

A New Design Knowledge Retrieval Model Based on Granularity and Clustering Theories

Journal:	<i>Part C: Journal of Mechanical Engineering Science</i>
Manuscript ID	JMES-16-0089.R1
Manuscript Type:	Original article
Date Submitted by the Author:	16-Jul-2016
Complete List of Authors:	Liu, Hongwei; Sichuan University, School of Manufacturing Science and Engineering Li, Yan; Sichuan University, School of Manufacturing Science and Engineering Chen, Jian; University of Southampton, Institute of Sound and Vibration Research Xiong, Yan; Sichuan University, School of Manufacturing Science and Engineering Du, Xiaojiao; Sichuan University, School of Manufacturing Science and Engineering Shi, Qian; Sichuan University, School of Manufacturing Science and Engineering
Keywords:	product innovative design, knowledge retrieval, Ontology, granularity, clustering
Abstract:	Former research has proved that the design knowledge involved in the conceptual design stage has multiple attributes/perspectives, which can be further abstracted into the concepts with different abstract/ granularity levels, from coarse to fine. Based on this, we believe that it is necessary to use multiple attributes to describe design knowledge during the knowledge retrieval process. Firstly, an attribute ontology with multi-perspective and multi-granularity was built in the new retrieval model. Therefore, the knowledge documents can be abstracted by the concepts of the ontology. Based on the correlation of the concepts, the clustering theory is introduced into the new model to cluster knowledge documents and the combined keyword concepts of the clusters are generated as well. The keyword concepts of the clusters, rather than the keyword concepts of documents, are used as the index of the retrieval. During the knowledge matching process, the keywords and their semantic extension of design problems are extracted, and the similarity between the abstract descriptions of the knowledge clusters and design problems is calculated, so the best cluster can be found by the calculated results. Based on the granularity levels of the keywords (concepts) the design problems, finally the documents in the selected clustered are ranked in the order of granularity levels. The selected document by the model is that with the highest relevance and most suitable granularity level about the design problem. In the last section of the paper, we used a simple real case to evaluate the new design model, also developed the scheme of the whole design knowledge retrieval system for the future work.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

SCHOLARONE™
Manuscripts

For Peer Review

Title: A New Design Knowledge Retrieval Model Based on Granularity and Clustering Theories

Authors: Hongwei Liu¹, Yan Li¹, Jian Chen², Yan Xiong¹, Xiaojiao Du¹, Qian Shi¹.

Email:scu_liuhongwei@163.com; liyan@scu.edu.cn; jc2d14@soton.ac.uk; xy@scu.edu.cn;

1095719250@qq.com; 1571979875@qq.com.

Address: ¹School of Manufacturing Science and Engineering, Sichuan University, NanYihuan Road No.24, WuHou District, Chengdu 610065, Sichuan Province, China.

²Faculty of Engineering and the Environment, University of Southampton, Southampton, SO17 1BJ, UK.

Institution: ¹School of Manufacturing Science and Engineering, Sichuan University.

²Faculty of Engineering and the Environment, University of Southampton.

Corresponding author: Yan Li, Professor, School of Manufacturing Science and Engineering, Sichuan University, NanYihuan Road No.24, WuHou District, Chengdu, 610065, China;
Tel: 86-28-85403211; Fax: 86-28-85403211; liyan@scu.edu.cn.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

A New Design Knowledge Retrieval Model Based on Granularity and Clustering Theories

Hongwei LIU¹, Yan LI^{1*}, Jian CHEN², Yan XIONG¹, Xiaojiao DU¹, Qian SHI¹

Abstract: Former research has proved that the design knowledge involved in the conceptual design stage has multiple attributes/perspectives, which can be further abstracted into the concepts with different abstract/ granularity levels, from coarse to fine. Based on this, we believe that it is necessary to use multiple attributes to describe design knowledge during the knowledge retrieval process. Firstly, an attribute ontology with multi-perspective and multi-granularity was built in the new retrieval model. Therefore, the knowledge documents can be abstracted by the concepts of the ontology. Based on the correlation of the concepts, the clustering theory is introduced into the new model to cluster knowledge documents and the combined keyword concepts of the clusters are generated as well. The keyword concepts of the clusters, rather than the keyword concepts of documents, are used as the index of the retrieval. During the knowledge matching process, the keywords and their semantic extension of design problems are extracted, and the similarity between the abstract descriptions of the knowledge clusters and design problems is calculated, so the best cluster can be found by the calculated results. Based on the granularity levels of the keywords (concepts) the design problems, finally the documents in the selected clustered are ranked in the order of granularity levels. The selected document by the model is that with the highest relevance and most suitable granularity level about the design problem. In the last section of the paper, we used a simple real case to evaluate the new design model, also developed the scheme of the whole design knowledge retrieval system for the future work.

Keywords: product innovative design, knowledge retrieval, ontology, granularity, clustering

1. Introduction

Knowledge plays an important role in the product innovative design process. Sufficient knowledge resource is the critical to prompt the innovation designs in modern companies [1]. Since the conceptual design stage determines most of the innovation elements of new products, if product designers can efficiently acquire the relevant knowledge during the conceptual design stage, the innovation levels of the designed products will have significant improvements [2]. More importantly, the knowledge retrieval system of the conceptual design process should inspire the designers in the design thinking to generate innovative solutions from various aspects, such as customer needs, product functions, product structures, abstract principles, etc [3]. Therefore, the key issue in the knowledge-based conceptual design stage is to discover the multi-perspective correlation between design problems and existing knowledge. In order to solve this issue, it is very necessary to study the design knowledge from multiple perspectives and build its multi-perspective correlation. Clearer and more comprehensive knowledge correlation can efficiently help the designers to realize the knowledge

¹School of Manufacturing Science and Engineering, Sichuan University, Chengdu 610065, Sichuan Province, China.

² Faculty of Engineering and the Environment, University of Southampton, Southampton, SO17 1BJ, UK.

***Corresponding author:**

Yan Li, School of Manufacturing Science and Engineering, Sichuan University, NanYihuan Road No.24, WuHou District,Chengdu 610065, Sichuan Province, China. Tel: 86-28-85403211.

E-mail: liyan@scu.edu.cn

transfer [4] and knowledge recombination [5]. Up to now, many methods (or models) have been developed to support the product conceptual innovative design, such as the theory of inventive problems solving (TRIZ) [6], the function-behaviour-structure model [7], and the axiomatic design method (AD) [8]. Based on these methods, many design knowledge perspectives can be extracted, such as, function [9], behaviour-flow [10], physical effects [11], inventive principles [12] and technical principles [13], etc.

Most of the commercial Computer Aided Innovation (CAI) systems, such as Goldfire [14], CREAX Innovator Suite [15] and Pro/Innovator [16], can only support the function-based [17] knowledge retrieval (only correlate the knowledge from function perspective); therefore, they can only retrieve the relevant knowledge when the designers obtain the functional attribute of the design problems. This situation hinders the stimulation of innovative thinking. More recently, some researchers [18, 19] have made some improvements on the knowledge retrieval model by introducing both the function-based and TRIZ-based knowledge retrieval methods into the CAI systems. Compared with the single function-perspective based system, both Refs [18, 19] have verified that the improved systems (with two perspectives) can retrieve more relevant knowledge to support the innovation conceptual design. However, the improved system still have the limitation that it can only retrieve the relevant knowledge based on defined functional attribute or TRIZ principle attribute. If the designers need solve innovative design problems from the effect knowledge perspective or behaviour knowledge perspective, the improved systems may provide the designers the inappropriate/inaccurate function-based knowledge or TRIZ principle based knowledge. Due to the shortcomings of these existing retrieval models, we believe that the new knowledge retrieval system should include more perspectives to represent the knowledge and its correlation.

As mentioned above, another important issue of design knowledge retrieval is how to realize the multi-perspective correlation of design knowledge, also the correlation between design problems and existing knowledge. Ontology is a common method to build the correlation of general knowledge; therefore, we believe that ontology should be also a promising method for the correlation of design knowledge. In recent years, some researchers have applied the granularity theory in the general knowledge retrieval model [20-22]. This application is still in the infant stage, but these existing works can provide a useful guide for the application of ontology on design knowledge retrieval. Ontology is a method to describe the entities by a set of representational terms [23], and it is a specification of a conceptualization of the entities/subjects [24]. Ontology can correlate the entities by means of the semantic correlation of the defined concepts. Ontology based knowledge correlation models normally include two main aspects, one is the abstraction of knowledge (conceptualization process), and the other is the correlation of knowledge. However, most existing ontology-based methods [25-27] for design knowledge retrieval only focused on the correlation of concrete knowledge, such as customer needs, product components, etc., and researchers paid limited attention on the correlation of the concepts of design knowledge for the conceptual design stage.

Granularity is a common way to represent and measure the abstract degree of knowledge [28], and it is also a easy way to represent the levels of the concepts that extracted from design problems. Well-trained designers often solve problems from a coarse granularity level to a fine granularity level [29, 30]. Moreover, knowledge granularity is considered as the foundation to realize knowledge transferring and knowledge reuse [31]. Up to now, only a very few of researchers have attempted to abstract the design knowledge based on the granularity theory. Both Refs. [32, 33] built a granular computing model to retrieve the knowledge with different abstract levels, and verified that granularity theory could efficiently help the designers in the conceptual innovation design process. On the other hand, the existing commercial CAI systems aforementioned have no relevant function/model to conceptualizing design knowledge according to the

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

granularity levels. Therefore, there is still much work to be done on the granularity-based design knowledge conceptualization.

As analysed above, the design knowledge retrieval model with multi-perspective and multi-granularity should has a capability to retrieve high-quality knowledge (with high inspiration for innovation) to solve the design problems, but on the other hand, there has been a trend that the conceptual innovation design of modern products involve more and more knowledge recently (in quantity aspect), so how to organize a huge amount of design knowledge is also very important. Clustering theory should be a potential solution to this problem. Clustering can assign labels to large-scale unlabelled data and classify them into different groups or subsets such that the data in each group share some similarity [34]. So clustering can indirectly reduce the management and computing scale of data. In recent years, clustering theory has been introduced into the knowledge retrieval [35-38]. Particularly, for the knowledge-based conceptual product design, some scholars [39-41] revealed that clustering can help the designers to accurately retrieve sufficient relevant design knowledge from a large amount of existing knowledge, consequently expand (from quantity aspect) the operation space for the knowledge transferring and recombination.

From the above analysis on the knowledge requirements of the conceptual design process, this paper proposed a new design knowledge retrieval model based on the granularity theory and clustering theory. The model derives from the conceptualization and correlation of design knowledge, and it provides a practicable approach to retrieve design knowledge. Meanwhile, the model integrates the advantage of clustering on knowledge's classification, so the designer can search much broader range of relevant knowledge during the problem solving process. Based on the theories of multiple-perspective, clustering and granularity, a new knowledge matching method was proposed in this model. An attribute ontology was built from the extracted multiple attributes (i.e. corresponding perspectives) of design knowledge and the different granularity levels of these attributes. The ontology works as the semantic basis for the following clustering process and matching process. The model selects the most relevant knowledge cluster (with multiple attributes) in the first step of the matching process. Next, the knowledge documents in the selected the cluster are ranked according to their granularity levels. Finally the model provides the designers the knowledge document with the most relevance and suitable granularity level about the design problems. In the last section of this paper, a case study was carried out to validate the feasibility and performance of the new developed model, and the final results showed that the model can efficiently help the designers to retrieve more relevant knowledge which has the multiple-correlation with the design problem, and can also fulfil the designers' requirement on the knowledge with the suitable granularity levels.

2. Overall scheme of new design knowledge retrieval model

The schematic diagram of the new granularity and clustering based knowledge retrieval model for innovative conceptual design is shown in Figure 1. The main part of the model is a “attribute ontology-clustering index-knowledge resource” three-layer mapping structure, which is used to represent various design knowledge. Based on the attributes extracted from design knowledge and the granularity levels of attributes, we set up an attribute ontology which can provide the common understanding on innovation design. Meanwhile, the WordNet semantically extends the attribute ontology into the *semantic space*. The knowledge resource layer consists of knowledge documents, which are mainly from local archives and internet and saved beforehand in the database by some standard formats. It is worth pointing out that the knowledge documents can be considered as a hidden input of the model, but the processing (searching, processing and saving) of these documents has been finished before the practical retrieval process of the designers, therefore this input is not a real “dynamic” input of the new retrieval model. These documents are abstracted by the conceptual terminologies in the semantic space, and we called these the conceptual terminologies as keyword concepts. The model also allows the

knowledge in the knowledge resource layer to be clustered into different clusters, and consequently the *semantic space* generates the features for the different knowledge clusters. The features for clustering can function as the clustering index in the knowledge matching process.

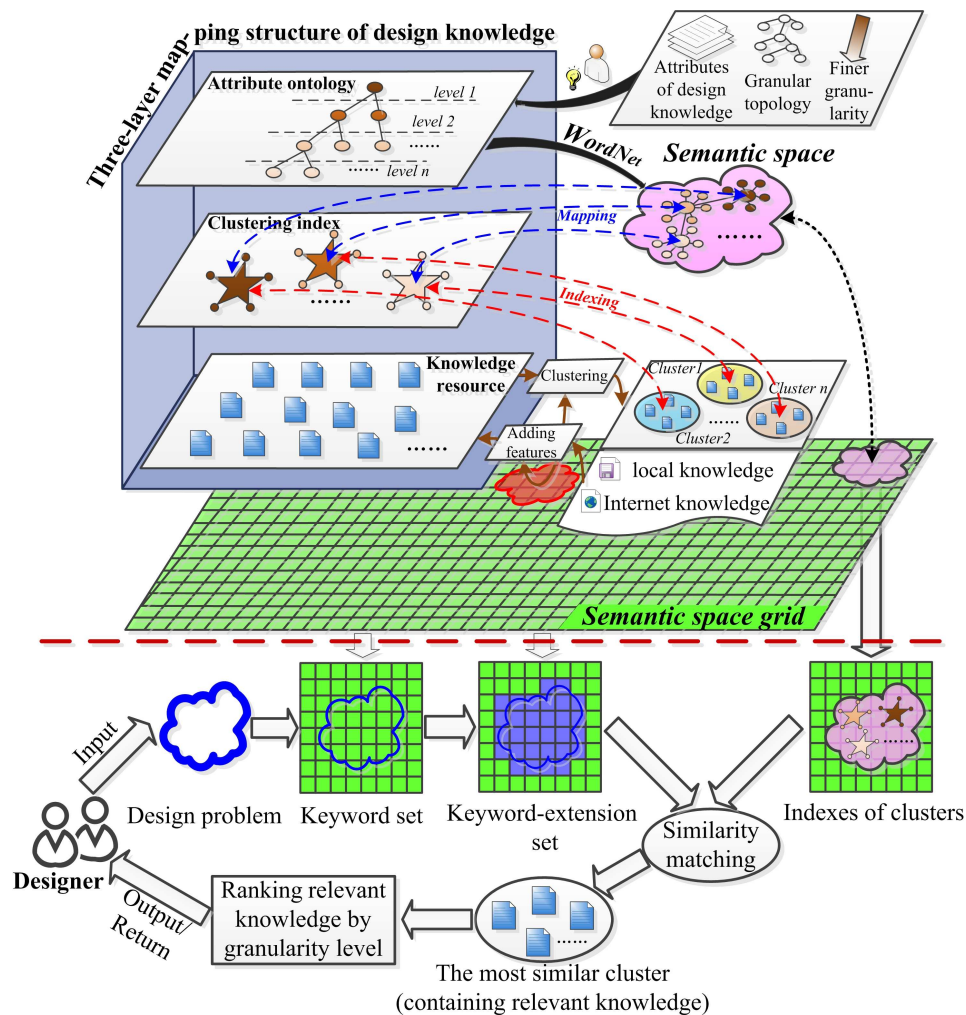


Figure 1. Schematic diagram of proposed design knowledge retrieval model

The diagram of knowledge matching is in the bottom section of Figure 1. The design problems, as the real input of the model, are semantically analysed by the conceptual terminologies in the *semantic space*, then their keyword-extension sets are generated. Next, the model properly matches the knowledge clusters with design problems based on the calculated semantic similarity between them. Finally, the knowledge documents in the cluster, which has the highest similarity level, are ranked based on their granularity levels, and they are returned to the designers to support the design problem solving.

It can be seen that the building of ontology with granularity hierarchy and the generation of keyword concepts and clustering index are two critical elements in the new knowledge retrieve model. Based on these two elements, the matching between the design problems and knowledge becomes more straightforward. The details of the new model and the involved techniques will be specified in the following sections.

3 Two critical elements in the design knowledge retrieval model

3.1 Attribute ontology

As mentioned above, ontology is a very comprehensive technique to represent design knowledge, and it can efficiently correlate design knowledge from multiple perspectives and interpret the abstract (conceptual) levels of the design knowledge as well. Ontology defines the concepts, terminologies, and relations of design knowledge, so it is the semantic basis of knowledge organization and retrieval. In this paper, a quadruple ontology $Ont = \{ontC\ ontA\ ontR\ ontH\}$ is defined. C refers to the concepts extracted from *design knowledge*. A denotes the attributes (perspectives) (in this paper, they are derived (extracted) from the summary of our group’s previous research about knowledge representation for innovative design [32]), including TRIZ principle, function, behaviour-flow, effect, and domain (we consider that these five attributes represent five main perspectives). R denotes the relations between the attributes, such as “*is-a*”, “*instance-of*”, “*sub-class-of*”, “*kind-of*”, etc, and H refers to the granularity hierarchy of the ontology.

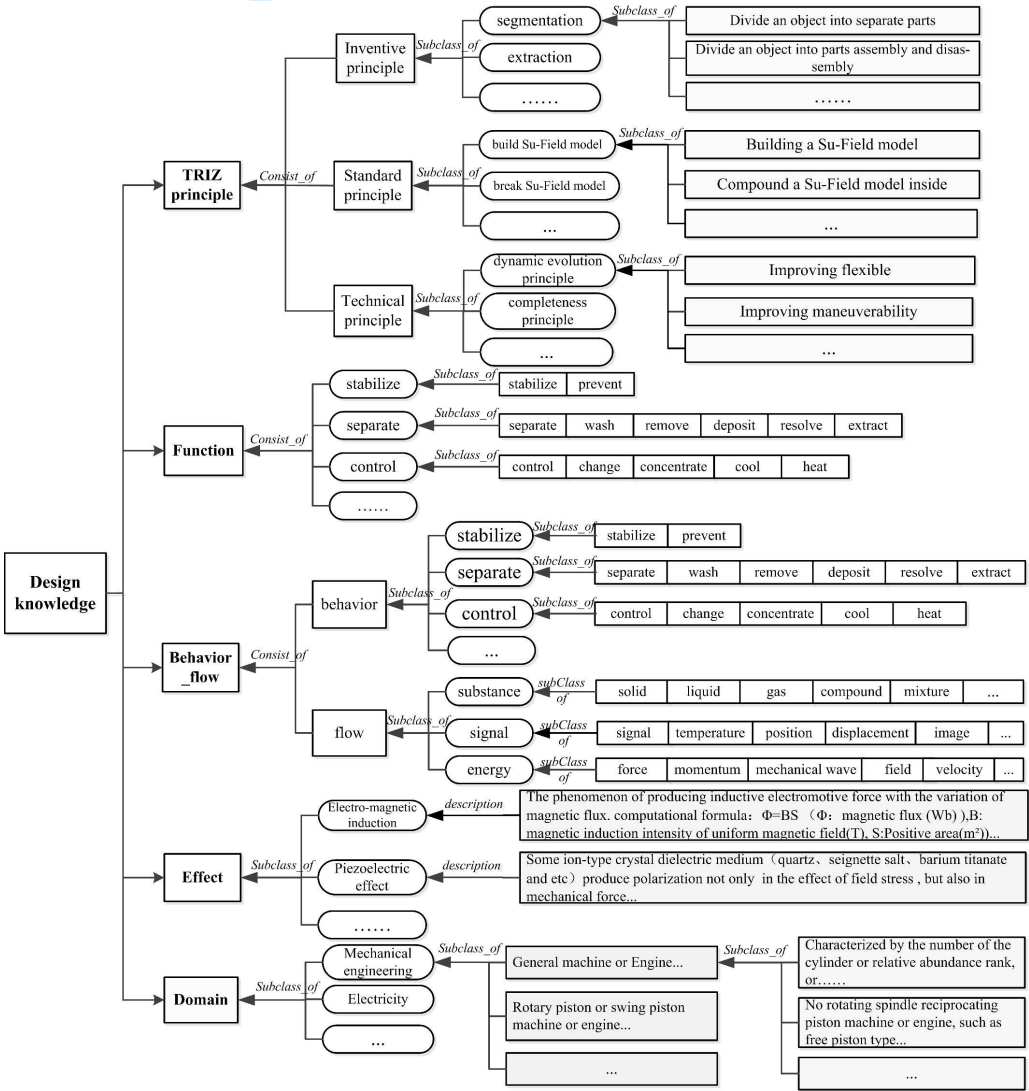


Figure 2. Hierarchical structure of attribute ontology

(1) *Granularity hierarchy of the attributes* From Figure 2, we can see that the ontology in this model starts from the five attributes (perspective), which are the coarsest concepts and also in top granularity level of the hierarchical tree. All five top concepts have secondary concepts, which can be further divided into finer granularity concepts. For example, the secondary concepts of the TRIZ principle include inventive principle, standard principle and technical principle. Moreover, the inventive principle is divided into 40 inventive principles, the standard principle is divided into 76 standard solutions, and technical principle is divided into 11 technology evolution principles; additionally, all of the 127 principles can be further divided into detailed concept descriptions with the finest granularity.

(2) *Relations between different attributes* As shown in Figure 3, the attributes relating to function include the behaviour-flow and effect which realize this function, and the domain that the functional entity of this function belongs to. The attributes relating to TRIZ principle include the domain that the knowledge entity of this TRIZ principle belongs to, and the effect implementing this TRIZ principle. The attributes relating to behaviour-flow include the function realized by this behaviour process, and the TRIZ principle knowledge and domain information involved in this behaviour. The attributes relating to effect include the function that realizes this effect, and the corresponding behaviour-flow knowledge and domain information involved in this effect. Finally, relational terms, such as “referring_”, “achieved_”, “_input/output”, etc, are used to describe the semantic relationships among these attributes. The configuration of the semantic relationships of the ontology can be illustrated by the Ontology Web Language (OWL), and is shown in Figure 3.

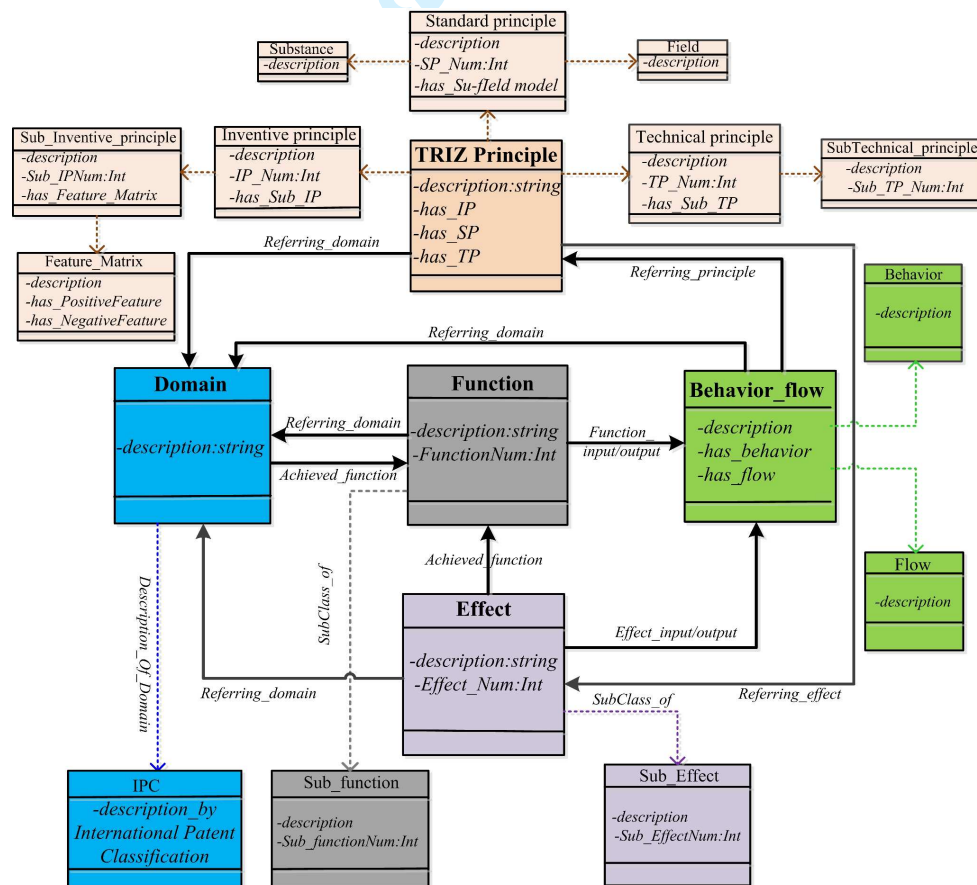


Figure 3. The semantic relationships of attribute ontology

3.2 Build the keyword concept set of document and the index of cluster

The new knowledge retrieval model applies the clustering method from Ref. [43] to cluster a large group of knowledge documents. Prior to the clustering processing, the Keyword Concept Set (KCS) of each knowledge document is generated and each knowledge document includes a group of keywords. It is worth pointing out that the generation of KCS is also the process of label-add (i.e. feature-add) to the knowledge documents. Next, the knowledge documents described by KCSs are clustered into different clusters and the cluster feature, we call it as $KCS_{cluster}$, are generated as well. The cluster feature works as the cluster indexes in the following matching process. So all relevant knowledge documents can be found in the selected cluster during the retrieval process. In fact, KCSs not only represent the semantic features of knowledge documents in this model, the semantic similarities among KCSs are also the foundation of the document clustering. The process of building the KCS of document and the index of cluster is shown in Figure 4.

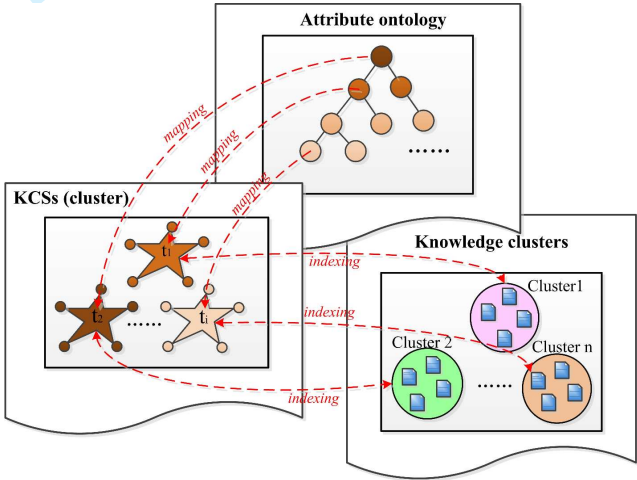


Figure 4. Diagram of building the KCS of document and the index of cluster

3.2.1 Build the keyword concept set of document

The keyword concepts are obtained by mapping of the Concept extension Set of the concepts from Attribute Ontology (CSAO). It is worth pointing out that the CSAO is actually the practical entity of the *semantic space* in Section 2. Because each keyword is mapped to one concept, each KCS can be represented by a group of concept tuples “ $t_i = (s_i, w_i)$ ”, where s_i denotes the semantic extension of concept c_i from the attribute ontology, w_i denotes the feature weight of concept c_i , and $i=\{1,2...N\}$, N is the number of the concepts of the KCS. The weight “ w ” for each keyword concept is computed by the “*tf-idf*” (term frequency-inverse document frequency)

$$w_{xi} = \frac{f_{xi}}{\sum_{k=1}^n f_{xk}} \log \frac{N_d}{df_i + 1} \tag{1}$$

where f_{xi} is the number of the occurrences of concept i in KCS_x (document x), n is the number of the concepts in KCS_x , df_i is the number of KCS which contains the concept i , N_d is the total number of KCSs (documents) of the knowledge document set (note that each document corresponds only one KCS).

By means of the WordNet, the concepts of the attribute ontology are extended to generate the CSAO. The WordNet provides the semantic relations of words, including hypernym, hyponym, synonymy, antonym, etc. Moreover, the words (including noun, verb, adjective, and adverb) in the WordNet are clustered into synonym sets respectively [44]. These synonym sets consist of basic conceptualized words in the same category and the semantic relations among these words. Taking the behaviour concept “concentrate” of the attribute ontology for example, its semantic extension by the WordNet is shown in Figure 5.

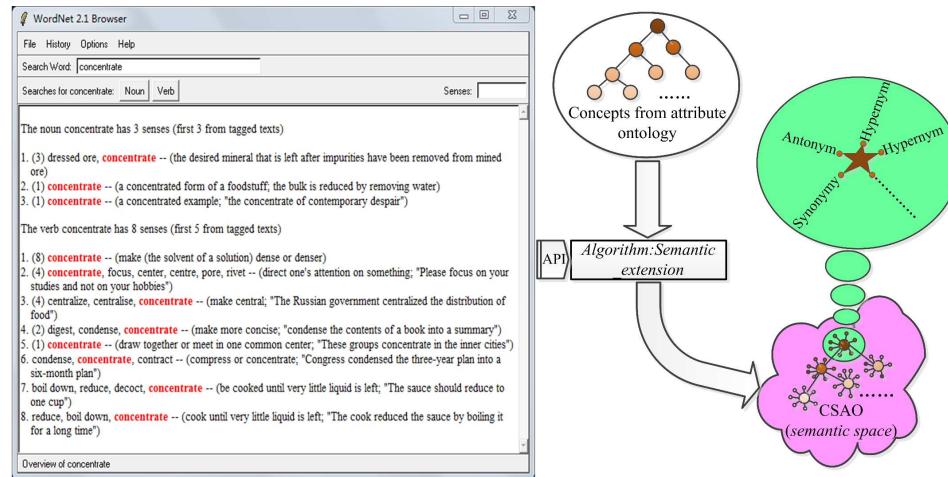


Figure 5. Example of the semantic extension of a concept based on the WordNet

The WordNet adopted by this paper is the Version 2.1, which contains about 150,000 morphologies and 200,000 semantics. The results of semantic extension of words are presented in the forms of verb and noun. The algorithm of invoking the WordNet 2.1 to generate the CSAO is shown in Figure 6.

Algorithm 1. Semantic extension based on the WordNet 2.1

Input: concepts from attribute ontology, “C” .

Output: extension set of concepts from attribute ontology, “CSAO” .

Algorithm_start:

1. *From nltk.corpus import WordNet as wn*
2. *Python. IDLE: File_InputStream (C): adding concepts from attribute ontology*
3. *Python. IDLE: initializing the WordNet Library*
4. *loop over each concept “C”*
5. *If C is void*
6. *Then report errors and skip*
7. *Else index concept “C” in WordNet Library*
8. *get synsets of “C” by inputting : wn.synsets (‘C’)*
9. *C = wn.synset (‘C. verb or noun. 0x ’)*
10. *CSAO= (C. hypernoms (); C. hyponyms (); C. synonyms ();*
C. Antonyms (); ...)
11. *Print_CSAO*
12. *Endif*

Algorithm_End.

Figure 6. Algorithm for the semantic extension based on the WordNet 2.1

3.2.2 Clustering

Building the index of cluster includes two main steps, the first step is to compute the semantic similarity among the KCSs of documents, and the second step is to cluster the documents and generate $KCS_{cluster}$.

For any pair of KCSs, KCS_x and KCS_y , by comparing the concepts comprising KCS_x with the those comprising KCS_y , the semantic similarity of KCS_x about KCS_y can be presented by the similarity set $p = \{l_1, l_2, \dots, l_m\}$ (m means that there are m number of concepts in KCS_x that are similar to the concepts in KCS_y). Likewise, by comparing the concepts comprising KCS_y with those comprising KCS_x , the semantic similarity of KCS_y about KCS_x can be presented by another similarity set $q = \{k_1, k_2, \dots, k_n\}$ (n means that there are n number of concepts in KCS_y are similar to the concepts in KCS_x). The individual similarity values (for individual concept comparison pair) in both similarity sets are computed according to the shortest path principle of the WordNet, and the similarity for arbitrary concept pair cx_i (from KCS_x) and cy_j (from KCS_y) is given by

$$csim(cx_i, cy_j) = C - length(cx_i, cy_j) - kd \quad (2)$$

where cx_i represents concept i of KCS_x (i can be any number from 1 to N_x , and N_x is the number of the concepts in KCS_x), cy_j represents concept j of KCS_y (likewise, j can be any number from 1 to N_y , and N_y is the number of the concepts in KCS_y), d denotes the number of direction changes in the path, and both C and k are constants. Finally, the semantic similarity between KCS_x and KCS_y is given by

$$tsim(KCS_x, KCS_y) = \frac{\sum_{i=1}^m l_i + \sum_{j=1}^n k_j}{m + n} \quad (3)$$

It is worth pointing out that, as defined by Ref. [45], the length of the path $length(cx_i, cy_j)$ in Eq. (2) is not too long and does not change direction too often in WordNet, meanwhile Ref. [45] also illustrated that the path length between two concepts may not exist, so when the path length “from concept i to concept j ” has a value, the path length “from concept j to concept i ” may do not have a value. Therefore, m and n in Eq. (3) may not equal, and Eq. (2) presents the general situation. Meanwhile, as described by Ref. [45], for any two concepts with existing path length between them, the similarity value of them (Eq. 2) ranges from 0 to 1. Therefore, the value of $tsim(KCS_x, KCS_y)$ in Eq. (3) also ranges from 0 to 1.

The clustering of knowledge documents is based on the principle of hierarchical clustering in Refs. [46, 47]. The algorithm of the clustering is shown in Figure 7. In this algorithm, the KCS of a knowledge cluster ($KCS_{cluster}$) is determined by the combination of the KCSs of all knowledge documents in the cluster, and the same keyword concepts from these KCSs will be merged together (no repeat for the same keyword concepts), so the number of the keyword concepts in the cluster will reduce. The detail of the clustering process and the combination of KCSs in the clustering process will be specified in the case study of Section 5.1. As mentioned in previous section, $KCS_{cluster}$ will function as the cluster’s index during the knowledge matching process. Because $KCS_{cluster}$ derive from the concepts of the attribute ontology with clear granularity hierarchy, the knowledge documents inside the clusters are also characterized with the

corresponding granularity (abstract) levels.

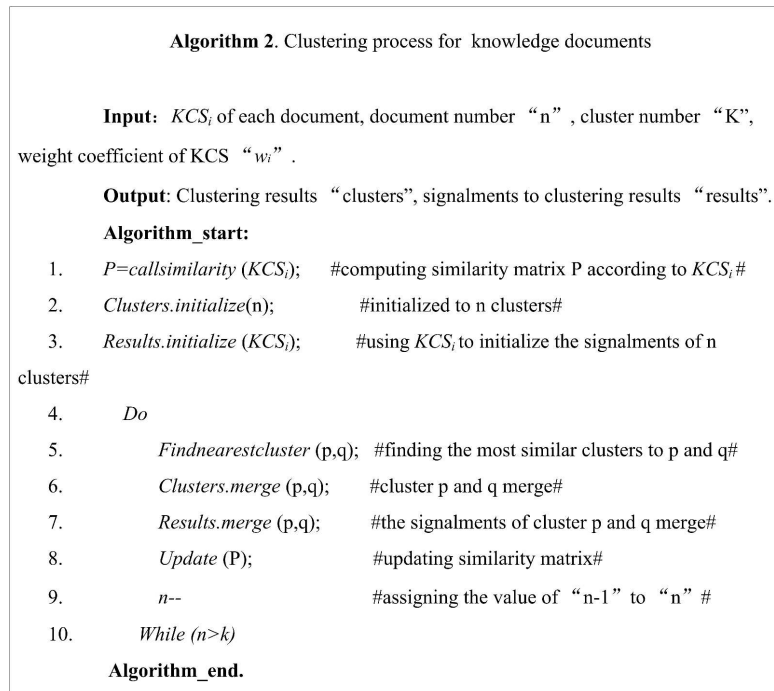


Figure 7. Document-clustering algorithm

4. New method for knowledge matching

The knowledge-based ontology is the standardized description of all aspects in the conceptual design stage. Any design problem in the conceptual design stage can also be described /abstracted by the ontology. Therefore, through the correlation of the concepts in the ontology, the mapping/matching between knowledge and design problems can be set up.

4.1 Semantic analysis on design problem

Before semantically analysing on a design problem, it is necessary to do the text pre-processing on this design problem. This pre-processing includes keyword extracting, keyword extending, word frequency statistics, and stop word removing. Based on the text description of the design problem, the keywords thereof are extracted, and the total number of the keywords is M . Next, similar to the generation of KCSs of knowledge documents in Section 3.2.1, the semantic extension concepts of these keywords is acquired by the mapping of the CSAO, and these concepts include the required TRIZ principle, the domain that this problem belongs to, the required function, the input/output of the behaviour flow that realizes the related function, and the involved effect. We define the keywords and their semantic extension as the Keyword Retrieval Set (KRS) of the design problem. Each KRS consists of a group of keyword tuples $t_j = (s_j, w_j)$, where s_j represents keyword j and its semantic extension, $j = \{1, 2, \dots, M\}$, and M equals the number of the keywords of the design problem, and w_j represents the feature weight of keyword, given by

$$w_j = \frac{f_j}{\sum_{k=1}^M f_k} \quad (4)$$

where f_j is the number of the occurrences of keyword j in the keyword set of the design problem. After the unnecessary words are removed, finally the keywords and keywords' semantic extension of the design problem are generated. Because the KRS of the design problem is generated by the mapping of the CSAO where the ontology has very clear granularity hierarchy, the KRS also has the corresponding granularity (abstract) levels.

4.2 Knowledge matching based on the similarity between KRS and KCS

4.2.1 Select cluster based on the similarity between KRS and $KCS_{cluster}$

As specified in the previous sections, we have applied KRS and KCS to describe the design problem and the design knowledge (the combined KCS for a cluster, as described in Section 3.2.2) respectively. As shown in Figure 8, the semantic extension structures of KRS and $KCS_{cluster}$ can be represented by the star-like elements that originate from the concepts of the attribute ontology. Note that the star-like elements (t element) are exact same as the structure shown in the right hand part of Figure 5, and they represent the different keyword concepts. During the matching process, a group of knowledge clusters are considered as the candidate clusters, and the semantic similarities between all candidate $KCS_{cluster}$ and KRS are calculated. The calculated results are used to indicate the similarity between the design problem and individual cluster.

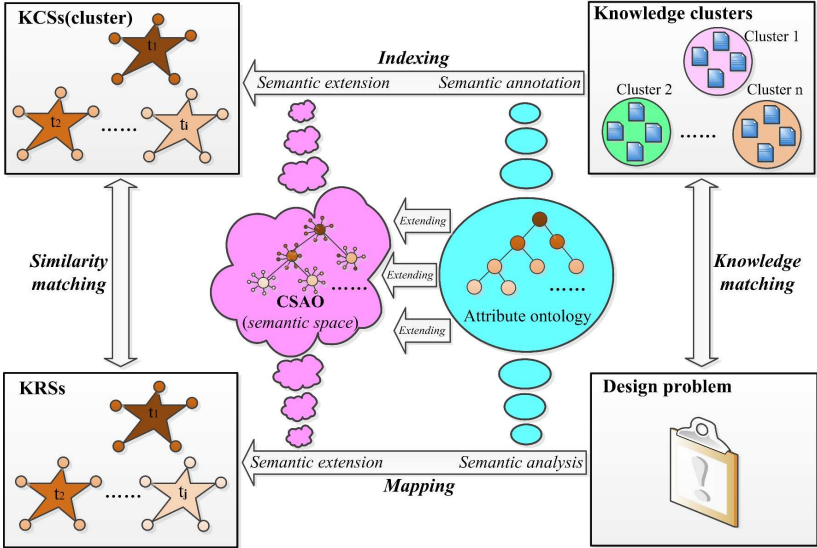


Figure 8. The knowledge matching based on the mapping between KRS and KCS

Similar to the similarity computing among KCSs in Section 3.2.2, the computing of the semantic similarity between a KRS and a $KCS_{cluster}$ is described as follows.

By comparing the concepts comprising the KRS with those comprising the $KCS_{cluster}$, the similarity of the KRS about the $KCS_{cluster}$ can be presented by the similarity set $p=\{l_1,l_2,...,l_m\}$ (m means there are m number of concepts in the KRS similar to the concepts in the $KCS_{cluster}$). The individual similarity value in the similarity set can be computed by the shortest path principle of the WordNet, and the similarity of arbitrary concept pair c_{krs_i} and c_{kcs_j} is given as

$$csim(c_{krs_i},c_{kcs_j})=C-length(c_{krs_i},c_{kcs_j})-kd \tag{5}$$

The definitions of the most of the items in Eq. (5) are same as those in Eq. (2), except that the subscript i can be any value from 1 to M (M is the number of the concepts in the KRS) and the subscript j can be any value from 1 to N (N is the number of the concepts in the $KCS_{cluster}$). Likewise, comparing the concepts comprising the $KCS_{cluster}$ with those comprising the KRS, the similarity of the $KCS_{cluster}$ about the KRS can be represented by another similarity set $q = \{k_1, k_2, \dots, k_n\}$ (n means that there are n number of concepts in the $KCS_{cluster}$ are similar to the concepts in the KRS), likewise the individual similarity value in the similarity set q can be computed by Eq. (5).

Considering the weight “ w_{kcs} ” of the $KCS_{cluster}$ (calculated by Eq. (3)) and the feature weights “ w_{krs} ” of the KRS (calculated by Eq. (4)), the similarity between the KRS and the $KCS_{cluster}$ is defined as

$$tsim(KRS, KCS_{cluster}) = \frac{\sum_{i=1}^m w_{krs_i} l_i + \sum_{j=1}^n w_{kcs_j} k_j}{m+n} \quad (6)$$

It can be seen that the value of $tsim(KRS, KCS_{cluster})$ ranges from 0 to 1. When the similarity is close to 1, it means there is high similarity between the KRS and the $KCS_{cluster}$.

As mentioned above, during the matching process, a group of clusters are regarded as the candidate clusters, so the similarity between each $KCS_{cluster}$ and KRS are calculated by Eq. (6) one by one, and the cluster that has the highest similarity value is selected. The selected cluster includes a series of knowledge documents which have high relevant with the design problem. It is worth pointing out that, as mentioned in the Section 3.2.2, the quantity of the keywords concepts in a $KCS_{cluster}$ is normally less than the overall quantity of the keyword concepts in the constituent KCSs (documents), so the computing cost of knowledge matching reduces to some extent. More documents in the whole database means more computing time saving. Therefore, the new retrieval model should show stronger advantage for a huge number of knowledge documents.

4.2.2 Ranking the documents according to granularity level

The knowledge documents in the selected cluster are not organized in the order of granularity levels at this moment. In order to find the documents with the suitable granularity level, the granularity properties of the concepts in the KCSs of these documents should be considered. Therefore, the following step of matching is to find the document with the expected granularity level. According to Ref. [48], if the shortest path distance in the WordNet between the concepts in a KCS and a KRS is smaller, and the KCS (their component concepts) has the closer abstraction degree (granularity level) with the KRS. Note that the concepts in the KRS have different granularity levels, and we can re-organize the concepts of the KRS in the order of granularity levels. Therefore, if the weights on granularity levels are considered for the concepts of the KRS in the similarity calculation, the order the KCSs (documents) about the granularity levels can be found from the similarity ranking. Consequently, Eq. (6) can be re-written as

$$tsim(KRS, KCS) = \frac{\sum_{i=1}^m W_{krs_i} w_{krs_i} l_i + \sum_{j=1}^n w_{kcs_j} k_j}{m+n} \quad (7)$$

where the “ w_{kcs} ” is again calculated by Eq. 1, but note that the number N of the KCSs has reduced to the number of the documents in the selected cluster. W_{krs_i} is the weights of the granularity of keyword concept i . W_{krs} is only decided by

granularity level, so the concepts with same granularity level have the same weight values. As a result, the $\sum_{i=1}^m W_{krs_i} W_{krs_i} l_i$ in Eq. (7) can be further re-written as

$$\sum_{i=1}^m W_{krs_i} W_{krs_i} l_i = W_{krs_1} \sum_{i=1}^{m_1} W_{krs_i} l_i + W_{krs_2} \sum_{i=m_1+1}^{m_2} W_{krs_i} l_i + \dots + W_{krs_k} \sum_{i=m_{k-1}+1}^{m_k} W_{krs_i} l_i + W_{krs_k+1} \sum_{i=m_k+1}^m W_{krs_i} l_i \quad (8)$$

where W_{krs_1} is the weight assigned for the granularity level 1, W_{krs_k} is the weight for the granularity level k , and there are $k+1$ granularity levels for the KRS ($k \geq 0$). From level 1 to level $k+1$, the granularity level becomes finer and finer. The subscript number of the keyword concepts in level 1 is from 1 to m_1 . The subscript number of the keyword concepts in level 2 is from m_1+1 to m_2 . The subscript numbers of the keyword concepts in remaining levels are defined in the same manner (note that the concepts in the KRS have previously been re-organized according to the order of their granularity levels).

Based on Eqs. (5), (7) and (8), the similarity value between each KCS and the KRS can be calculated, and the KCS that has higher similarity with the KRS should be in a relative coarse granularity level (note that we put higher weight for the coarser granularity level in Eq. (7)). Consequently, the corresponding documents can be ranked according to their granularity levels (from coarse to fine), and the model will show the designers which document is the most relevant one. It is worth pointing out that the remaining documents (except the most relevant document) in selected clusters can also be easily viewed and accessed by the designers in this model. All these remaining documents are actually relevant to the design problems, and the designers can choose to synthesize some of them with the most relevant documents (dependent on the different requirements of the design problems), finally generate more comprehensive design scheme. This is another advantage to cluster the knowledge documents before the matching process.

5 Case study and the design of whole knowledge retrieval system

5.1 Case study

In this section, we will simply implement the whole model proposed in the previous sections and employ some real design problems to demonstrate the advantages of the new model. In order to compare some classic knowledge retrieval models, we selected 100 knowledge documents from CAIP system (Ref [19]) as the retrieval source, and these documents also have keyword labels, co-reference labels, and functional labels. Based on the attribute ontology that was built by Protégé (as shown in Figure 9), different keyword concepts were assigned to the documents such that these keyword concepts could represent all five perspectives in the attribute ontology. The WordNet was invoked by the text corpus of Python to semantically extend the keyword concepts of the documents according to the algorithm shown in Figure 6. At this moment, we manually realized the mapping of the CSAO in this case study, but in the future, this process can be implemented by some automatic semantic processing techniques [49, 50]. Meanwhile the “*tf-idf*” equation (Eq.1) was used to calculate the weights of these keyword concepts. Subsequently the KCSs of the documents were obtained, in the form of $KCS_x (concept_{xi}, w_{xi})$, (x is the document number and equals to 1, 2...100, i is the keyword concept number and the maximal value of this number varies for different documents).

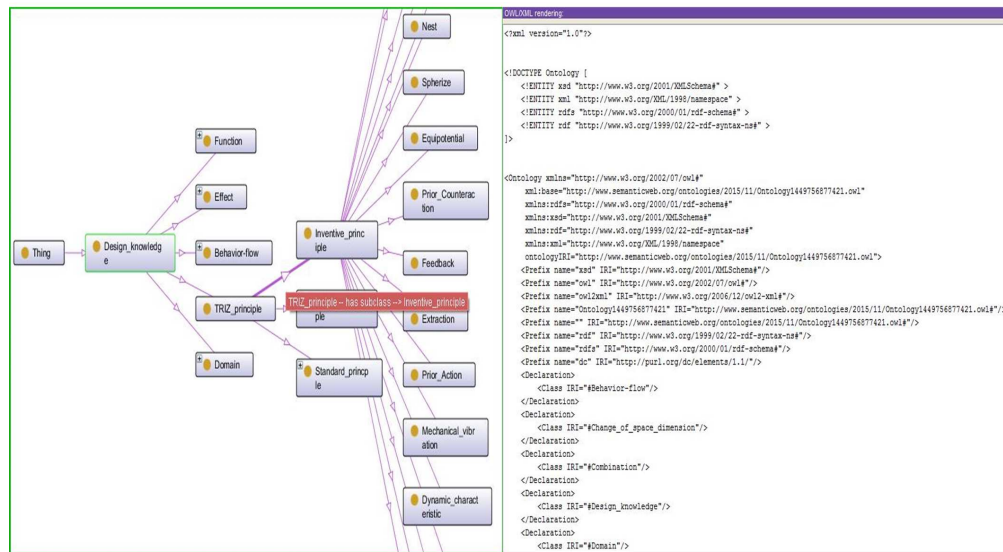


Figure 9. Attribute ontology built by Protégé

Next, based on Eqs. (2 & 3), the WordNet was invoked again by Python to calculate the length of the shortest path of each keyword concept pair $length(cx_i, cy_j)$ and also the similarity values among the KCSs. Take the similarity between KCS_1 ($concept_{11}, concept_{12} \dots concept_{1i}$) and KCS_3 ($concept_{31}, concept_{32} \dots concept_{3j}$) for example, the $concept_{11}$ is the synonym of “motor” and the $concept_{31}$ is the synonym of “valve”, Figure 10 is the snapshot of the similarity calculation of this two concepts through the WordNet (the length of the path) in Python, and the calculated similarity value is $csim(concept_{11}, concept_{31})=0.128$. After the similarity values between all concepts from KCS_1 and KCS_3 were computed, the similarity between KCS_1 and KCS_3 was calculated as $tsim(KCS_1, KCS_3)=0.201$. Repeated the similarity calculations on other KCSs, the similarity values among the 100 KCSs was finally obtained, and the results (matrix) is shown in Table 1.

Figure 10. Snapshot of the similarity calculation between two concepts in Python through the WordNet

Table 1. Calculated similarity results (matrix) among 100 KCSs

Similarity	KCS_1	KCS_2	KCS_3	KCS_4	...	KCS_{97}	KCS_{98}	KCS_{99}	KCS_{100}
KCS_1	1	0.536	0.201	0.176	...	0.603	0.474	0.185	0.103
KCS_2	0.536	1	0.417	0.213	...	0.566	0.613	0.216	0.497
KCS_3	0.201	0.417	1	0.491	...	0.227	0.362	0.483	0.681
KCS_4	0.176	0.213	0.491	1	...	0.208	0.272	0.653	0.511
...
KCS_{97}	0.603	0.566	0.227	0.208	...	1	0.581	0.193	0.128
KCS_{98}	0.474	0.613	0.362	0.272	...	0.581	1	0.233	0.549
KCS_{99}	0.185	0.216	0.483	0.653	...	0.193	0.233	1	0.630
KCS_{100}	0.103	0.497	0.681	0.511	...	0.128	0.549	0.630	1

Based on the similarity results (matrix) among the KCSs, the clustering of the 100 documents was carried out in MATLAB according to the algorithm shown in Figure 7, and the process is specified as follow.

Step1: Initially classified the 100 documents into 100 clusters; meanwhile assigned the KCS of individual document to the corresponding cluster, so the similarity values among these initialized clusters are same as those in Table 1.

Step 2: According to the similarity matrix shown in Table 1 (note that this matrix is a symmetric matrix), found out two clusters that have the highest similarity values and combined them into a new cluster.

Step 3: Combined the keyword concepts of the KCSs of the two clusters mentioned in step 2, and re-calculated the weight of each keyword concept, consequently generated the KCS of the new cluster, meanwhile updated the similarity matrix. Take the combination of *Cluster1* and *Cluster2* for an example to illustrate this step, the keyword concepts of the new combined cluster include “*concept_{1,1}, concept_{1,2}...concept_{1,i-n}* and *concept_{2,1}, concept_{2,2}...concept_{2,j-n}*” (*i* and *j* are the total numbers of keyword concepts in *Cluster1* and 2 respectively, *n* is the number of common keyword concepts between *Cluster1* and 2). Next, based on all clusters, including the remaining initialized clusters and new combined *cluster1_2*, the weight of the each keyword concept in *cluster1_2* was calculate through Eq.(1) and the KCS of *cluster1_2* can be created as $KCS_{cluster1_2}$. Finally, the new similarity matrix about $KCS_{cluster1_2}, KCS_3, KCS_4...KCS_{100}$ was computed.

Step 4: Repeated Step2 and Step3, so that all clusters with similarity values over the threshold were classified into different clusters.

The KCSs of the clusters were generated in the form of $KCS_{cluster}(concept_{cluster}, w)$ and the classified keyword concepts of the clusters (snapshot) is shown in Table 2. Not that these clusters’ KCSs will function as the index of cluster in the following knowledge matching process.

Table 2. Classified keyword concepts of the clusters

Cluster	Number of documents	Keyword concepts of cluster
<i>Cluster1</i>	23	Nesting, mechanical-drive, supercharger, electromechanical, etc.
<i>Cluster2</i>	20	Electromagnetic, periodic-effect, electricity-producing, motor, etc.
<i>Cluster3</i>	19	Feedback, regulating, signal-control, automation, valve, etc.
<i>Cluster4</i>	14	Separation, Cutting, dynamic-functioning, combined-device, etc.
<i>Cluster5</i>	13	Multifunction, combination, motion-producing, machine, etc.
<i>Cluster6</i>	9	Mechanical-wave, vibration, mediator, energy-producing, etc.
<i>Cluster7</i>	1	Thermogenic, prior action, leakage-protection and thermo-electricity.
<i>Cluster8</i>	1	Deformation, superplasticity, extrusion and compound-process.

Following the KCS extraction and clustering on the knowledge documents, we carried out the knowledge matching for a specific design problem. The example design problem is “improvement design of a sensor faucet, with energy generation and storage”. According to the attribute ontology (shown in Figure 9), the keywords of the design problem were manually generated, including ‘Generator, Accumulator, Power-storage, Feedback, Electromagnetic’. But similar to the situation above, the automatic mapping of the CSAO can be implemented by some automatic semantic processing techniques in the future. The WordNet was invoked by the text corpus of Python to semantically extend these keywords, and the weights of the keywords were calculated by Eq. (4). Consequently, we obtained the KRS of the design problem. Based on the Eqs. (5, 6), the similarity values between the KCSs (clusters) and the KRS were computed, and the results are shown in Table 3. According to the ranking of the similarity values (from the highest to the lowest), we further carried out similarity analysis on *Cluster2* and it was found that the knowledge thereof involves multiple perspectives (effect, inventive principle, function, domain, etc.) and also has strong relevance with the design problem, therefore *Cluster2* was selected as the most relevant cluster. Meanwhile, the recall rate of the searching (matching) was computed in MATLAB as $Recall=0.57$, and the precision rate is $Precision=0.65$.

Table 3. Similarity values between the KCSs (clusters) and KRS

Similarity	$KCS_{cluster1}$	$KCS_{cluster2}$	$KCS_{cluster3}$	$KCS_{cluster4}$	$KCS_{cluster5}$	$KCS_{cluster6}$	$KCS_{cluster7}$	$KCS_{cluster8}$
<i>KRS</i>	0.295	0.361	0.286	0.162	0.115	0.223	0.103	0.074

In order to rank the documents in the selected *Cluster2* according to the granularity levels, it is necessary to re-study the keywords and their weights of the KRS of the design problem. Totally there are five keyword concept tuples in *KRS*, “(*concept1*, w_{krs_1}), (*concept2*, w_{krs_2}), (*concept3*, w_{krs_3}), (*concept4*, w_{krs_4}), (*concept5*, w_{krs_5})”. It was found that these keywords in *KRS* are belong to two granularity levels, “*concept1* (Generator)” and “*concept2* (Accumulator)” are in a same fine granularity level, and “*concept3* (Power-storage)”, “*concept4* (Feedback)” and “*concept5* (Electromagnetic)” are in a same coarse granularity level. Therefore, we introduced a higher granularity weight to *concept1*&2 and a lower granularity weight to *concept3*, 4 &5. After manually adjusting granularity weights several times (similar to the trial and error principle), the new similarity values between *KRS* and *KCSs* (the *KCSs* of the documents in *Cluster2*) were calculated by Eqs. (7, 8), and we finally rank the documents in *Cluster2* according to the similarity values. The calculation and ranking results for each weighting adjustment are shown in Table 4. Note that there are total 20 documents in *Cluster2*, but we still kept the document number as the original document number in 100-document set.

Table 4. Similarity values between KRS and KCSs (documents in *Cluster2*) for different granularity weights

Similarity	KCS_1	KCS_2	KCS_7	KCS_{19}	KCS_{21}	...	KCS_{73}	KCS_{74}	KCS_{79}	KCS_{97}	KCS_{98}
------------	---------	---------	---------	------------	------------	-----	------------	------------	------------	------------	------------

<i>KRS_{adjustment1}</i>	0.341	0.215	0.337	0.163	0.315	...	0.194	0.236	0.310	0.221	0.167
Ranking1	<i>Doc₆₁</i>	<i>Doc₄₂</i>	<i>Doc₇₄</i>	<i>Doc₁</i>	<i>Doc₇</i>	...	<i>Doc₃₇</i>	<i>Doc₅₂</i>	<i>Doc₉₈</i>	<i>Doc₁₉</i>	<i>Doc₄₉</i>
<i>KRS_{adjustment2}</i>	0.326	0.229	0.349	0.182	0.301	...	0.170	0.285	0.291	0.239	0.153
Ranking2	<i>Doc₆₁</i>	<i>Doc₄₂</i>	<i>Doc₇</i>	<i>Doc₄₂</i>	<i>Doc₁</i>	...	<i>Doc₇₃</i>	<i>Doc₅₂</i>	<i>Doc₁₉</i>	<i>Doc₉₈</i>	<i>Doc₄₉</i>
<i>KRS_{adjustment3}</i>	0.297	0.241	0.392	0.234	0.265	...	0.162	0.359	0.279	0.255	0.148
Ranking3	<i>Doc₇</i>	<i>Doc₃₅</i>	<i>Doc₆₁</i>	<i>Doc₄₂</i>	<i>Doc₅₁</i>	...	<i>Doc₆₉</i>	<i>Doc₄₉</i>	<i>Doc₇₃</i>	<i>Doc₉₈</i>	<i>Doc₅₂</i>
<i>KRS_{adjustment4}</i>	0.203	0.274	0.430	0.283	0.199	...	0.138	0.398	0.260	0.284	0.142
Ranking4	<i>Doc₇</i>	<i>Doc₃₅</i>	<i>Doc₅₁</i>	<i>Doc₇₄</i>	<i>Doc₆₁</i>	...	<i>Doc₆₉</i>	<i>Doc₉₈</i>	<i>Doc₄₉</i>	<i>Doc₇₃</i>	<i>Doc₅₂</i>

	<i>Wkrs_1</i>	<i>Wkrs_2</i>	<i>Wkrs_3</i>	<i>Wkrs_4</i>	<i>Wkrs_5</i>	The most similar document's granularity becoming finer?	Documents' ranking reflects granular order?
Adjustment1	0.26	0.26	0.16	0.16	0.16	No	No
Adjustment2	0.32	0.32	0.12	0.12	0.12	No	No
Adjustment3	0.395	0.395	0.07	0.07	0.07	Yes	No
Adjustment4	0.485	0.485	0.01	0.01	0.01	Yes	Yes

Figure 11. Analysis on the ranking of the documents in *Cluster2* according to the granularity levels

The further analysed results on the ranking of the documents in *Cluster2* are shown in Figure 11. It was found that, with the No.4 weight adjustment, *Document7* is the document with the highest similarity value and the keyword concepts in *Document7* have the finest granularity level, more importantly, the ranking order of all 20 documents in *Cluster2* were perfectly in line with the order from the finest granularity level to the coarsest granularity level. Therefore, we reckoned that this weight adjusting has reached the final ranking target. *Document7* was selected as the most relevant document. The knowledge description of *Document7* is “an instance of using existing components/conditions to generate electricity and store power: a type of energy-recovering equipment”. The design problem is not very complicated and it is a practical/improvement oriented issue. Compared to other documents in *Cluster2*, the detail knowledge in *Document7* belongs to the fine granularity level and can provide more insights for a practical design. The knowledge in other course granularity levels, such as the principle knowledge and domain knowledge cannot support the solving of the example problem efficiently. However, as mentioned above other remaining documents in *Cluster2* are relevant to this design problem, so the designer can directly view these documents in this model and also easily refer to ideas of the documents during the problem solving process.

Based on the knowledge described in *Document7*, the design procedures for a new sensor faucet with energy generation and storage are shown in Figure 12. The sensor of the old faucet is powered by normal batteries, and this causes the issue of inconvenience (frequent change of dry battery/cell) and pollution. Therefore, based on these resource constraints, the improvable elements are from two aspects, including electricity generation and power storage. The referential elements (as shown in Figure 12) from *Document7* can be briefly summarized as “integrating the electric motor and accumulator to transfer the gravitational potential energy, which was from normal operation, into electricity power, and also store the power”. Based on the inspiration from the referential elements, the improvement design scheme was generated: adding a

hydraulic generator into the control system of the faucet, replacing the dry cell by a storage battery, so the energy from the running water (when the faucet operates) can drive the hydraulic generator to generate electricity power, meanwhile the power is stored in the storage battery to power the sensor for a long time.

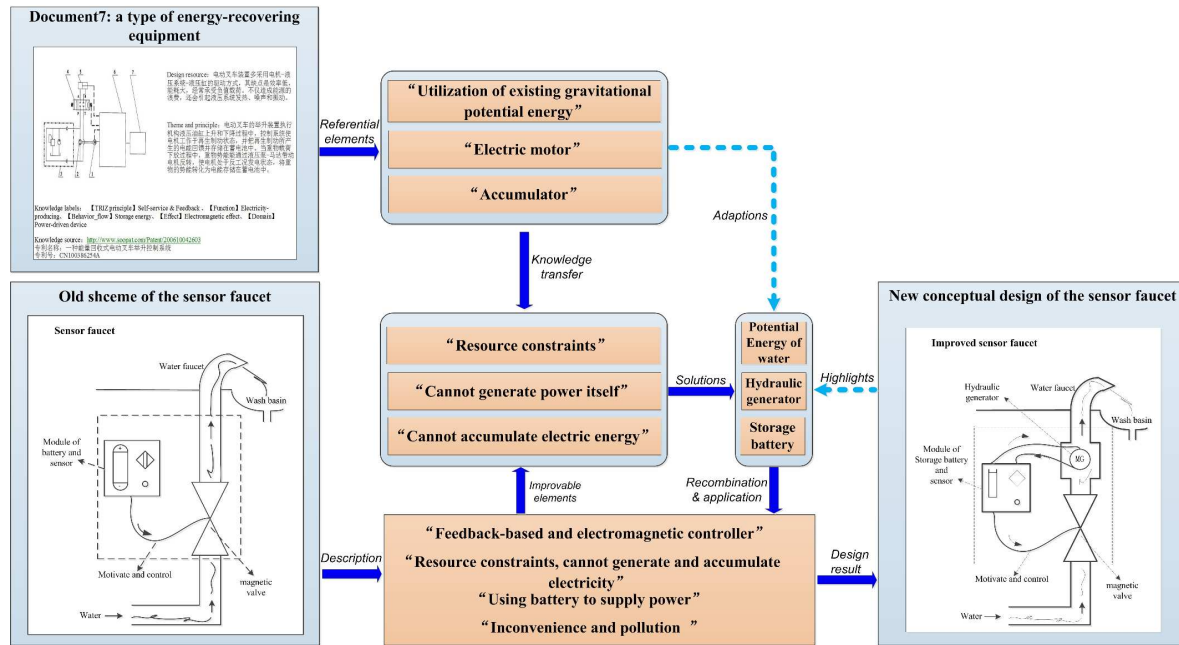


Figure 12. Improvement design scheme for a new sensor faucet with energy generation and storage

Furthermore, in order to evaluate the advantage of the new model, we introduced 10 design problems and repeated the knowledge retrieval process above. On the other hand, three existing knowledge retrieval methods, as mentioned in Introduction Section, were applied to these 10 design problems as well. These existing methods include the Keyword-based method, the Co-reference based method, and the Function ontology based method. The calculated recall-precision curves of three classic methods were considered as the references of that from new model. The comparison of the final results is shown in Figure 13. It can be found that the results from the classic method, i.e. Keyword based method, is the worst in the comparison, and the Co-reference based method (TRIZ & Function based) shows better performance and higher accuracy than the Function ontology based method in some retrieval conditions, but not always. More importantly, in the comparison with other three existing methods, the new designed method achieved the best performance in all conditions.

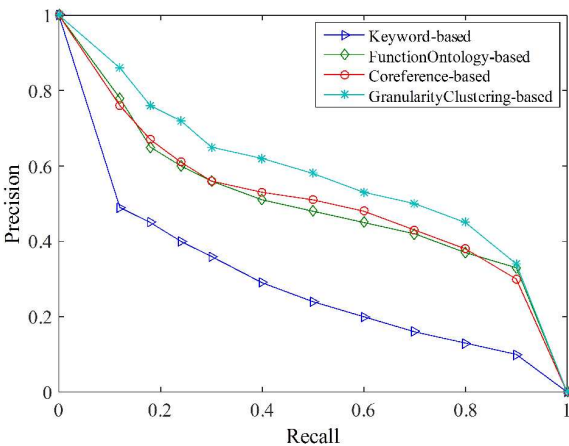


Figure 13. Recall-precision comparison between the new retrieval models and three classic methods

5.2 Design of whole knowledge retrieval system

Some of the procedures in the above case study were manually implemented according to the algorithms designed in Section 2, 3 and 4. For instance, 100 knowledge documents were manually inputted from the knowledge database, and the keywords concepts of the knowledge documents and the design problems were manually mapped from the attribute ontology. There is a great potential to implement these procedures by some advanced big-data storage/processing techniques, automated techniques and artificial intelligent techniques in the future. Therefore, after considering the integration of the new model with the systems developed in our former works [19, 32, 51], in this section we designed the comprehensive knowledge retrieval system (as shown in Figure 14), which will become the guide of our future work.

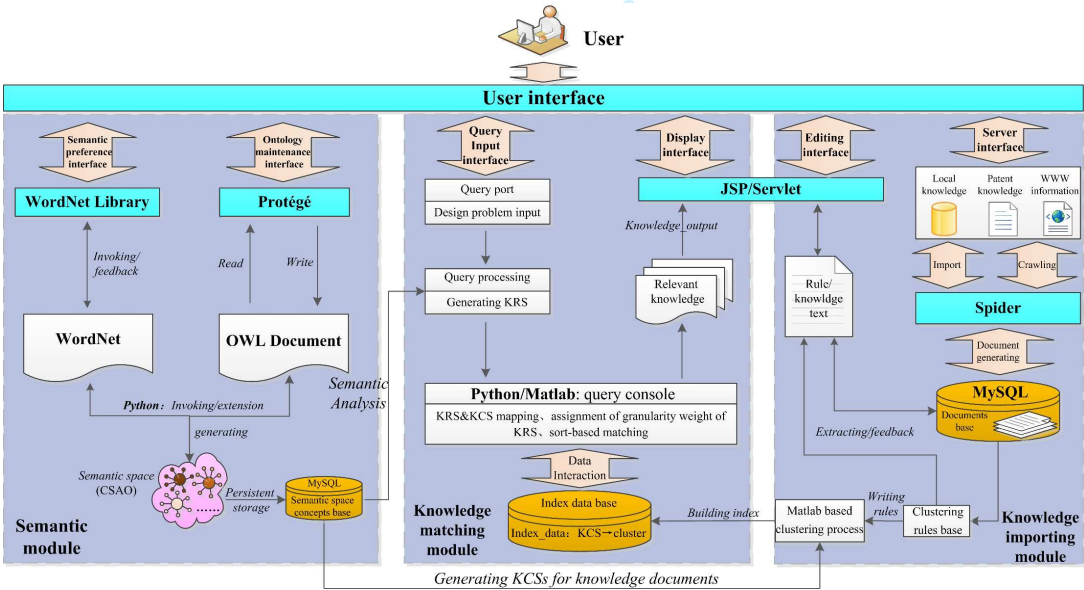


Figure 14. Diagram of whole knowledge retrieval system

This whole knowledge retrieval system includes three modules, i.e. semantic module, knowledge-matching module and knowledge importing module.

Semantic model mainly involves using Protégé to build an attribute ontology (as shown in Figure 9) and mapping the

keywords of both knowledge document and design problem to the concepts in attribute ontology (namely CSAO). As mentioned in Section 5.1, in the future the mapping of the CSAO can be implemented by some automatic semantic processing techniques [49, 50], and attribute ontology and the all mapped keyword concepts can be permanently stored by SQL. As mentioend in Section 3.1, the storage format of the ontology is OWL.

Knowledge matching module involves the generation of the KRS of design problems, the similarity calculation between KRS and KCS_{sclusters}, the selection of most relevant cluster, and the ranking of the documents in the selected cluster according to the granularity level. As described in Section 5.1, the semantical extension of design problem's keywords were carried out by the WordNet and Python, the weight calculation of the keywords, the similarity calculation and the ranking of the documents was coded in MATLAB as well. Likewise, in the future, the automatic mapping of the CSAO about the semantically extending of design problems' keywords can also be implemented by some automatic semantic processing techniques, and the weight adjustment about granularity levels in the document ranking process can be implemented by some artificial intelligent techniques, such as Neural Networks.

Knowledge importing module is actually a comprehensive knowledge database, which is the foundation of the KCS' generation and clustering. This knowledge includes local knowledge, patent knowledge, and the knowledge from general scientific/technical websites. The local knowledge was manually imported from books, articles, technical documents etc.. The Web Spider (Web Crawler) was used to crawl into patent websites and general websites to scrape relevant knowledge. Our group has already made some research on the web crawler and artificial patent knowledge checking [52]. Figure 15 is the snapshot of the Web Spider developed for a patent website, and Figure 16 shows the interface for the artificial patent knowledge checking. As mentioned in section 2, the knowledge importing in this module is a hidden input, the real input of the system is the problem descriptions input by the designers (in the matching modules).

```

BaiTengPatentSpider.java
...
continue;
}
patent.setTitle(BaiTengPatentParser.getInstance().parseTitle(content));
patent.setAddress(BaiTengPatentParser.getInstance().parseAddress(content));
patent.setApplier(BaiTengPatentParser.getInstance().parseApplier(content));
patent.setApplyDate(BaiTengPatentParser.getInstance().parseApplyDate(content));
patent.setInventor(BaiTengPatentParser.getInstance().parseInventor(content));
patent.setMainClassNum(BaiTengPatentParser.getInstance().parseMainClassNum(content));
patent.setContent(BaiTengPatentParser.getInstance().parseContent(content));
patent.setTime(BaiTengPatentParser.getInstance().parseTime());
patent.setUrl(BaiTengPatentParser.getInstance().parseUrl(patent.getApplyNum()));
patent.setFirstClass(BIPatentClassifier.getInstance().getFirstClassCategory(patent.getContent()));
patent.setFirstClassConfidence(BIPatentClassifier.getInstance().getFirstClassProbability(patent.getContent()));
ClassifyResult second = NBModel.getInstance().classifyInstance(patent.getFirstClass(), patent.getContent());
patent.setSecondClass(second.getClassName());
patent.setSecondClassConfidence(second.getConfidence());
patents.add(patent);
} else {
    System.out.println("非有效发明专利");
}
}

return patents;
}

/**
 * 得到批量文本中的起始页码
 * @return
 */
public int getStartPageNum() {
    int startPageNum = 0;
    PatentState ps = Configuration.getConfiguration();
    startPageNum = Integer.parseInt(ps.getStartPage());
    return startPageNum;
}

```

Figure 15. Web Spider for the patent website 'www.51patent.com'



Figure 16. Patent knowledge checking interface

Finally, all these knowledge documents were edited and stored in standard forms, meanwhile the JSP (Java Server Page)/Servlet will be used to post-process and browse the knowledge document in the future. Figure 17 shows a document presented in the form of template and sample. All knowledge was saved into the database by SQL, and currently there are 4750 knowledge documents, but we are trying to create much more documents in the future.

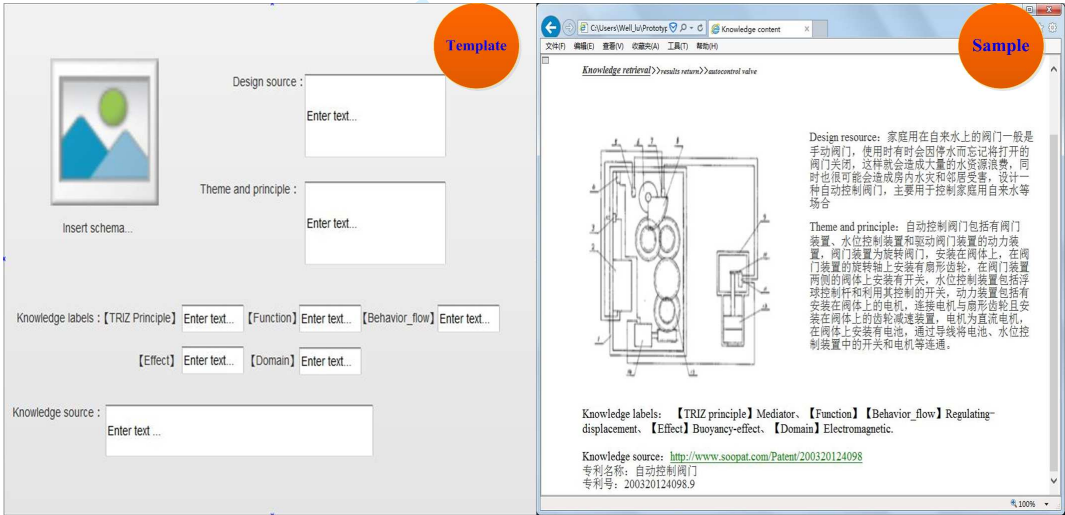


Figure 17. Template and sample of design knowledge

7. Conclusion

Since the solving process of the design problems in the conceptual design stage involves multiple-perspective knowledge, and these different perspectives can also be further extended into concepts with different abstract levels. Therefore, we built a new multi-perspective and multi-granularity model in this paper to efficiently retrieve the knowledge such as the selected knowledge document has the highest relevance and suitable granularity about the design problems. The critical point of the model is the knowledge based ontology, and the ontology includes two aspects, one is conceptualization and the other is the correlation among the concepts. The conceptualization also consists of two aspects, abstract description (attributes and concepts) and abstract level (granularity). In this model, the keyword concepts of any knowledge document can be mapped from the concepts of the ontology. Moreover, based on the correlation of the concepts in the ontology, the documents are classified into different clusters, so the combined keyword concepts of the clusters become the index of knowledge matching process. After the best cluster is selected in the knowledge matching process, the model

can arrange the order of the documents in the selected cluster according to their granularity levels. Finally, the model provides the designer the best knowledge document. In the last section, the case study demonstrated that the knowledge document retrieved by the model could provide the best idea for designers in the design problem solving. Moreover, through the comparison of the retrieval results, it verified that that new model had better performance than other existing retrieval methods. Additionally, we also designed the overall knowledge retrieval system, which can be the guide of the future works.

Acknowledgements

This research is supported in part by the National Natural Science Foundation of China (No.51435011, No.51505309), and the Research Fund for the Doctoral Program of Higher Education of China (No.20130181130011).

References

- Bertola P. and Teixeira J.C. Design as a knowledge agent: How design as a knowledge process is embedded into organizations to foster innovation, *Design Studies* 2003; 24(2): 181-194.
- Senthil K. Chandrasegaran, Karthik Ramani, Ram D. Sriram, et al. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design* 2013; 45: 204-228.
- Chen Yong, Zhao Meng, Xie Youbai, et al. A new model of conceptual design based on Scientific Ontology and intentionality theory. Part II: The process model. *Design Studies* 2015; 38: 139-160.
- Lindlof L, Soderberg B and Persson M. Practices supporting knowledge transfer - an analysis of lean product development. *International Journal of Computer Integrated Manufacturing* 2013; 26(12): 1128-1135.
- Mukherjee S, Uzzi B, Jones B, et al. A New Method for Identifying Recombinations of Existing Knowledge Associated with High-Impact Innovation, *Journal of Product Innovation Management* 2016; 33(2): 224-236.
- Altshuller G S. *The inventor algorithm, TRIZ, systematic innovation and technical creativity*. Worcester, Mass. , USA: Technical Innovation Center, 1999.
- Gero J. Design prototypes: a knowledge representation schema for design. *AI Magazine* 1990; 11(4): 26-36.
- Suh N. P. *The principles of design*. Oxford: Oxford University Press, 1990.
- Chen Y., Zhang Z. N., Liu Z. L., et al. Toward a scientific ontology-based concept of function. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 2013; 27(3): 241-248.
- Paul Prickett and Ivan Aparicio. The development of a modified TRIZ Technical System ontology. *Computers in Industry* 2012; 63(3): 252-264.
- Yan W, Zanni-Merk C., Cavallucci D., et al. An ontology-based approach for using physical effects in inventive design. *Engineering Applications of Artificial Intelligence* 2014; 42: 2264-2275.
- He Cong and Loh Han Tong. Grouping of TRIZ Inventive Principles to facilitate automatic patent classification. *Expert Systems with Applications* 2008; 34(1): 788-795.
- Yoon Janghyeok and Kim Kwangsoo. An automated method for identifying TRIZ evolution trends from patents. *Expert Systems with Applications* 2011; 38(12): 15540-15548.
- Invention Machine Corporation. Goldfire innovator, www.inventionmachine.com (2009, accessed 22 April 2016).
- CREAX. CREAX Innovation Suite 3.1, www.creaxinnovationsuite.com (2007, accessed 22 April 2016).
- IWINT Inc. Pro/Innovator, www.iwint.com.cn (2009, accessed 22 April 2016).
- Stone R. B. and Wood K. L. Develop of a functional basis for design, *Journal of Mechanical Design* 2002; 122(4): 359-370.

18. Li Zhen, Tate Derrick, Lane Christopher, et al. A framework for automatic TRIZ level of invention estimation of patents using natural language processing, knowledge-transfer and patent citation metrics. *Computer-Aided Design* 2012; 44: 987-1010.

19. Tu Jianwei, Li Yan and Li Wenqiang. Knowledge Retrieval Model and Implementation for Product Innovative Design, *Computer Integrated Manufacturing System* 2013; 19(2): 300-308.

20. Sun Lin, Xu Jiucheng and Liu Yangyang. Concept Granule-Based Granular Lattice and Application in Knowledge Retrieval. *Journal of software* 2014; 9(7): 1976-1983.

21. Zhou Danchen. Material Similarity Algorithm for Process Cases Retrieval Based on Granular Computing. *Chinese Journal of Mechanical Engineering* 2014; 50(13): 170-177.

22. Yan Xin, Raymond Y.K. Lau, Song Dawei, et al. Toward a Semantic Granularity Model for Domain-Specific Information Retrieval. *ACM Transactions on Information Systems* 2011; 29(3): 39-100.

23. Gruber T. A translation approach to portable ontology specifications. *Knowledge Acquisition* 1993; 5(2): 199-200.

24. Gruber T. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer studies* 1995; 43(5/6): 907-928.

25. Zhang D, Hu D, Xu Y, et al. A Framework for Design knowledge Management and Reuse for Product-Service Systems in Construction Machinery Industry. *Computers in Industry* 2012; 63(4): 328-337.

26. Sun Jie, Lu Wen Feng and Loh Han Tong. Building a database for product design knowledge retrieval—A case study in robotic design database. *Robotics and Computer-Integrated Manufacturing* 2010; 26(3): 224-229.

27. Zhang Zhinan, Liu Zelin, Chen Yong, et al. Knowledge flow in engineering design: An ontological framework. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 2013; 227(4): 760-770.

28. Pedrycz W, Russo B, Succi G, et al. Knowledge transfer in system modelling and its realization through an optimal allocation of information granularity. *Applied Soft Computing* 2012; 12(8): 1985-1995.

29. Robertson S L. *Psychology of problem solving*. Beijing: China Light Industry Press, 2004.

30. Zhang Qinghua. *Multi-granularity knowledge and measurement*. Beijing: Science Press, 2013.

31. Wang G Y. Extension of rough set under incomplete information systems. *Journal of Computer Research and Development* 2002; 39(10): 1098-1103.

32. Liu Longfan, Li Yan, Ma Jinlong, et al. Application of TRIZ in Creative Design Based on Knowledge Granularity. *Chinese Journal of Mechanical Engineering* 2016; 52(5): 22-32.

33. Li Guangrui, Guo Gang, Tang Huamao, et al. Granular computing model for conflict resolution in product design. *Journal of Chongqing University* 2011; 34(11):37-43.

34. Havens T. C., Bezdek C. J., Leckie C., et al. Fuzzy c-Means Algorithms for Very Large Data. *IEEE Transactions on Fuzzy Systems* 2012; 20(6): 1130-1146.

35. Garcia Piquer A, Sancho Asensio A, Fornells A, et al. Toward high performance solution retrieval in multiobjective clustering. *Information Sciences* 2015; 320: 12-25.

36. Yildizer E., Balci A.M., Jarada T.N., et al. Integrating wavelets with clustering and indexing for effective content-based image retrieval. *Knowledge-Based Systems* 2012; 31: 55-66.

37. Zolkepli Maslina, Dong Fangyan and Hirota Kaoru. Automatic Switching of Clustering Methods based on Fuzzy Inference in Bibliographic Big Data Retrieval System. *International Journal of Fuzzy Logic and Intelligent systems* 2015; 14(4): 256-267.

38. Niemann Michael, Siebenhaar Melanie, Schulte Stefan et al. Comparison and retrieval of process models using related cluster pairs. *Computers in Industry* 2012; 63(2): 168-180.

39. Chan K. Y., Kwong C. K. and Hu B. Q.. Market segmentation and ideal point identification for new product design

- using fuzzy data compression and fuzzy clustering methods. *Applied Soft Computing* 2012; 12(4): 1371-1378.
40. Jing Hui, Li Cong, Huang Meifa, et al. A fast retrieval method based on K-means clustering for mechanical product design. *Advanced Manufacturing Technology* 2011; 156-157: 98-101.
41. Zhang Guohai and Li Yusheng. Retrieval and clustering for multiple disciplines design knowledge based on ontology and semantic. *Applied Mechanic and Materials* 2013; 300-301: 155-159.
42. Tatiana A. Gavrilova and Irina A. Leshcheva. Ontology design and individual cognitive peculiarities: A pilot study. *Expert Systems with Applications* 2015; 42(8): 3883-3892.
43. Zhu Huifeng, Zuo Wanli and He Fenglin. A Novel Text Clustering Method Based on Ontology. *Jilin University Journal (Science Edition)* 2010; 48: 277-283.
44. Wei Tingting, Lu Yonghe, Chang Huiyou, et al. A semantic approach for text clustering using Wordnet and lexical chains. *Expert Systems with Applications* 2015; 42(4): 2264-2275.
45. Budanitsky A and Hitst G. *Semantic Distance in WordNet: an Experimental, Application-Oriented Evaluation of Five Measure*. Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources, Pittsburgh: Carnegie Mellon University Press, 2001; 29-34.
46. Luger F. George. *Artificial Intelligence: Structure and Strategies for Complex Problem Solving, Sixth Edition*. Beijing: China Machine Press, 2009; 301-303.
47. Christos Bouras & Vassilis Tsogkas. A clustering technique for news articles using WordNet. *Knowledge-Based Systems* 2012; 36:115-128.
48. ZHANG Dongna, ZHOU Chunguang, LIU Yanbing, et al. A semantic similarity computing approach based on Wordnet and Corpus statistics. *Journal of Jilin University (Science Edition)* 2010; 48(5): 811-816.
49. Sowa F. J. Principles of Semantic Networks: Explorations in the Representation of Knowledge. *Frame Problem in Artificial Intelligence*, 1991, 95-107
50. González G. R. A Semantic Web approach to Digital Rights Management, *Ph.D. Thesis, Universitat Pompeu Fabra, Spain*, 2005
51. Yang Kun, Li Yan, Xiong Yan, et al. A model for computer-aided creative design based on cognition and iteration. *Proc IMechE Part C: Journal of Mechanical Engineering Science* 2015; DOI: 10.1177/0954406215611438.
52. Gong Jianwen, Li Yan and Li Wenqiang. Construction and application of the patent knowledge system for supporting product innovation. *Chinese Journal Engineering Design*, 2013; 20(4): 265-271.