Title:
Product Design Knowledge Representation Combining Case-Based Reasoning with Knowledge Graphs

Authors:
Yingzhong Zhang, zhangyz@dlut.edu.cn, Dalian University of Technology
Xu Liu, jhsolid@163.com, Dalian University of Technology
Jia Jia, 49978527@qq.com, Dalian University of Technology

Keywords:
Design Knowledge, Knowledge Representation, Case-Based Reasoning, Knowledge Graph, Ontology

DOI: 10.14733/cadconfP.2019.283-287

Introduction:
Product design relies heavily on design knowledge, including various kinds of explicit knowledge and tacit knowledge [3]. The research on knowledge-based intelligent CAD or expert systems has always been attracted. A number of knowledge-based CAD systems have been successfully applied. However, as most design knowledge is implicit, empirical and unstructured, it is very difficult to be captured and formally represented as rules or models. Meanwhile, in general design activities, it is widely accepted that designer's practices rely heavily on their past design experiences, instead of designing everything from scratch. Case based reasoning (CBR) methodology is to solve new problems by adapting previous successful solutions to similar current problems. CBR is similar to human beings' reasoning processes, which can learn over time, and reason with concepts that have not been fully defined or modeled. Aamodt and Plaza [1] described CBR typically as a cyclical process comprising four steps of retrieval, reuse, revising, and retaining.

Since CBR methodology was proposed it has been researched intensively. A large number of applications have demonstrated CBR is a promising methodology for the knowledge-based product design. Lee and Luo [9] presented a study on die-casting die design based on CBR. Hashemi et al. [7] proposed a case-based reasoning approach for design of machining fixture. In the CBR method, the retrieval of cases is the most crucial step. In practice, it is not enough to retrieve cases based on a match of keywords, and it needs to understand the really meaning of query problems and to retrieve semantically matched cases with query problems, which requires a conceptualized case representation for different level cases and semantic retrieval that can understand design intent. In addition, sharing and common understanding for cases, and scalability and flexibility of the case base are still issues to be solved.

In order to address above issues in the traditional CBR method, ontology has been introduced into the CBR methodology. Gao and Deng [5] presented a strategy that domain ontology is constructed as the basis of knowledge structure and the concrete case knowledge is represented as the instance of relevant concept of domain ontology, through which knowledge of different systems is shared. Guo et al. [6] proposed an intelligent retrieval method by integrating ontology technology into CBR systems to design new injection molds. Bejarano et al. [2] proposed an integrated CBR approach based on ontology and preference modeling. Chen et al. [4] proposed an ontology and CBR based automated decision-making method for the disassembly of mechanical products. From the published literature, there are the following shortcomings: (i) lack of high-level conceptual models that can reflect the essence and processes of product design as most of research focuses on ontology modeling of a specific product design; (ii) lack of semantic hierarchy (or granularity) representation of design cases;

and (iii) low efficiency and poor flexibility in the design case construction because case representation needs a lot of case analysis on design problems and solutions.

Recently, Google presented a knowledge base using knowledge graph (KG) technology to enhance its search engine's results with information gathered from a variety of sources [11]. 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 is represented as a particular triple consisting of subject, predicate, and object. All triples together are formed a graph, where nodes represent entities, and directed edges represent relationships. As a result, knowledge graphs can provide much more powerful capabilities on semantic representation and knowledge retrieval. In fact, the instances in knowledge graphs can be used to constitute different level cases in CBR. Hence, combining CBR with knowledge graphs to represent design knowledge will be a feasible solution.

This paper aims at above issues and challenges, focuses on the design knowledge representation of general mechanical products, and presents a design knowledge representation framework combining CBR with knowledge graphs. The main contribution includes using knowledge graphs to represent design cases and presenting and presenting a novel design case retrieval method.

Main Idea:

A product design knowledge representation framework combining CBR with knowledge graphs

According to requirements of design knowledge representation and characteristics of CBR and knowledge graphs, we propose a product design knowledge representation framework combining CBR with knowledge graphs, which is as shown in Fig. 1. This framework consists of a knowledge representation layer and a knowledge operation layer as follows:

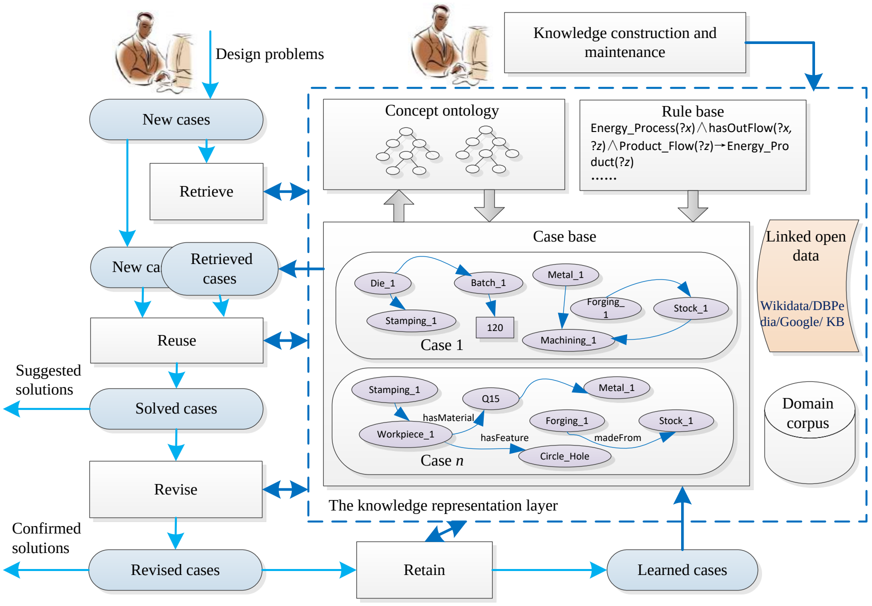

Fig. 1: A design knowledge representation framework combining CBR with knowledge graphs.

- Knowledge representation layer

The knowledge representation layer is the core of the whole framework. It includes three bases: a concept ontology base which provides a common sharing and explicit description for all concepts in the knowledge base, a rule base where the causal design knowledge is defined as rules, and a case base where design cases are represented as a set of directed knowledge graphs.

- Knowledge operation layer

The knowledge operation layer is outside the knowledge representation layer, which includes two kinds of knowledge operations. One is the knowledge construction and maintenance operation, which is performed by a human-computer interface to construct the domain ontology, the rule base, and the case base. The other is a set of typical CBR application operations, which include retrieving, reuse, revising and retaining of design cases. These CBR application operations form a CBR cycle.

Semantic representation for design cases using knowledge graphs

A concept is an abstract form to represent knowledge entities and relationships. Ontology can provide a formal and sharing semantic representation for various concepts. In the CBR method, a case consists of one knowledge entity or multiple knowledge entities linked by relations. It is necessary to abstract common properties of all cases in a specific design domain and to define them as high-level domain ontologies. With the defined domain ontologies, constructing new design cases can be guided and performed with less ambiguity. In addition, the defined domain ontologies can help case retrieval.

Case representation is the foundation of a CBR system as it directly affects the retrieval, reuse, and retain of cases. According to traditional CBR terminology, a case usually denotes a problem situation [1]. In the most current research, the CBR problem situation is usually represented as a set of problem–solution tuples or problem–solution–evaluation triples [8]. However, there are the following issues: (i) a complex design problem usually needs to be decomposed into a hierarchy of sub-problems, which requires a formal structure to represent the problems and the relations between them; (ii) it is difficult or not an easy task for a designer to extract and define the design problems and the associated solutions, which results in poor efficiency of design case construction; and (iii) in the design process, many design requirements and problems are not well identified, so it is difficult to carry out comparisons with feature-value pairs of the problem–solution.

In the light of above issues, we propose an approach to represent design cases using knowledge graphs. The following gives a definition on the design case.

Definition 1. A design case is a set of factual instances interrelated by design results and relations between them on a topic within a product design. The topic determines the level and granularity of a design case. A design case is represented as a design knowledge graph or a segment of the graph.

A design case is a graph $G=<V, E>$, where V is a set of design instance nodes and E is a set of directed relation edges that link two instances. The design instances are usually labeled by ontologies or resources. The design case base consists of a set of independent graphs: $CB=\{G_1, G_2, \ldots, G_n\}$.

Using knowledge graphs to represent design case has the following characteristics:

- Design cases are classified as explicit design cases and implicit design cases.
- The design case graph can represent the hierarchy of design cases.
- The design case construction only needs to record the final design results of a product under the ontological guidance.

It is easier and more straightforward to construct design cases from recording the design results than from extracting design problems and solutions. The problem and solution are implicitly embedded in the semantics of relations between instances. As a result, the CBR system is much more flexible and scalable. The design case base is much easier to be constructed.

Combining case reasoning and description logic reasoning

Currently, the symbolic rule is still one of the most popular knowledge representation and reasoning methods. Description logic has much better reasoning capabilities, especially the inference of implicit knowledge from explicit knowledge contained in the design case base. In the presented approach, ontologies and knowledge graphs are employed to represent design cases. The foundation of knowledge graphs is description logic. Hence, it is very necessary to combine case reasoning and description logic reasoning.

In the design case graph, the design problems and solutions are not explicitly defined but implicitly described in the various relations, which requires a powerful semantic reasoning mechanism to implement design case retrieval from knowledge graphs accorder to design queries.

Retrieving design cases from design knowledge graphs

In the CBR cycle, an important step is to retrieve previous cases that can be used to solve the new target problem. In this paper, the design cases are represented as a set of KGs and the design problems are not predefined. This method can represent and retrieve design cases with much more flexibility and scalability. Correspondingly, we propose a new method to retrieve design cases as shown in Fig. 2. The presented method mainly includes the following procedures:

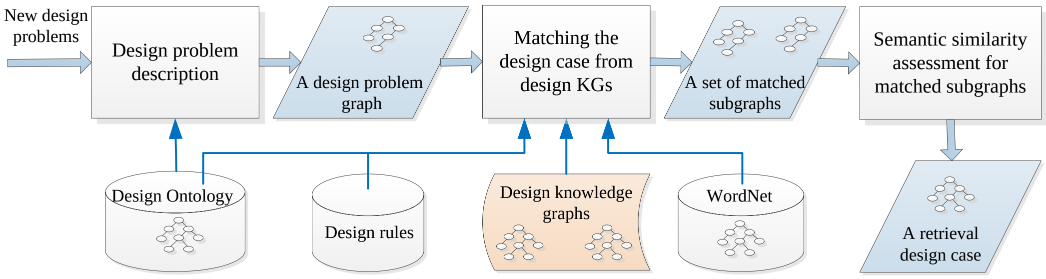

Fig. 2: Overview of retrieving a design case from KGs.

- Design problem description

In general, a complex design problem involves multiple design requirements and constraints. Hence, it is very difficult to use keywords to describe a complex design problem. In this paper, we decompose a design problem into multiple simple design problems consisting of one concept and one relation, which can form a problem query graph. Then, the problem query graph is converted into a design query statement represented by first-order logic.

- Matching the design problem graph with design KGs

In this procedure, the constructed problem query graph is used to match design KGs in order to find similar design cases. The match is a process to search an isomorphic subgraph to the problem query graph from KGs. If the match is successful, the whole graph where the matched sub-graph locates is the desired design case.

First, we convert the retrieval of design cases into an issue on semantic query from design KGs. However, using above SPARQL pattern to query design case from design KGs, the query results may be disappointment. For a new design, there is no such a subgraph that is completely consistent with the problem query graph in previous case graphs. Based on referring SQWRL [10] and under the Closed-World Assumption, we developed a semantic query approach [12] like SQWRL using backward chained reasoning. In this paper, we use the semantic query approach to obtain a set of similar design cases.

- Semantic similarity assessment

In this procedure, the design cases with low similarity obtained in above steps need to be further carried out semantic similarity assessment in order to retrieve the most similar design case from the case candidates. The semantic similarity assessment is implemented between the problem query graph and isomorphic subgraphs in the case candidate, and is performed according to the following strategies:

(1) Semantic similarity measurement for concepts.

Using the path-based method, the semantic similarity of the concept pair can be calculated, which can be used as the semantic similarity between two instances labeled by the concept pair.

(2) Numerical similarity measurement for attributes.

In the engineering design, the numerical value of attributes affects the selection of cases. We perform the similarity measurement for certainty numerical values, interval values and fuzzy numerical values.

(3) Ranking matched subgraphs based on similarity assessment

Semantic similarity assessment between two graphs is relatively difficult. We propose an approach to ranking subgraphs by combing node similarity and node weight, which can obtain the most similar design case.

Obviously, the similarity of two graphs relies on the similarity of their nodes and their graph structures. Hence, we can calculate the similarity of nodes in the two graphs to be compared according to above methods respectively. The computed similarity result also forms a hierarchical graph same as the matched subgraph.

The similarity of nodes located on different levels will have different contributions to the similarity of the whole graph. Specifically, the similarity contribution of nodes on the top level is greater than on the bottom level. In addition, the different nodes may have different connecting edges; obviously, the node that has more out connecting edges has much more influences on the similarity to the compared graph. Hence, we introduce the node weight to represent the similarity influence for each node.

Conclusions:

CBR methodology can provide a feasible and effective solution to represent complex empirical and unstructured design knowledge. In the light of issues and challenges existing in the current CBR research, this paper presents a design knowledge representation framework combining CBR with knowledge graphs. In this framework, the design cases are represented as a set of design knowledge graphs. A novel approach to retrieving design cases from knowledge graphs is presented, in which new design problem are represented as a RDF graph instead of retrieval keywords and the design case retrieval is converted a semantic relational query. The construction of design cases only needs to record the final design results under ontological guidance. As a result, it is much more straightforward, flexible and scalable to construct design case bases than current CBR methods.

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] Aamodt, A.; Plaza, E.: Case based reasoning: foundational issues, methodological variations, and system approaches, AI Communications, 7(1), 1994, 39-59.

[2] 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. Artif Intell Eng Des Anal Manuf, 28(1), 2014, 49–69.

[3] Chandrasegaran, S. K.; Ramani, K.; Sriram, R. D.; Horváth, I.; Bernard, A.; Harik, R. F.; Gao, W.: The evolution, challenges, and future of knowledge representation in product design systems, Computer-Aided Design, 45(2), 2013, 204-228. https://doi.org/10.1016/j.cad.2012.08.006.

[4] Chen, S.; Yi, J.; Jiang, H.; Zhu, X.: Ontology and CBR based automated decision-making method for the disassembly of mechanical products, Advanced Engineering Informatics, 30(3), 2016, 564-584. https://doi.org/10.1016/j.aei.2016.06.005

[5] Gao, J.; Deng, G.: The research of applying domain ontology to case based reasoning system, Proceedings of international conference on services systems and services management, Chongqing, China, 2005. https://doi.org/10.1109/ICSSSM.2005.1500169

[6] Guo, Y.; Hu, J.; Peng, Y.: A CBR system for injection mould design based on ontology: a case study, Computer-Aided Design, 44 (6), 2012, 496-508. https://doi.org/10.1016/j.cad.2011.12.007

[7] Hashemi, H.; Shaharoun, A. M.; Sudin, I.: A case-based reasoning approach for design of machining fixture, The International Journal of Advanced Manufacturing Technology, 74, 2014, 113-124. https://doi.org/10.1007/s00170-014-5930-4

[8] Hu, J.; Qi, J.; Peng, Y.: New CBR adaptation method combining with problem–solution relational analysis for mechanical design, Computers in Industry, 66, 2015, 41–51. https://doi.org/10.1016/j.compind.2014.08.004

[9] Lee, K. S.; Luo, C.: Application of case-based reasoning in die-casting die design, The International Journal of Advanced Manufacturing Technology, 20, 2002, 284–295.

[10] O'Connor, M.; Das, A.: SQWRL: A Query Language for OWL, OWLED'09 Proceedings of the 6th International Conference on OWL, Chantilly, VA, 2009, 208-215.

[11] Singhal, A.: Introducing the knowledge graph: things, not strings, 2012, http://googleblog.blogspot.com/2012/05/introducingknowledge-graph-things-not.html

[12] Zhang, Y.; Luo, X.; Zhang, B.; Zhang S.: Semantic approach to the automatic recognition of features, The International Journal of Advanced Manufacturing Technology, 89(1–4), 2017, 417–37. https://doi.org/10.1007/s00170-016-9056-8