2.3. Tolerance specification for a part

Tolerance specification for a part can be carried out through designing tolerance specification scheme for each toleranced feature of the part according to the proposed ontology-supported CBR approach. As an example, the tolerance specification schemes for the remaining toleranced features of the part in Fig. 5 were designed by a similar way as the design of the tolerance specification scheme for the toleranced feature A(f_5). The result of tolerance specification for this part is shown in Fig. 10.

4.3. Evaluation

4.3.1. Theoretical comparison

Since pcc value is used to quantify the accuracy of tolerance specification case retrieval in the proposed ontology-supported CBR approach, the effectiveness of the designed case retrieval algorithm can be evaluated by comparing the pcc values of the measures considering one attribute, the measures considering two attributes, and the measure determined by the former function of the algorithm.

Formally, let $(C_{T,i}, C_{P,i})$ ($i = 1, 2, \dots, N$) be N sample case pairs that are randomly selected from the tolerance specification case base, $Sim_J(C_{T,i}, C_{P,i})$ be the judged similarities of $(C_{T,i}, C_{P,i})$, $Sim_{1,1}(C_T, C_P)$, $Sim_{1,2}(C_T, C_P)$, and

$Sim_{1,3}(C_T, C_P)$ be the measures considering one attribute, $Sim_{2,1}(C_T, C_P)$, $Sim_{2,2}(C_T, C_P)$, and $Sim_{2,3}(C_T, C_P)$ be the measures considering two attributes, and $Sim(C_T, C_P)$ (Expression (4)) be the measure determined by the former function of the algorithm. Then according to Expression (4), the following expressions can be obtained:

$$Sim_{1,1}(C_T, C_P) = Sim_{Sanchez}(F_{T,T}, F_{T,P}) \quad (11)$$

$$Sim_{1,2}(C_T, C_P) = Sim_{Sanchez}(F_{P,T}, F_{P,P}) \quad (12)$$

$$Sim_{1,3}(C_T, C_P) = Sim_{Sanchez}(R_{T,T}, R_{T,P}) \quad (13)$$

$$Sim_{2,1}(C_T, C_P) = \lambda_{2,1,1}Sim_{Sanchez}(F_{T,T}, F_{T,P}) + \lambda_{2,1,2}Sim_{Sanchez}(F_{P,T}, F_{P,P}) \quad (14)$$

$$Sim_{2,2}(C_T, C_P) = \lambda_{2,2,1}Sim_{Sanchez}(F_{T,T}, F_{T,P}) + \lambda_{2,2,2}Sim_{Sanchez}(R_{T,T}, R_{T,P}) \quad (15)$$

$$Sim_{2,3}(C_T, C_P) = \lambda_{2,3,1}Sim_{Sanchez}(F_{P,T}, F_{P,P}) + \lambda_{2,3,2}Sim_{Sanchez}(R_{T,T}, R_{T,P}) \quad (16)$$

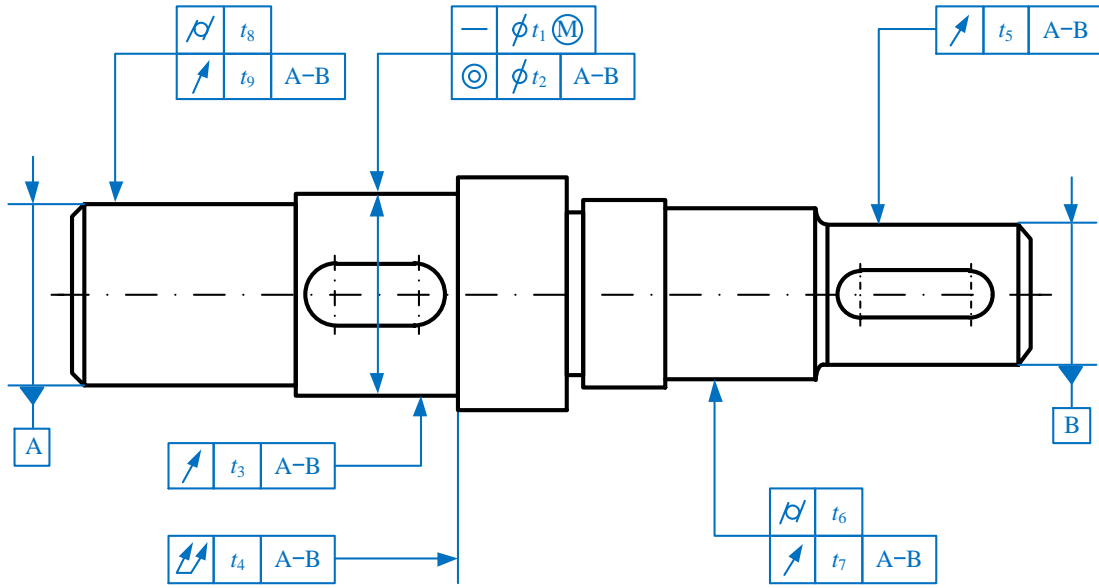


Fig. 10. The result of tolerance specification for the part in Fig. 5.

Further, let $pcc_D = pcc(Sim(C_{T,i}, C_{P,i}), Sim_j(C_{T,i}, C_{P,i}))$, and $pcc_{j,k} = pcc(Sim_{j,k}(C_{T,i}, C_{P,i}), Sim_j(C_{T,i}, C_{P,i}))$ ($j = 1, 2$; $k = 1, 2, 3$). If the proposition “ $pcc_D \geq pcc_{j,k}$ for all $j = 1, 2$ and $k = 1, 2, 3$ ” holds, then the designed case retrieval algorithm is effective. The proof of this proposition is as follow:

Proof. Let vector $U = [Sim_{Sanchez}(F_{T,T}, F_{T,P}), Sim_{Sanchez}(F_{P,T}, F_{P,P}), Sim_{Sanchez}(R_{T,T}, R_{T,P})]^T$, vector $V = [Sim_j(C_T, C_P)]^T$, and vector $\lambda = [\lambda_1, \lambda_2, \lambda_3]^T$ be the weight vector solved by the weight solving procedure. According to the algorithm, $pcc(\lambda^T U, V)$ is the greatest pcc among the pccs between the similarities of $(C_{T,i}, C_{P,i})$ assessed by all possible linear combinations of $Sim_{Sanchez}(F_{T,T}, F_{T,P})$, $Sim_{Sanchez}(F_{P,T}, F_{P,P})$, and $Sim_{Sanchez}(R_{T,T}, R_{T,P})$ and the judged similarities $Sim_j(C_{T,i}, C_{P,i})$, which include the following seven cases:

- If pcc_D obtains the maximum value when $\lambda_2 = \lambda_3 = 0$ and $\lambda_1 = 1$, then (a) $pcc_D = pcc_{1,1}$;
- If pcc_D obtains the maximum value when $\lambda_1 = \lambda_3 = 0$ and $\lambda_2 = 1$, then (b) $pcc_D = pcc_{1,2}$;
- If pcc_D obtains the maximum value when $\lambda_1 = \lambda_2 = 0$ and $\lambda_3 = 1$, then (c) $pcc_D = pcc_{1,3}$;
- If pcc_D obtains the maximum value when $\lambda_3 = 0$ and $\lambda_1, \lambda_2 \neq 0$, then (d) $pcc_D = pcc_{2,1}$;
- If pcc_D obtains the maximum value when $\lambda_2 = 0$ and $\lambda_1, \lambda_3 \neq 0$, then (e) $pcc_D = pcc_{2,2}$;

- If pcc_D obtains the maximum value when $\lambda_1 = 0$ and $\lambda_2, \lambda_3 \neq 0$, then **(f)** $pcc_D = pcc_{2,3}$;
- If pcc_D obtains the maximum value when $\lambda_1, \lambda_2, \lambda_3 \neq 0$, then **(g)** $pcc_D > pcc_{j,k}$ for all $j = 1, 2$ and $k = 1, 2, 3$.

Based on the conclusions from **(a)** to **(g)**, the proposition “ $pcc_D \geq pcc_{j,k}$ for all $j = 1, 2$ and $k = 1, 2, 3$ ” holds. \square

4.3.2. Experimental comparison

Generally, an experimental comparison for evaluating the accuracy of similarity-based information retrieval can be made using an identical benchmark consisting of a certain number of sample pairs and their judged similarities. The greater the Pearson correlation coefficient between the computed and judged similarities, the higher the accuracy is. To evaluate the accuracy of the designed case retrieval algorithm, a specialized tolerance specification case benchmark in the field of computer-aided tolerance specification is firstly required to be designed.

In the previous example, 30 sample case pairs have been randomly selected from the constructed tolerance specification case base containing 100 previous cases, and their judged similarities have been obtained (see Table 3). These 30 sample case pairs and their judged similarities can be directly used as a benchmark to evaluate the accuracy of the design algorithm. To be more specific, they are used to evaluate the accuracies of the measures considering one attribute (Expressions (11)–(13)), the measures considering two attributes (Expressions (14)–(16)), and the measure determined by the former function of the algorithm (Expressions (4)).

Based on the above description, an experiment taking the 30 sample case pairs and their judged similarities in Table 3 as input was carried out by comparing the pcc values of Expressions (11)–(13), Expressions (14)–(16), and Expressions (4). For the purpose of enabling fair comparison, the weights in each expression were computed by the presented weight solving procedure in Section 3. Specifically, the experiment firstly used the weight solving procedure to calculate the values of each group of weights in Expressions (11)–(13), Expressions (14)–(16), and Expressions (4). It then used each expression to compute the similarities of the 30 sample case pairs. Finally, the experiment respectively calculated the Pearson correlation coefficient between each group of the computed similarities and the judged similarities of the 30 sample case pairs. The calculated each group of weights and Pearson correlation coefficient are listed in Table 4.

Table 4

The calculated each group of weights and Pearson correlation coefficient in the comparison experiment.

λ & pcc	$Sim_{1,1}(C_T, C_P)$	$Sim_{1,2}(C_T, C_P)$	$Sim_{1,3}(C_T, C_P)$	$Sim_{2,1}(C_T, C_P)$	$Sim_{2,2}(C_T, C_P)$	$Sim_{2,3}(C_T, C_P)$	$Sim(C_T, C_P)$
$\lambda_{j,k,1} (\lambda_1)$	1.0000	1.0000	1.0000	0.8188	0.5111	0.2963	0.3518
$\lambda_{j,k,2} (\lambda_2)$	————	————	————	0.1812	0.4889	0.7037	0.2267
(λ_3)	————	————	————	————	————	————	0.4215
$pcc_{j,k} (pcc_D)$	0.8839	0.2594	0.8857	0.8902	0.9168	0.9058	0.9279

4.3.3. Comparison result analysis

The theoretical comparison has proved that the measure determined by the former function of the algorithm has the highest accuracy among all of the measures considering one or two features. This is actually a proof of the effectiveness of the weight solving procedure in the designed algorithm since the objective of weight calculation is to maximize the accuracy of this measure. In general similarity calculation, weights are manually assigned by domain experts according to their experience. This does not necessarily achieve the highest accuracy. While in the weight solving procedure, weights can be adjusted to maximize the accuracy of the determined measure. So compared to

manual assignment method, the weight solving procedure has an advantage of constantly obtaining the highest accuracy.

In the experimental comparison, an experiment has been carried out to verify the correctness of the theoretical comparison result and to evaluate the accuracies of all possible linear combinations of $Sim_{Sanchez}(F_{T,T}, F_{T,P})$, $Sim_{Sanchez}(F_{P,T}, F_{P,P})$, and $Sim_{Sanchez}(R_{T,T}, R_{T,P})$. As can be seen from the result of the experiment (Table 4), the proposition “ $pcc_D \geq pcc_{j,k}$ for all $j = 1, 2$ and $k = 1, 2, 3$ ” holds, which experimentally demonstrates the effectiveness of the designed algorithm. In addition, the pcc value of the determined measure can reach 0.9279, which is very close to 1. This indicates that the tolerance specification case retrieval in the designed algorithm highly correlates with the retrieval based on domain experts’ judgement. As can also be seen from Table 4, the measure $Sim_{1,2}(C_T, C_P)$ has a very low pcc value (0.2594), while the pcc values of both $Sim_{1,1}(C_T, C_P)$ and $Sim_{1,3}(C_T, C_P)$ are greater than 0.8000. This signifies that both tolerated feature and topological relation play important role in the design of tolerance specification scheme, while the influence of part features on the design of tolerance specification scheme is relatively small. These are consistent with the actual situation in practice.

In summary, the results and findings of the theoretical and experimental comparisons are as follows: (1) The measure determined by the designed algorithm was proved and verified to be more accurate than all linear combinations of its three components. (2) The weight solving procedure in the designed algorithm has an advantage of constantly obtaining the highest accuracy of tolerance specification case retrieval compared to manual assignment method. (3) The tolerance specification case retrieval in the designed algorithm can highly correlate with the retrieval based on domain experts’ judgement.

5. Conclusion

In this paper, an ontology-supported CBR approach for computer-aided tolerance specification has been proposed. This approach mainly consists of three parts: the representation of tolerance specification cases, the retrieval of tolerance specification cases, and the reuse or revision of tolerance specification cases. The first part used OWL ontology to formalize the past tolerance specification cases as previous cases and to formalize each new tolerance specification problem as a target case. These formalizations can provide a semantic enrichment model for the information of tolerance specification schemes benefiting from the semantic representation capability of OWL ontology. The second part described how to use an ontology-based similarity measure to assess the similarity of tolerated features, the similarity of part features, and the similarity of topological relations and how to linearly combine these similarities to establish an ontology-based similarity measure with high accuracy to implement the retrieval of similar previous cases for a target case. The tolerance specification schemes in some of the retrieved previous cases are valuable resources for designers when designing tolerance specification schemes. The third part clarified the process of the reuse and revision of tolerance specification cases. The paper has also reported the implementation, illustration, and evaluation of the approach. The illustration shows that the approach can provide useful information to assist tolerance specification. The evaluation suggests that the tolerance specification case retrieval in the approach can highly correlate with the retrieval based on domain experts’ judgement.

The main contributions of the paper can be briefly summarized as follows: (1) The paper constructs and develops a tolerance specification ontology to represent the tolerance specification information in CAD systems. This ontology achieves an important change of representing tolerance specification information from a computer-readable level to a

computer-interpretable level. It provides a feasible model of tolerance specification information for further implementing automatic tolerance specification in CAD systems. (2) The paper designs and implements a tolerance specification case retrieval algorithm to retrieve reference schemes for each new tolerance specification problem. This algorithm directly takes the CAD model of an assembly as input and returns reference tolerance specification schemes for each tolerated feature of each part of the assembly as output. This process is fully automatic and does not need know-how. In addition, the algorithm is applicable for all shapes of parts and has involved the generation of a complete tolerance specification scheme.

Future studies will aim especially at overcoming the following two limitations of the ontology-supported CBR approach: (1) The time complexity of the tolerance specification case retrieval algorithm is $O(n)$, which may not be an acceptable upper bound if the number of previous cases n is very large. The ontology-supported CBR approach mainly considers obtaining high case retrieval accuracy. But like case retrieval accuracy, case retrieval speed is also a very important indicator to measure the performance of case retrieval. Hence, it is also of necessity to study the improvement of case retrieval speed when case retrieval accuracy is high enough. (2) The ontology-supported CBR approach mainly considers the influence of geometric information on tolerance specification in tolerance specification case retrieval (i.e. only implements first-order tolerance specification). This is not comprehensive enough since in addition to geometric information, design requirement information and manufacturing process information are other important types of information having influences on tolerance specification (they respectively correspond to second-order and third-order tolerance specifications). Thus, it is of significance to study the synthetic influences of the geometric information, design requirement information, and manufacturing process information on tolerance specification (i.e. study third-order tolerance specification).

Acknowledgements

The authors would like to appreciate the insightful comments from the four anonymous reviewers for the improvement of the paper. The authors also would like to acknowledge the financial supports by the National Natural Science Foundation of China (Nos. 51475190, 51365009, and 51421062), the Hubei Provincial Natural Science Foundation of China (No. 2015CFA109), the Doctoral Dissertation Innovation Foundation of Huazhong University of Science and Technology, and the National Scholarship of China Scholarship Council.

Disclaimer

Certain commercial software product is mentioned in the present paper. It was used only for citation and demonstration purposes. This use does not imply the approval or endorsement by the authors' institutions, nor does it imply that these products are necessarily the best available for the purpose.

References

- [1] ISO 1101 2012. Geometrical Product Specifications (GPS)—Geometrical tolerancing—Tolerances of form, orientation, location and run-out. Geneva: International Organization for Standardization; 2012.
- [2] ASME Y14.5 2009. Dimensioning and tolerancing. New York: American Society of Mechanical Engineers; 2009.
- [3] Armillotta A. A method for computer-aided specification of geometric tolerances. *Comput Aided Des* 2013; 45(12): 1604–1616.
- [4] Zhang Y, Li Z, Gao J, Hong J. New reasoning algorithm for assembly tolerance specifications and corresponding

- tolerance zone types. *Comput Aided Des* 2011; 43(12): 1606–1628.
- [5] Sarigecili MI, Roy U, Rachuri S. Interpreting the semantics of GD&T specifications of a product for tolerance analysis. *Comput Aided Des* 2014; 47(2): 72–84.
 - [6] Richter MM, Weber RO. *Case-Based Reasoning*. Berlin: Springer-Verlag Berlin; 2016.
 - [7] Aamodt A, Plaza E. *Case-based reasoning: Foundational issues, methodological variations, and system approaches*. *AI Commun* 1994; 7(1): 39–59.
 - [8] Staab S, Studer R, editors. *Handbook on ontologies*. 2nd ed. Berlin: Springer-Verlag Berlin Heidelberg; 2009.
 - [9] Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF. *The description logic handbook: Theory, implementation and applications*. 2nd ed. Cambridge: Cambridge University Press; 2010.
 - [10] Amailef K, Lu J. Ontology-supported case-based reasoning approach for intelligent m-Government emergency response services. *Decis Support Syst* 2013; 55(1): 79–97.
 - [11] Armillotta A, Semeraro Q. Geometric tolerance specification. In: Colosimo BM, Senin N, editors. *Geometric tolerances: Impact on product design, quality inspection and statistical process monitoring*. London: Springer London; 2011. p. 3–37.
 - [12] Clement A, Riviere A, Serre P, Valade C. The TTRSs: 13 constraints for dimensioning and tolerancing. In: *Proc 5th CIRP Conf Comput Aided Tolerancing*. 1998.
 - [13] Salomons OW, Haalboom FJ, Jonge Poerink HJ, Van Slooten F, Van Houten FJAM, Kals HJJ. A computer aided tolerancing tool I: Tolerance specification. *Comput Ind* 1996; 31(2): 161–174.
 - [14] Toulorge H, Riviere A, Bellacicco A, Sellakh R. Towards a digital functional assistance process for tolerancing. *ASME Trans J Comput Inf Sci Eng* 2003; 3(1): 39–44.
 - [15] Zhang H, Cao Y, Wei Y, Yang J. A concurrent design method for functional tolerance and structure based on the principle of decomposition and reconstitution. In: *Proc 12th CIRP Conf Comput Aided Tolerancing*. 2012.
 - [16] Shah JJ, Yan Y, Zhang BC. Dimension and tolerance modeling and transformations in feature based design and manufacturing. *J Intell Manuf* 1998; 9(5): 475–488.
 - [17] Kandikjan T, Shah JJ, Davidson JK. A mechanism for validating dimensioning and tolerancing schemes in CAD systems. *Comput Aided Des* 2001; 33(10): 721–737.
 - [18] Wu Y, Shah JJ, Davidson JK. Computer modeling of geometric variations in mechanical parts and assemblies. *ASME Trans J Comput Inf Sci Eng* 2003; 3(1): 54–63.
 - [19] Ballu A, Mathieu L. Choice of functional specifications using graphs within the framework of education. In: *Proc 6th CIRP Conf Comput Aided Tolerancing*. 1999.
 - [20] Dantan JY, Anwer N, Mathieu L. Integrated tolerancing process for conceptual design. *CIRP Ann Manuf Technol* 2002; 52(1): 135–138.
 - [21] Costadoat R, Mathieu L, Falgarone H, Fricero B. Integrated design method based on geometric variations to generate specification since the beginning of the product life-cycle. In: *Proc 11th CIRP Conf Comput Aided Tolerancing*. 2009.
 - [22] Srinivasan V. A geometrical product specification language based on a classification of symmetry group. *Comput Aided Des* 1999; 31(11): 659–668.
 - [23] Haghghi P, Mohan P, Kalish N, Vemulapalli P, Shah JJ, Davidson JK. Toward automatic tolerancing of mechanical assemblies: First-order GD&T schema development and tolerance allocation. *ASME Trans J Comput Inf Sci Eng* 2015; 15(4): 041003.
 - [24] Wang H, Roy U, Sudarsan R, Sriram RD, Lyons KW. Functional tolerancing of a gearbox. In: *Proc North Am Manuf Res Conf*. 2003.
 - [25] Armillotta A, Semeraro Q. Tolerance specification through automated generation of assembly requirements. In: *Proc 10th CIRP Conf Comput Aided Tolerancing*. 2007.
 - [26] Zhong Y, Qin Y, Huang M, Lu W, Gao W, Du Y. Automatically generating assembly tolerance types with an ontology-based approach. *Comput Aided Des* 2013; 45(11): 1253–1275.
 - [27] Qin Y, Zhong Y, Huang M, Liu F. An assembly tolerance representation model based on spatial relations for generating assembly tolerance types. *Proc IMechE Part C: J Mech Eng Sci* 2014; 228(6): 1005–1020.
 - [28] Anselmetti B, Mawussi K. Computer aided tolerancing using positioning features. *ASME Trans J Comput Inf Sci Eng* 2003; 3(1): 15–21.
 - [29] Mejbri H, Anselmetti B, Mawussi K. Functional tolerancing of complex mechanisms: Identification and specification of key parts. *Comput Ind Eng* 2005; 49(2): 241–265.
 - [30] Anselmetti B. Generation of functional tolerancing based on positioning features. *Comput Aided Des* 2006; 38(8): 902–919.
 - [31] Anselmetti B, Chavanne R, Yang J, Anwer N. Quick GPS: A new CAT system for single-part tolerancing. *Com-*

- put Aided Des 2010; 42(9): 768–780.
- [32] Cao Y, Zhang H, Li B, Wu Z, Yang J. Study on functional specification scheme on interface based on positioning features. *Proc IMechE Part B: J Eng Manuf* 2013; 227(5): 745–753.
- [33] Hu J, Xiong G, Wu Z. A variational geometric constraints network for a tolerance types specification. *Int J Adv Manuf Technol* 2004; 24(3–4): 214–222.
- [34] Hu J, Peng Y. Development of a function oriented computer aided tolerancing (FOCAT) system. *Proc IMechE Part B: J Eng Manuf* 2011; 225(7): 1189–1203.
- [35] Zhang Y, Li Z, Xu L, Wang J. A new method for automatic synthesis of tolerances for complex assemblies based on polychromatic sets. *Enterp Inf Syst* 2011; 5(3): 337–358.
- [36] Fernandez-Lopez M, Gomez-Perez A. Overview and analysis of methodologies for building ontologies. *Knowl Eng Rev* 2002; 17(2): 129–156.
- [37] Noy NF, McGuinness DL. *Ontology development 101: A guide to creating your first ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Stanford: Stanford University; 2001.
- [38] Fiorentini X, Gambino I, Liang VC, Fofou S, Rachuri R, Mani M, Bock C. *An Ontology for Assembly Representation*. NIST Interagency/Internal Report 7436. Gaithersburg: National Institute of Standards and Technology; 2007.
- [39] Lu W, Qin Y, Liu X, Huang M, Zhou L, Jiang X. Enriching the semantics of variational geometric constraint data with ontology. *Comput Aided Des* 2015; 63(6): 72–85.
- [40] Ahmed F, Han S. Interoperability of product and manufacturing information using ontology. *Concurrent Eng Res Appl* 2015; 23(3): 265–278.
- [41] Qin Y, Lu W, Liu X, Huang M, Zhou L, Jiang X. Description logic-based automatic generation of geometric tolerance zones. *Int J Adv Manuf Technol* 2015; 79(5–8): 1221–1237.
- [42] Maedche A, Staab S. *Comparing ontologies—Similarity measures and a comparison study*. Institute AIFB Internal Report No. 408. Karlsruhe: University of Karlsruhe; 2001.
- [43] Rodriguez MA, Egenhofer MJ. Determining semantic similarity among entity classes from different ontologies. *IEEE Trans Knowl Data Eng* 2003; 15(2): 442–456.
- [44] Sánchez D, Batet M, Isern D, Valls A. Ontology-based semantic similarity: A new feature-based approach. *Expert Syst Appl* 2012; 39(9): 7718–7728.
- [45] De Mantaras RL, McSherry D, Bridge D, Leake D, Smyth B, Craw S, Faltings B, Maher ML, Cox MT, Forbus K, Keane M, Aamodt A, Watson I. Retrieval, reuse, revision and retention in case-based reasoning. *Knowl Eng Rev* 2005; 20(3): 215–240.
- [46] Johnson RA, Wichern DW. *Applied multivariate statistical analysis*. 6th ed. Upper Saddle River: Pearson Prentice Hall; 2007.



Yuchu Qin is currently a working partner at the EPSRC Centre for Innovative Manufacturing in Advanced Metrology, University of Huddersfield, UK. He received his PhD in Measurement Technology and Instrument from School of Mechanical Science and Engineering, Huazhong University of Science and Technology in 2017. He has a MEng degree in Computer Application Technology and a BEng degree in Computer Science. His research interests lie in computer-aided geometrical product specifications, application of the Semantic Web technologies, and knowledge representation and reasoning. He has published nine papers about representation of the semantics of product and manufacturing information in international journals. He has also co-authored and published one academic monograph about the representation of geometrical product specifications.



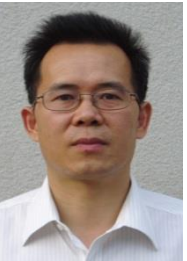
Wenlong Lu is currently an associate professor in School of Mechanical Science and Engineering, Huazhong University of Science and Technology, China. He received his Ph.D. degree from the EPSRC Centre for Innovative Manufacturing in Advanced Metrology at the University of Huddersfield in 2012. His main research interests are focused on semantic interoperability of the information of geometrical product specifications and verification, application of Semantic Web technologies in computer-aided design, and surface optical measurement and instrumentation.



Qunfen Qi is currently a research fellow at the EPSRC Centre for Innovative Manufacturing in Advanced Metrology, University of Huddersfield. She was awarded a PhD degree in Precision Engineering by the University of Huddersfield on 2013. She has an MSc degree in Measurement Technology and a BEng degree in Mechanical Engineering. She focused her research on fundamental structure and knowledge representation of geometrical product specifications. She is also involved in other research projects about surface texture design, computer-aided design, and uncertainties of geometrical product specifications. She has published a number of academic papers in international journals or conference proceedings. She has also co-authored and published four book sections about knowledge representation of surface texture.



Xiaojun Liu is currently a professor and the director of Department of Instrument Science and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, PR China. He received his Ph.D. degree from the Hong Kong University of Science and Technology in 2003, M.Eng. degree from Huazhong University of Science and Technology in 1996, and B.Eng. degree from Harbin Institute of Technology in 1990. His main research interests are in precision engineering, surface measurement theory and techniques, and computer-aided design.



Meifa Huang is currently a professor in School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, China. As well, he currently serves as a doctoral supervisor of Beijing University of Posts and Telecommunications and Guilin University of Electronic Technology. He received his PhD in mechanical manufacture and automation from Huazhong University of Science and Technology in 2004. His main research interests are in computer-aided tolerancing, precision design of mechatronic systems and intelligent measurement method.



Paul J Scott is currently a professor at the EPSRC Centre for Innovative Manufacturing in Advanced Metrology, University of Huddersfield, UK. He received his PhD from the Imperial College London on 1983. His research interests are in geometrical product specifications and verification and foundations of specifying and characterizing products and solutions for real world industrial problems. He is a fellow of the Royal Statistical Society (FRSS), an EPSRC Fellow of Manufacturing, a leading member of ISO TC/213 (Dimensional and geometrical product specifications and verification), a founder member of ISO TC/213 AG1 and AG2, a convenor of ISO TC/213 WG15 and AG12, a core member of BSI TDW4, a convenor of BSI TDW4/-/9, a visiting industrial professor of Taylor Hobson Ltd, and the Taylor Hobson Chair for Computational Geometry.



Dame Xiangqian Jiang is currently the chair professor and the director of the EPSRC Centre for Innovative Manufacturing in Advanced Metrology of the University of Huddersfield and the Royal Academy of Engineering and Renishaw Chair in Precision Metrology. She has a PhD in Surface Metrology and a DSc in Precision Engineering. Her research interests are in surface measurement and precision engineering. She is a Dame Commander of the Order of the British Empire for services to Engineering and Manufacturing (DBE), a fellow of Royal Academy of Engineering (FREng), a fellow of Institute of Engineering Technologies (FIET), a fellow of Royal Society of Arts (FRSA), a fellow of the International Academy of Production Research (FCIRP), a fellow of the International Society for Nano-manufacturing (FISNM), a principle member of ISO/TC 213 (Dimensional and geometrical product specifications and verification) and BSI TW/4, an advisory member for the UK national measurement system, and the UK Chairman of the International Academy of Production Research.