Title:

An Approach to Automatically Selecting Tolerance Types Using Relational Learning for Knowledge Graphs

Authors:
Yingzhong Zhang, zhangyz@dlut.edu.cn, Dalian University of Technology
Jia Jia, 49978527@qq.com, Dalian University of Technology
Mohamed Saad, msaadmahany@mail.dlut.edu.cn, Dalian University of Technology

Keywords:
Tolerance Specification, Tolerance Type, Automatic Selection, Relational Learning, Knowledge Graphs

DOI: 10.14733/cadconfP.2020.358-362

Introduction:

In general, a mechanical product is made by assembling many parts. In order to meet the functional requirements of the product, it is necessary to design tolerance specification for the shape and dimension of the manufactured parts [1]. The tolerance specification does not only affect the quality of products, but also the manufacturing cost and service life of products. As a result, it is an important task to specify reasonable tolerances in the product design. On the other hand, the tolerance specification is a very complex work, which does not only need to consider functional requirements of the product and the manufacturing, measurement and assembling of parts, but also heavily depends on the judgment and experience of designers [12]. In the current practical design process, tolerance specification is usually manually specified by designers, which consumes the labor of designers and affects the design quality. The automatic tolerance specification had been one of challenges in the engineering design automation.

During the past few decades, a number of works have been devoted to the research on tolerance representation and automatic generation of tolerance specification. Tolerance representation is mainly used to organize and represent tolerance information. A number of tolerance representation models have been proposed. The typical tolerance representation models can be classified into: (1) surface and graph models [8], in which the tolerance information is represented by surfaces and graphs; (2) variational geometry models, in which tolerance information is represented by the variations of nominal geometry [8],[15]; (3) structural models, in which the technologically and topologically related surfaces (TTRS) model [4] is the most widely used; and (4) constraint models, in which tolerance information is represented by a finite set of geometric constraints. On the tolerance representation, the tolerance design automation has been paid more and more attention. Many approaches have been proposed to implementing the automatic generation of tolerance types. Zhong et al. [15-16] proposed an ontology-based approach for automatically generating assembly tolerance types, in which a meta-ontology for assembly tolerance representations is constructed. With this meta-ontology, the domain-specific assembly tolerance knowledge can be derived by reusing or inheriting the defined classes or properties. The mapping relations between spatial relations and assembly tolerance types are represented using Semantic Web Rule Language (SWRL). Qin et al. [12] proposed an ontology-supported case-based reasoning (CBR) approach for computer-aided tolerance specification. The presented approach first defines the past tolerance specification problems and their schemes as previous cases and the new tolerance specification problems as target cases, and then reuses previous cases by measuring the similarity between the target case and previous cases.

However, tolerance specification is a very complex work and needs a lot of design knowledge, including explicit knowledge and tacit empirical knowledge. Unfortunately, the tacit design knowledge is implicit, empirical and unstructured, and is very difficult to be captured and formally represented as design rules or design models. Although as mentioned above the research on automatic tolerance specifications had been carried out early, it is still not completely realized in current commercial computer aided-tolerancing (CAT) systems. Although the CBR approach [3],[5-6],[14] provides an effective solution to represent and reuse empirical knowledge, the CBR approach needs to code cases according to certain coding rules, which results in low efficiency and poor flexibility and scalability in the construction of design cases.

Recently, Google presented knowledge graph (KG) [13] technology to enhance its search engine's results. Knowledge graphs model information in the form of entities and relationships between them, and use the W3C Resource Description Framework (RDF) to represent knowledge instances (or facts) in the form of binary relationships, which provides a new feasible solution to formally represent design result facts or design cases.

This paper aims at above issues and challenges and presents an approach to automatic tolerance specification for the product design. The presented approach first captures the tacit knowledge of tolerance specification by constructing tolerance specification knowledge graphs, and then reuses the captured tolerance knowledge to automatically select tolerance types by relational learning for knowledge graphs.

Main Idea:

Tolerance Specification

Currently, the tolerance specifications are usually carried out conformance with the tolerance standards (e.g. ISO 1101 [7], ASME Y14.5 [2]). A tolerance specification scheme mainly consists of four parts as follows:

- Tolerance elements. Tolerance elements are geometric elements applied to tolerance specification scheme, which are classified into four categories: point, line, surface and dimension elements. The surface element includes spherical surface, cylindrical surface, plane, spiral surface, rotating surface, prismatic surface and complex surface [7]. Dimension elements mainly include the ball center, the axis of the cylinder and the center plane of the groove.
- Datum (if necessary). The datum is an ideal feature (ideal point, line or plane) used to constrain the direction and position of tolerance elements. It is required in the design of orientation tolerance, position tolerance and run out tolerance. A tolerance specification may contain one datum, two datums or three datums.
- Tolerance type. Tolerance types are used to describe the geometric characteristics of tolerance elements, including linear dimensional tolerances, angle tolerances, shape tolerances, orientation tolerances, and position tolerances. Shape tolerances include flatness, roundness, cylindricity, profile any line, and profile any surface. Orientation tolerances include parallelism, perpendicularity, inclination, etc. Position tolerances include position degree, concentricity degree, coaxiality, symmetry, etc. In addition, there are also runout tolerance (including circular runout and total runout).
- Tolerance principle (if necessary). When it contains one datum, it is called datum directly; when it contains two datums, it is called first datum and second datum respectively; when it contains three datums, it is called first datum, second datum and third datum respectively.

Overview for the presented approach

From the practical design for tolerance specifications, it can be drawn that selecting tolerance types mainly depends on the geometric characteristics of tolerances elements, constraints from assembling matching, topological relations between tolerances elements. More importantly, the tacit design experience knowledge of designers is a critical factor to select tolerance types. This paper proposes an approach to automatically selecting tolerance types. The main idea is first to capture tacit tolerance specification knowledge (TSK) by constructing its knowledge graphs, and then to reuse the captured TSK to select tolerance types. The tacit knowledge reuse is realized by relational learning for tolerance specification knowledge graphs. The framework of the presented approach is shown as in Fig. 1.

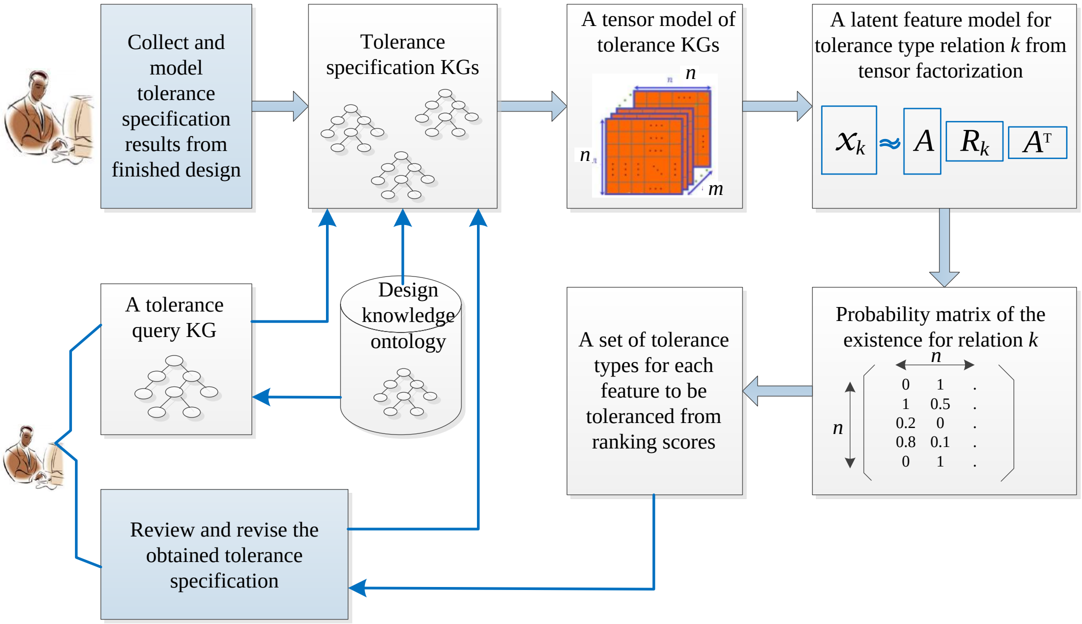

Fig. 1: The framework of the presented approach.

From Fig. 1, it can be seen that this approach includes the following two parts. One is the construction of TSK graphs, which can capture the tacit TSK. The other is to reuse the captured tacit TSK using relational learning. The following mainly outlines the two parts.

TSK Graphs

According to above analysis, TSK mainly involves the following aspects on tolerance specification:

- Tolerance types of the tolerance object.
- Geometric characteristics of the tolerance object.
- Assembling constraints between the tolerance object and its matching parts.
- Topological relations between the tolerance object and other tolerance objects.
- Functional requirements for the tolerance object in the assembly.

In this paper, the TSK is represented as a graph $G=<V, E>$, where V is a set of nodes that represent toleranced object instance or other instances of knowledge entities, and E is a set of directed relation edges that link two nodes. The directed relation can be defined from above five aspects. For example, flatness of tolerance types can be represented as a deviation between the toleranced plane and its ideal plane. The toleranced plane is related to the flatness instance by the relation “hasFlatness”.

In the presented TSK graph, the nodes are labeled as instances of concept classes by ontologies or resources. The TSK base consists of a set of independent graphs: CB = $\{G_1, G_2, \ldots, G_n\}$. Fig. 2 illustrates a part of TSK graphs.

Relational Learning for Tolerance Specification Knowledge Graphs

In statistical relational learning, the representation of an object can contain its relationships to other objects. Relational learning means to acquire undiscovered or missing facts or relationships through statistical modeling for a large number of existing observable data and their relationships. Thus the fact data is usually in the form of a graph, consisting of nodes (entities) and labeled edges (relationships between entities). As provided above, we model the tacit TSK as a set of knowledge graphs. Currently, a large number of learning methods based on distributed representation have been proposed [9], [11]. In this paper, we employ relational approach based on tensor factorization [9-10].

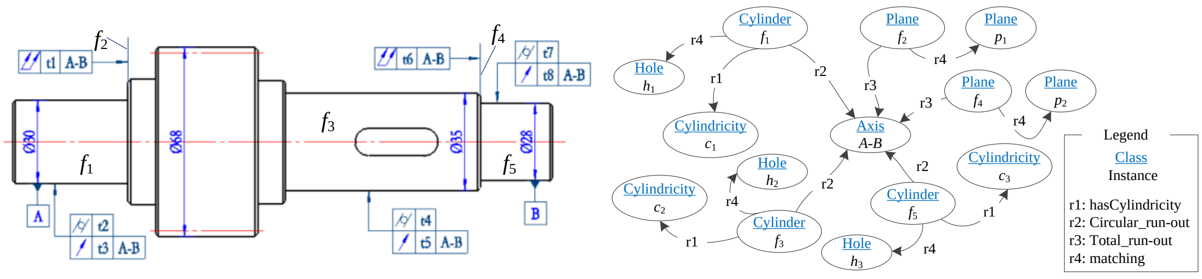

Fig. 2: Illustration for tolerance specification KGs: (a) A shaft part, (b) A part of KGs.

In a knowledge graph, the existence of each possible triple $x_{ijk} = (e_i, r_k, e_j)$ is a binary random variable $y_{ijk} \in \{0, 1\}$. If the triple (e_i, r_k, e_j) exists, $y_{ijk} = 1$, else $y_{ijk} = 0$. As a result, all possible triples in TSK graphs can be grouped naturally in a third-order tensor (three-way array). Tensor factorization can decompose a high-dimensional array into multiple low-dimensional matrices. A three-way tensor $\mathcal{X}$ is employed in which two nodes are identically formed by the concatenated entities of the domain and the third mode which holds the relations. A tensor entry $x_{ijk} = 1$ represents that the fact (e_i, r_k, e_j) exists. If not, for unknown and unseen relations, the entry is set to zero. Then, the triplet score is calculated by the vector obtained through factorization, and the candidate with the high score is selected as the desirable result.

Currently, the RESCAL model [10] is a representative approach for the tensor factorization model. RESCAL decomposes high-dimensional and multi-relational data into a third-order tensor. According to RESCAL, each slice $\mathcal{X}_k$ of tensor $\mathcal{X}$ is factorized as

$$\mathcal{X}_k \approx A R_k A^T, \text{ for } k=1,\dots, m \tag{1}$$

Where A is an $n \times r$ factor matrix that contains the latent-component representation of the entities in the domain and R_k is an asymmetric $r \times r$ matrix that models the interactions of the latent components in the kth relation. The latent components are not directly observed in the data, which reflect the tacit knowledge. The factor matrices A and R_k can be computed by solving the regularized minimization problem [10]. Hence, x_{ijk} can be computed, which is also called as a score to represent entity e_i and entity e_j in the kth relation.

We model each tolerance type as a relation of TSK graphs. When a new tolerance type query is submitted, a TSK graph for the query is constructed, and is appended into the existed knowledge graphs. Then, the score for each type relation is computed. If the value of x_{ijk} in the computation of the kth type relation is greater than a threshold (e.g. 0.9), there is the kth relation between tolerance entity e_i and entity e_j. As a result, we can select a tolerance type from the kth relation as the tolerance type to be queried.

Conclusions:

The tolerance specification design heavily relies on design knowledge, including various kinds of explicit knowledge and tacit knowledge. It is difficult to represent and process tacit design knowledge using traditional approaches based on logic reasoning. In addition, CBR methods lack flexibility and scalability, and have low capturing efficacy. In this paper, a novel approach to automatically selecting tolerance types is presented. The presented approach captures the knowledge of tolerance specification by constructing tolerance specification knowledge graphs, and then reuses the captured tolerance knowledge to automatically select tolerance types by relational learning. Compared with the traditional logic approaches, this approach is a data-driven approach and can combine explicit semantic information and tacit design knowledge. Furthermore, tolerance specification experiments using Python programming have been carried out, which shows that the presented approach is feasible and effective.

Acknowledgements:
This work is supported by the National Science Foundation of China (Grant No. 51775081). The authors thank the anonymous reviewers for their helpful suggestions on this study.

References:

[1] Armillotta, A.: A method for computer-aided specification of geometric tolerances, Computer-Aided Design. 45 (12), 2013, 1604–1616. https://doi.org/10.1016/j.cad.2013.08.007
[2] ASME Y14.5: Dimensioning and tolerancing. American Society of Mechanical Engineers, New York, NY, 2009.
[3] Bejarano, J. C. R.; Coudert, T.; Vareilles, E.; Geneste, L.; Aldanondo, M.; Abeille, J.: Case-based reasoning and system design: an integrated approach based on ontology and preference modeling, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 28(1), 2014, 49–69. https://doi.org/10.1017/S0890060413000498
[4] Clément, A.; Rivière, A.; Serré, P.; Valade, C.: The TTRSs: 13 constraints for dimensioning and tolerancing, Geometric Design Tolerancing: Theories, Standards and Applications. Springer, Boston, MA, 1997. https://doi.org/10.1007/978-1-4615-5797-5_9
[5] El-Sappagh, S. H.; Elmogy, M.: Case based reasoning: Case representation methodologies, International Journal of Advanced Computer Science and Applications, 6(11), 2015, 192-208. https://doi.org/ 10.14569/IJACSA.2015.061126
[6] Farhan, U.; Tolouei-Rad, M.; Osseiran, A.: Indexing and retrieval using case-based reasoning in special purpose machine designs, The International Journal of Advanced Manufacturing Technology, 92(5-8), 2017, 2689–2703. https://doi.org/ 10.1007/s00170-017-0274-5
[7] ISO 1101: Geometrical Product Specifications (GPS)—geometrical tolerancing—tolerances of form, orientation, location and run-out. International Organization for Standardization, Geneva, Switzerland, 2004.
[8] Juster, N. P.: Modeling and representation of dimensions and tolerances: survey. Computer-Aided Design, 24(1), 1992, 3–17. https://doi.org/10.1016/0010-4485(92)90086-P
[9] Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 104(1), 2016, 11–33. https://doi.org/10.1109/JPROC.2015.2483592
[10] Nickel, M.; Tresp, V.; Kriegel, H. P.: A three-way model for collective learning on multi-relational data. Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, Washington, 2011, 809–816.
[11] Padia, A.; Kalpakis, K.; Ferraro, F.; Finin, T.: Knowledge graph fact prediction via knowledge-enriched tensor factorization, Journal of Web Semantics, 59, 2019, 100497. https://doi.org/10.1016/j.websem.2019.01.004
[12] Qin, Y.; Lu, W.; Qi, Q.; Liu, X.; Huang, M.; Scott, P. J.; Jiang, X.: Towards an ontology-supported case-base d reasoning approach for computer-aided tolerance specification, Knowledge-Based Systems, 141, 2018, 129–147. https://doi.org/10.1016/j.knosys.2017.11.013
[13] Singhal, A.: Introducing the Knowledge Graph: Things, Not Strings, 2012, https://www.blog.google/products/search/introducing-knowledge-graph-things-not/
[14] Zhang, Y.; Liu, X.; Jia, J.; Luo, X.: Knowledge representation framework combining case-based reasoning with knowledge graphs for product design, Computer-Aided Design & Applications, 17(4), 2020, 763-782. https://doi.org/10.14733/cadaps.2020.763-782
[15] Zhong, Y.; Qin, Y.; Huang, M.; Lu, W.; Gao, W.; Du, Y.: Automatically generating assembly tolerance types with an ontology-based approach, Computer-Aided Design, 45, 2013, 1253–1275. https://doi.org/10.1016/j.cad.2013.06.006
[16] Zhong, Y.; Qin, Y.; Huang, M.; Lu, W.; Chang, L.: Constructing a meta-model for assembly tolerance types with a description logic-based approach, Computer-Aided Design, 48, 2014, 1–16. https://doi.org/10.1016/j.cad.2013.10.009