Int J Adv Manuf Technol (2016) 84:103–118
DOI 10.1007/s00170-015-7996-z

ORIGINAL ARTICLE

Common intelligent semantic matching engines of cloud manufacturing service based on OWL-S

Buyun Sheng [1] · **Chenglei Zhang** [1] · **Xiyan Yin** [1] · **Qibing Lu** [1] · **Yuan Cheng** [1] · **Ting Xiao** [1] · **Huimin Liu** [2]

Received: 26 April 2015 / Accepted: 19 October 2015 / Published online: 17 November 2015

Abstract In order to improve the practicability of the search engine of cloud manufacturing platform in small- and medium-sized enterprises (SMEs), the paper studied the matching process' diversity, heterogeneity, and multi-constraints between the intelligent matching engine and cloud manufacturing (CMfg) service, then established an intelligent searching engine of CMfg service in SMEs which is based on ontology language for service (OWL-S), and analyzed its matching degree quantization problems of matching process between ontology concept parameters and constraint parameters. To verify its validity, we have applied it in automobile/motorcycle accessory industry. By combining a quantization method, a matching algorithm and semantic similarity, an automobile/motorcycle accessory ontology database is established. It has fulfilled the semantic inference of the service resources ontology and has realized the valid and rapid intelligent matching of cloud searching.

Keywords Cloud manufacturing (CMfg) · Small- and medium-sized enterprises (SMEs) · Knowledge base · OWL-S · Ontology library · Semantic similarity

✉ Buyun Sheng
shengby@whut.edu.cn

1 School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China

2 School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China

1 Introduction

With the expansion of business scale in small- and medium-sized manufacturing enterprises (SMEs), the significance of the product's data information of designing, production, testing, and management had increased in the past few years [1, 2]. In order to achieve sharing and on-demand of manufacturing resources and manufacturing capacity and improve resource utilization, cloud manufacturing will encapsulate all kinds of resources and the ability into cloud services. Cloud manufacturing resource information is dynamic unstructured information based on heterogeneous environment. Traditional information modeling methods cannot completely describe the manufacturing resource information and realize the service discovery, searching, and matching of cloud manufacturing environment resources.

At home and abroad, document [3] focuses on the interactive interface in the cloud manufacturing (CMfg) [4, 5] environment and analyzes the demand characteristics of the interface of CMfg system. At the same time, the research framework of pervasive Human-Computer Interaction Techniques and the dynamic of the user interface, the customization techniques used to personalize the interface human-equipment interaction technology in the CMfg environment, have been studied. Based on the researches of the hard manufacturing resource virtualization package, the characteristics of hard manufacturing resources, and resource-related virtualization technology, document [6] analyzes the characteristics of hard manufacturing resources and resource-related virtualization technology and clearly puts forward that hard manufacturing resources based on CMfg encapsulate the framework describing in detail by virtualization technology [7–9]. And, on this basis, the process of resources in which virtualization technology is utilized to encapsulate is studied, and crucial questions, such as resource description and matching, are elaborating.

The CMfg to which virtualization technology is utilized can realize overall and deep share in hard manufacturing resources, so that virtual resource cloud pool can become highly scalable and flexible. Document [10] establishes matching model of inputs, outputs, preconditions, and effects of CMfg services which is based on web ontology language for services by the semantic match of CMfg method which is based on ontology language for service (OWL-S) and is preceding application of case by three-dimensional inverse demand of automobile tail light which is digitized. Document [11] is proposed to a description framework of machine tools resource based on web Services Modeling Ontology (WSMO). The key technologies such as metadata ontology representation approach and service-oriented packaging of machine tools resource for realizing this framework are researched. This paper has analyzed the characteristics of the machine tool equipment resource and the method of semantic description in the CMfg environment.

Researches of the current CMfg mainly focus on the concept, pattern features, function system, technical framework, and so on, but multiple heterogeneous manufacturing resources and uniform description which are realized form the prerequisite and foundation for CMfg of circulation and popularization [12, 13]. But, current research in this area is relatively weak.

While, it is impossible for traditional manufacturing service management model to satisfy the increasing transaction and matching of CMfg service resources [14]. So, it is of great necessity to research the intelligent matching of supply and demand, in order to dispatch manufacturing resources rationally and improve the cooperation efficiency among SMEs.

Aiming at these deficiencies, this paper puts out a three-layer structure expressing knowledge (knowledge, the psychological result of perception, learning, and reasoning) name at domain ontology-the index of knowledge-the data resource (OKD) to unify the terminology (terminology is a system of words used to define things in a particular discipline) [15, 16].

CMfg service platform for intelligent matching supply and demand is around the whole life cycle of product manufacturing in this paper. The semantic description of CMfg service according to synonym service resource dictionary from CMfg service itself, subject's capacity of CMfg, manufacturing resources of CMfg and so on by setting up ontology library of manufacturing service resources of industry-classified. Therefore, firstly, this paper theoretically analyzes the algorithm of intelligent matching supply and demand and describes the CMfg service by Web Services Description Language. And, on this basis, this paper establishes services semantic matching model of CMfg service by using modeling approach of bibliographic knowledge. Secondly, model of phases of parameter matching, attribute matching, and integrate matching is analyzed, and then, quantitative methods of matching degree and matching algorithm are proposed in the matching processing. Finally, matching algorithms of semantic similarity and the semantic inference engine are introduced, so that match of keyword and semantic can realize four aspects: function, data, quality of service, and execution. The list of service eventually is made for users to select. It will take function, data, quality, and execution into consideration when searching service resources [17]. With the semantic matching of keywords, a service recommendation list will be formed for users to choose eventually.

2 Overview of supply and demand intelligent matching engine

This intelligent matching engine for supply and demand is studied to integrate the mass enterprise service resources, optimize the internal resources, share the manufacturing resources and manufacturing capabilities, and realize the management, such as intelligent scheduling and configuration, based on knowledge [18].

The active way of intelligent matching for supply and demand is classifying those service resources in a particular manner firstly and then establishing ontology library. Each ontology class, namely concept, will describe the corresponding CMfg service resources with Web Services Description Language (WSDL) document and label all of the parameters in the document with ontology concept. Especially for those demands, they also will be established into the ontology library. Each concept involved in the demand will be defined, and they will map to the related parameter from the service description document [19]. Each parameter in the demand will be described by those concepts from service domain ontology, and a formal demand mapping mode will be built as the demand ontology when users submit the service demand. Based on the above work, the intelligent matching engine can work easily. Parameters from service description document will correspond to the concept in the demand ontology, and they will take a comparison based on semantic. After the comparison, the engine will take the comparison value into a weighted summation and then offers an arrangement about those service resources as the comparison value for users to choose.

From Fig. 1, the supply side and the demand side of the system use service demand information resource function modules described via the Internet cloud. Descriptive text, through a standardized structure model, is standardized in accordance with industry standards and the body, between the text and the final description of conduct intelligence service needs to match supply and demand and access to useful system to meet the needs of users.

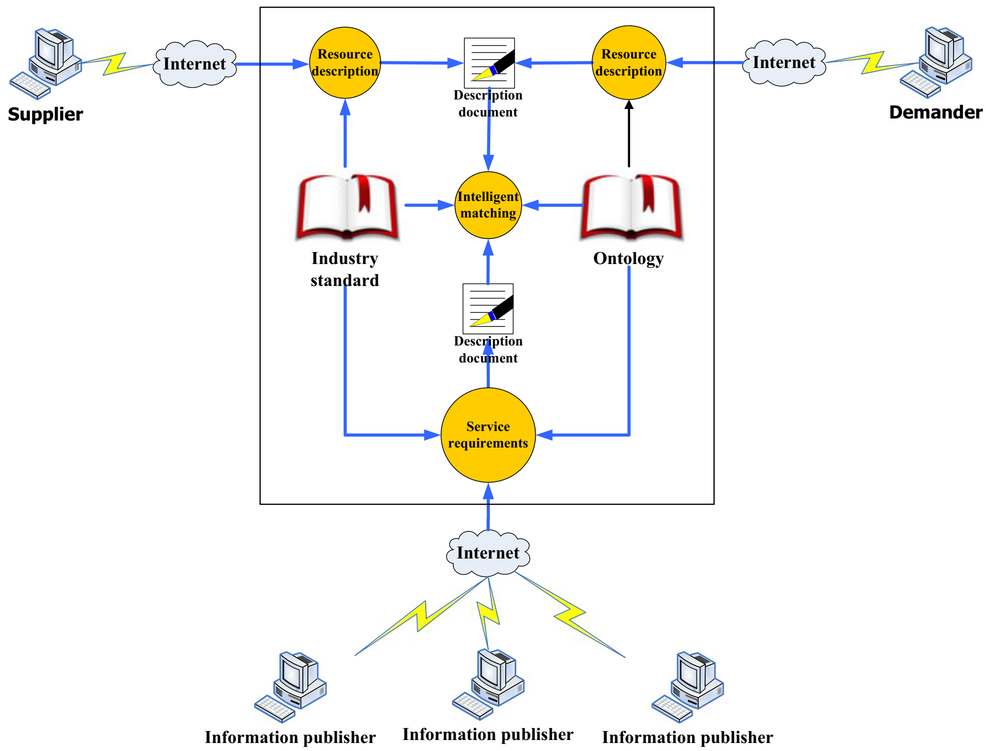


Fig. 1 The schematic of CMfg supply and demand intelligent matching

2.1 Keyword semantic intelligence

Nowadays, many searching methods (such as keyword searching, structure searching, vertical searching, and semantic searching) are used to search for all kinds of manufacturing service widely [20, 21]. Most of these searching engines are based on keywords. It will break the semantic relationship seriously when searching information by those searching method based on keyword. Due to none of semantic information in the searching process, users always cannot acquire satisfying results. Because of a lack of intelligence and knowledge, this searching mechanism is more and more difficult to meet people's growing knowledge demands [22, 23].

The keyword of semantic intelligent searching is used in this intelligent matching engine for supply and demand of manufacturing capability in this paper. The intelligent matching engine depends on efficient searching algorithms to return function, data, and service related to input conditions. With support from this engine, users can obtain results corresponding to input conditions and the best service. Keyword is the word which can generalize related information that users want to search for maximum and it's the generalization and centralization of information. Enterprise users can index manufacturing service resources that they need by keyword searching after logging into the main interface of the CMfg service platform. During the searching process, a searching expansion, namely synonyms and semantic contain extension, is taken to improve the shooting. The basic idea of query expansion is considering the similarity between semantic concepts firstly and then expanding the semantic based on similarity degree according to the classification structure of service resources [24]. The computational process of semantic similarity is as follows:

Definition 1 Assuming that there are two words named as W_1 and W_2, if W_1 has n concepts—$S_{11}, S_{12}, \ldots, S_{1n}$—and W_2 has m concepts—$S_{21}, S_{22}, \ldots, S_{2m}$—then the semantic similarity of W_1 and W_2 is the maximum value of each concept's similarity, namely

$$Sim(W_1, W_2) = \max_{i=1\ldots n, j=1\ldots m} Sim\left(S_{1i}, S_{2j}\right) \tag{1}$$

where $Sim(W_1, W_2)$ is the maximum value of each word's similarity and $\max_{i=1\ldots n, j=1\ldots m} Sim\left(S_{1i}, S_{2j}\right)$ is the maximum value of each concept's similarity.

Definition 2 Where $Sim(S_{1i}, S_{2j})$ is the concept of similarity, the calculation method is

$$Sim\left(S_{1i}, S_{2j}\right) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} Sim_j(S_1, S_2) \tag{2}$$

where S_1 is the concept of the first word, S_2 is the concept of the second word, and β is an influence operator.

The result will be as the follows:

$$Sim(W_1, W_2) = \max_{i=1\ldots n, j=1\ldots m} \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} Sim\left(S_{1i}, S_{2j}\right) \qquad (3)$$

Definition 3 The depth factor of two concepts (S_1, S_2) is $D_{dep}(S_1,S_2)$:

$$D_{dep}(S_1, S_2) = \frac{D_{dep}(S_1) + D_{dep}(S_2)}{\left|D_{dep}(S_1) - D_{dep}(S_2)\right| + 2Depth} \qquad (4)$$

where $D_{dep}(W_1)$ is the hierarchy network depth of concept nodeS_1, $Depth$ is the depth of hierarchical network diagram.

Definition 4 Concept S_1 to conceptS_2's initial semantic similarity is written as$Sim_{orj}(S_1,S_2)$:

$$Sim_{orj}(S_1, S_2) = \lambda_3 \left(\frac{\lambda_1}{\lambda_1 + \log_e(e-1+n) \times \left(\Sigma dist\left(link_1\left(S_i, S_j\right)\right)\right)^{\lambda_2}} \right) + \lambda_4 D_{dep}(S_1, S_2) + \lambda_5 D_{dep}(S_1, S_2) \qquad (5)$$

where S_1 is the concept of the first word, S2 is the concept of the second word. $\beta_i (1 \leq i \leq 4)$is an adjustable parameter,$\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$and $\beta_1 > \beta_2 > \beta_3 > \beta_4$, which reflects that the impact decreases from Sim1 to Sim4 in order in the overall similarity [25, 26]. When the semantic similarity is calculated, the calculation process of concept similarity is added with the greatest weight, because the method taking the concept similarity as the similarity of words is used.

In the above-mentioned intelligence semantic search algorithms, it is the most important to establish a strong enough service resources dictionary full of concept synonyms. In the CloudManu platform, the combination of those three classification methods (service resource type, engaged in the production industries, and transaction types) and knowledge ontology technology about service resources will realize the active matching of supply and demand finally.

2.2 Building synonym concept dictionary

In order to overcome the shortage of keyword searching, this method applies synonym concept dictionary to realize the specification, extension, and query of searching question from users and understand their search intention with the concept searching. The synonym concept dictionary always contains many professional words defined by experts with their hands or correlation concept space generated through machine learning [27]. Service resources synonym dictionary is a collection of main synonyms about each keyword, while the main synonym is the corresponding attribute of keyword. With the service resources synonym dictionary, this CMfg service system will shift each keyword used to describe service request to be service resources properties automatically [28]. So, a semantic similarity algorithm is imported to take a keyword semantic matching from the service name, credit rating, transaction data, and execution process of those service resources in this matching engine [1]. The service resources concept description related to the maximally cloud searching result will be listed in the service recommendation list as the comprehensive factor of the above aspects, so users can find satisfied service resources easily.

The computation of semantic similarity based on the concept of domain ontology is as follows:

Definition 5 If X and Y are any two concepts in the ontology hierarchy tree, so the semantic distance between the two is "Distance(X, Y)" [29]. In addition, Level(X) is the layer in which concept X is set in the ontology and Level(Y) is the corresponding layer of Y, so the gap between them is |Level(X) −Level(Y)|.

On the basis of the above two definitions, a formula to calculate the similarity between any two concepts, X and Y, in the domain ontology is proposed, and the formula can be described as the follows:

$$Sim(X, Y) = \begin{cases} 1, X=Y \\ \frac{\alpha}{(\text{Distance}(X,Y)+\alpha)} \times \frac{\beta}{|NodeSet(X) \cup NodeSet(Y)|} \times \frac{|NodeSet(X) \cup NodeSet(Y)|}{\left(\gamma \times \left|\text{Level}(X) - \text{Level}(Y) + 1\right.\right)}, X \neq Y \end{cases} \qquad (6)$$

When X equals to Y, namely $X=Y$, the calculation result will be "1," namely $Sim(X, Y)=1$, which means that the similarity between the two is 1; however, when X is not equal to Y, namely $X \neq Y$, the result will be as the follows:

$$Sim(X, Y) = \frac{\alpha}{(Distance(X, Y) + \alpha)} \times \frac{\beta}{|NodeSet(X) \cup NodeSet(Y)|} \times \frac{|NodeSet(X) \cup NodeSet(Y)|}{\left(\gamma \times \left|Level(X) - Level(Y) + 1\right.\right)}, X \neq Y \qquad (7)$$

where α is an adjustable parameter, the value of α can reflect the relationship of semantic similarity, and the value is a positive real number. α is an adjustable parameter, and β is used to adjust the influence degree that value of the coincidence over the similarity. The value of β in the range of (1, $D_{dep}(S_1,S_2)$)

and $D_{dep}(S_1,S_2)$ indicates the level of depth of the body. Ontology semantic overlap of the largest species of the two concepts is the largest body in the tree hierarchy node and its parent, semantic overlap; this value range β can guarantee the case of $X \neq Y$, Y is an adjustable parameters for regulating the level of poor concept of similarity, and Y ranges generally between (0,1).

Definition 6 If X and Y are any two nodes in the ontology concept correlation graph, $ShortestPath(X,Y)$ is the shortest route from X to Y. When X, Y is not connected, the value of $ShortestPath(X,Y)$ is∞.

Based on the above-proposed formula, a method to measure semantic association of any two concepts in ontology is as follows [30, 31]:

$$\mathrm{Re}l(X,Y) = \begin{cases} 1, X=Y \ or \ X,Y \ Equivalence \\ \frac{\lambda}{ShortestPath(X,Y)+\lambda}, Other \ \mathrm{Relationships} \end{cases} \quad (8)$$

where $\mathrm{Re}l(X,Y)$is the relevance between any two concepts in the domain ontology is, λ is an adjustable parameter, and $\beta_1+\beta_2+\beta_3+\beta_4=1$, namely the correlation of 0.5, is the shortest distance between the values of concepts.
where $Sim_\mathrm{Re}l(X,Y)$is combined with similarity and correlation comprehensive relationship, the calculation method is as follows:

$$Sim\_\mathrm{Re}l(X,Y) = Sim(X,Y) + \mathrm{Re}l(X,Y)\text{-}Sim(X,Y)*\mathrm{Re}l(X,Y) \quad (9)$$

Among them, $Sim\_\mathrm{Re}l(X,Y)$ is the integrated relationship of X and Y concept value, $Sim(X,Y)$ is the concept similarity of concept of X and Y, and $\mathrm{Re}l(X,Y)$ is the correlation of concept of X and Y.

3 Key issue discussion

CMfg service resources contain all the resources in the operation process and product life cycle from manufacturing enterprises. It is required to build an index knowledge base to establish the CMfg service resources ontology according to the requirement characteristic of small- and medium-sized enterprise. The index knowledge base is used to manage the knowledge characteristic and knowledge content [32, 33]. It can also provide knowledge link, but it does not manage the content and structure of the service resources themselves. Therefore, all kinds of resources can be accessed seamless without changing the structure of ontology knowledge base. We can generally know that new service resources ontology through the description and then determine whether to open the corresponding data resources fatherly.

In the building process, WSDL documents are used to specify the description, so as to index service data and take intelligent matching quickly.

3.1 Building index knowledge base

All kinds of resources need to be standardized before registration, which can contribute to the intelligent matching of supply and demand under the CMfg service environment. Index knowledge is applied to guide the service resources searching. On one hand, the definition of index knowledge content and structure works for the structure and lightweight of service resources data. On the other hand, the knowledge description of service resources data based on domain ontology realizes the semantic association between those service resources, thus improving the rapidity and accuracy of the searching.

Index knowledge contains lots of knowledge to design and develop; it can reflect the content of work flow about transactions between supply and demand at the same time. When dealing with the whole design document and product model of CMfg service resources, the following three tipples will be used to describe the index knowledge [9, 34].

$$\mathrm{IK} = \ < \mathrm{AD}, \mathrm{KD}, \mathrm{URL} >$$

in which, IK means index knowledge, AD means the description of knowledge attribute, KD means the description of knowledge, namely the relevant design knowledge of service resources, and URL means the address of knowledge.

AD and KD usually be described as the following:

<AD> {Name of the service resource, id of the service resource, type of the service resource, creation data, founder, audit, type, field, category, and key sets}

<KD> {Task goal, task description, input and output, solution, design experience, and description of service resource model}

The modeling of index knowledge is the process to extract those new service resources, and the building steps are shown in Fig. 2.

According to specification, the term ontology knowledge base has been established for the new service resources description text word processing documents. After being processed, ontology's service resources use new markup tools, which will draw automatic domain terms, and in accordance with TF-IDF algorithm (inverse document frequency, term frequency-inverse document frequency), automatic document features extraction of terms, terminology, and knowledge indexes' key set documentation of word document features. Among them, the feature set of words is one of the main fields of subsequent knowledge index matching [35, 36]. In the ontology knowledge library, the new service resources markup tool will automatically propose document-related attribute

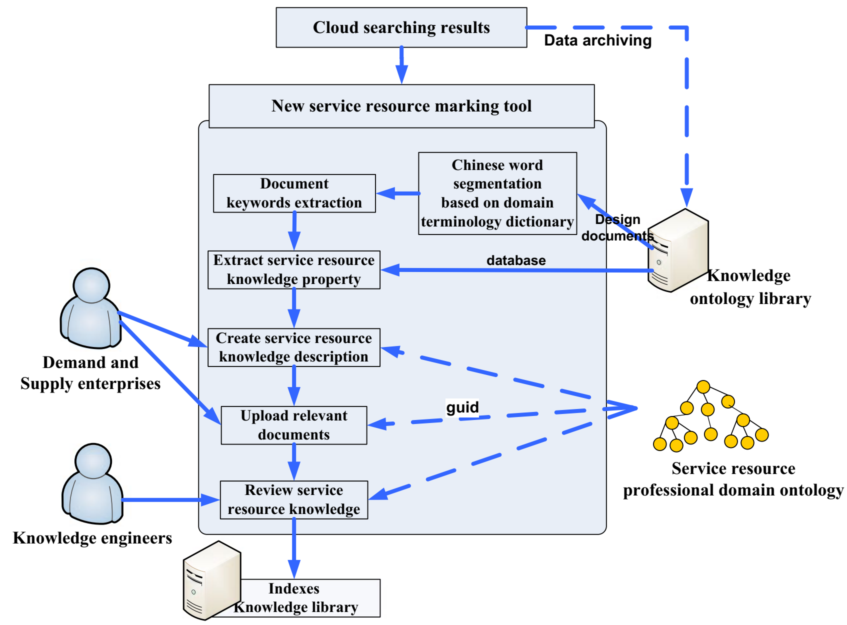


Fig. 2 Index knowledge library process diagram

information. To ensure the knowledge and stylistic unity of the description of a term, the new service resources markup tools feature set of words based on the service resources description document annotation knowledge and fill in the new knowledge-related resources and services (including the description of the service resources, solutions, and design experience knowledge sets), so as to form an index of knowledge. Therefore, it is not only describing the development knowledge of service resources description document, also allowing that both sides of supply and demand are obtained understanding and experience by the form of natural language description in the process of publishing tasks [37]. Finally, the knowledge engineer will review the legality and validity of knowledge. You have the ability to decide whether you can put in storage.

The manufacture resources attribute classification of cloud service platform and the classification map are shown in Fig. 3.

In the process of development, we integrate from above manufacturing resources properties that describe the unique attributes. As shown in Fig. 4, it is the property ontology meta-model of CMfg service resources:

3.2 Building characteristic vector model of searching words and index knowledge

After the semantic parsing of service resources, index knowledge of vector space model is used to matching algorithm. According to the ontology concept collection, ontology relationship weight and business demand knowledge weight are input to build characteristic vector model of searching words and index knowledge. The ontology relationship weight of service resources is used to distinguish the important degree of each attribute description in keywords to the searching, so as to acquire a more comprehensive cloud search result and improve the recall rate.

1. The characteristic vector expressing of search keyword sets

Generally speaking, keyword from original problem description will reflect the demand of supplier and demander. Thus, the highest weight original keyword system can give automatically “1.0” in the searching word sets, while the ontology concept relevance of extension words is given different weights. At the same time, developer can also give the corresponding weight according to their needs when searching. The design problem is converted to vector V_q eventually [38–40].

$$V_q = \left\{W_{q1}W_{o1}, W_{q2}W_{o2}, \ldots, W_{qi}W_{oi}, \ldots, W_{qn}W_{on}\right\} \quad (10)$$

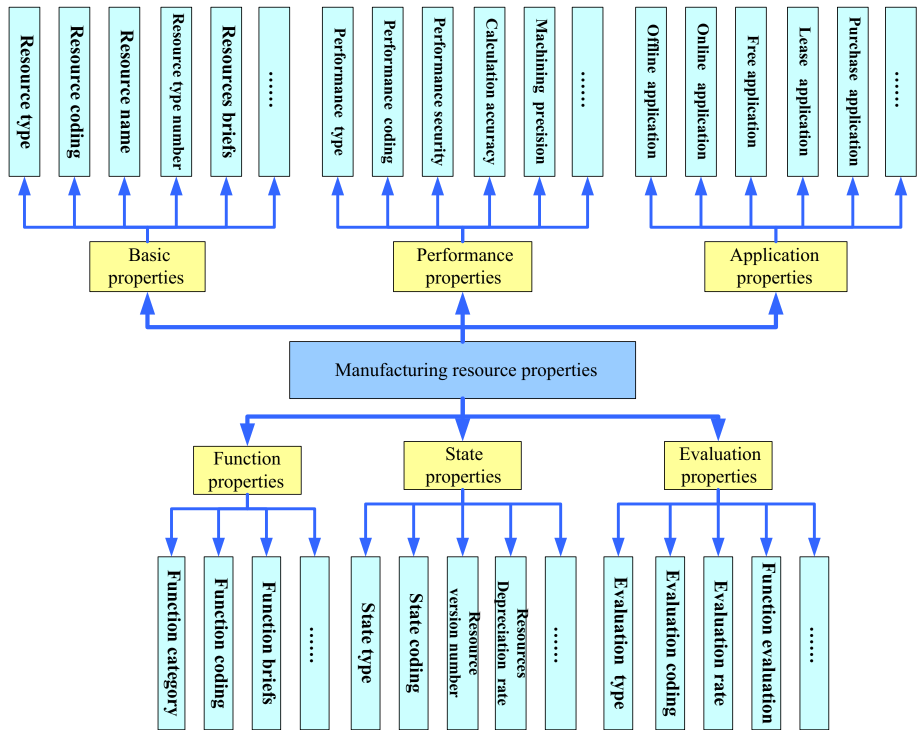
In addition, W_{qi} is the characteristic weight of term i in the searching word sets q, while W_{oi} means the ontology relationship weight between searching word i and the original keyword.

2. The characteristic vector expressing of index knowledge

Searching words may appear in different contents of one index knowledge, and different descriptions are put with different important degree, so it is necessary to distinguish the important degree of each content to improve recall rate.

Designing knowledge domain weight refers to giving different weights to different contents of index knowledge. Each content corresponds to a domain weight W_k

Fig. 3 The classification map of manufacturing rescore diagram

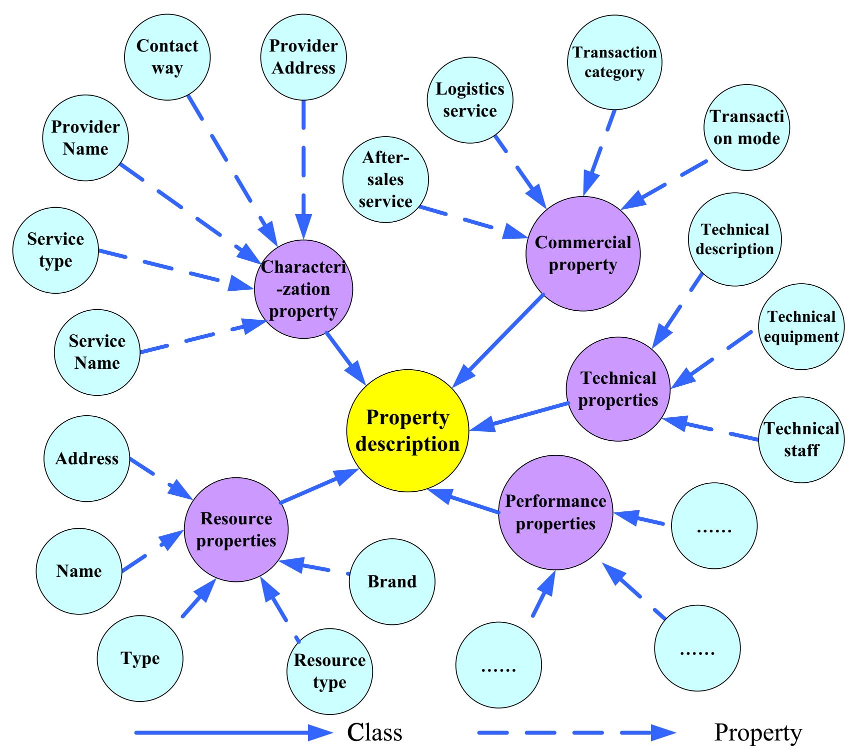


[41]. The higher is the value of W_k, the more important is the content of this index knowledge.

$$V_d = \left\{ W_{d_1} \cdot \prod_{j=1}^{m_1} W_{k_j}, W_{d_2} \cdot \prod_{j=1}^{m_2} W_{k_j}, \ldots, W_{d_i} \cdot \prod_{j=1}^{m_i} W_{k_j}, \ldots, W_{d_n} \cdot \prod_{j=1}^{m_n} W_{k_j} \right\} \quad (11)$$

In addition, W_{d_i} means the characteristic weight of the word in the index knowledge d, $\prod_{j=1}^{m_i} W_{k_j}$ means the knowledge domain weight that the word get, W_{k_j} means the domain weight of the content in the index knowledge, and m_i means

Fig. 4 CMfg service resources property ontology meta-model diagram



the occurrence number of the word in the index knowledge [42].

3.3 Index knowledge matching algorithm based on vector space model

On the basis of the using of index knowledge, synonyms, and semantic contain query, this paper takes a comparison of the whole similarity by semantic similarity matching algorithm [43]. At present, vector space model (VSM) is a mature Smart text retrieve system. Its design idea is as follows: creating multiple vector (W_1, W_2, W_3,……W_q) and assuming W_i as the weight of the key item firstly and then automatically extracting document key item to find the solution by the TF-IDF algorithm, calculating the cosine value between searching field and papering searching field of the keyword in vector space. If the value is more close to 1, the similarity will be higher.

$$W(t,d) = tf(t,d)\cdot idf(t,d) = \sqrt{n_t, d}*\{1+\log[N/(N_{t,d}+1)]\} \quad (12)$$

In addition, $W(t,d)$ is the t weight of key item in document term, $f(t,d)$ is the word frequency of document collection in all the document term, n_t,$d$is occurrence number of document in althea document, $idf(t,d)$ is the frequency of the inverse document, N means the number of all document, and $N_{t,d}$ means the document number of occurrence items.

3.4 The implementation of reasoning logic description

According to the specific requirement of web services capacity constraint description, the constraint conditions of predictions and effects are divided into two categories. One category is the numerical constraints; namely, the constraint is described by using numerical type to the web services outside the state of the world or input and output parameters; the other category is the object-type constraints; namely, the constraint is described by using object type to the web services outside the state of the world or input and output parameters of the hierarchical domain ontology in the class definition of the class and attribute property [44]. For example,

Service provider:

P $partsType(?x)\wedge shaftParts(?x)\wedge cylindricity(?y)\wedge smallORequal(?x,9)$

Service requester:

R $partsType(?x)\wedge steppedShaft(?x)\wedge cylindricity(?y)\wedge smallORequal(?x,10)$

By using OWL reasoning engine, reasoning logic derive out *steppedShaft* contains*shaftParts*. Then, web services capabilities needed constraints on the object-type service providers to meet the requirements, but the numerical constraints are not directly through the concept of OWL reasoning engine (including relationships); therefore, ontology web services capabilities in the field of numerical reasoning require a separate process. Therefore, according to numerical constraints and object constraints, reasoning logic needs to use different inference algorithms for matching. For numerical constraints, the method of domain ontology inference rule consistency constraints between reasoning. In ontology, numeric constraints matching rules are described to SWRL language. Thus, the rules and OWL domain ontology may be separately expressed in the rule-based reasoning system. For matching object-type constraints, domain ontology decryption logic inference engine consistency judgment.

1. Matching algorithm of numerical constraints

Define the numerical constraints in the form of the following:

$$numberCondition = \{numberTerm_1, numberTerm_2, \ldots, numberTerm_n\}$$

where *numberTerm*is a$r(x,v)$form of two predicate logics and x is an adjustable parameter. This variable can be defined as an instance of the class OWL in domain ontology, and $v$is a specific value, which $r$are the four predicate numerical constraint relations. Now, we define that the four kinds of numerical constraints described predicates from the equal (*Equal*), greater than or equal to (*LargeOrEqual*), and less than or equal to (*LessEqual*), within the scope of numerical constraints (*RangeIn*) [45]. The inference rules are defined as follows:

Definition 7 $Equal-Consistent$:

$$Equal(\mathrm{Provider}, ?x), Equal(\mathrm{Requester}, ?y), Equal(?x, ?y) \Rightarrow Equal-Consistent$$

Definition 8 $LessOrEqual-Consistent$:

$$LessOrEqual(\mathrm{Provider}, ?x), LessOrEqual(\mathrm{Requester}, ?y), LessOrEqual(?x, ?y), LessOrEqual-Consistent$$

Definition 9 $\mathrm{Large}OrEqual-Consistent$:

$$\mathrm{Large}OrEqual(\mathrm{Provider}, ?x), \mathrm{Large}OrEqual(\mathrm{Requester}, ?y), \mathrm{Large}OrEqual(?x, ?y) \Rightarrow \mathrm{Large}OrEqual-Consistent$$

Definition 10 *RangeIn−Consistent*:

$$RangeIn(\mathrm{Pr}ovider, ?x, ?y), RangeIn(\mathrm{Re}quester, ?z, ?w), LessOrEqual(?z, ?x), \mathrm{Large}OrEqual(?w, ?y) \Rightarrow RangeIn{-}Consistent$$

Among them, the Pr*ovider* and Re*quester*represent the service provider and requester both in describing the variables used in constraint conditions. The *x*,*y*,*z*, and *w*are concrete constraint values.

Note that we are reasoning that is based on the precondition predicate constraints matching the description. The match constrained parameters are the first to match and predicate in the constraint condition. If the predicate is not an exact match, the match constrained parameters will not be matched; if the predicate is an exact match, then the match constrained parameters will be again to match each individual word [13]. Of course, the field of ontology predicate variable in line with consensus, then it should also be semantically consistent (Table 1).

The matching relationship between numerical constraints can be judged by reasoning from the above rules, and then, the algorithm is described as follows:

```
Algorithm 1    matchingmaking:    numberConditionProver
        Input:    number Condition-Requester, number Condition-Provider
                Output：    Result-True or False.
                 Result=Emptiest
           Th=threshold
For each number Term in number Condition-Provider do
        If ∃  number Term in number Condition-Requester
        Where  r  is  equal,  x  is  semantic  consistent  and  number
Rule-Consistent
              Then
           Return Result-True
       Else
  return sort(Result-False）
```

2. Matching algorithm of object constraints

Object-type constraints matching algorithm thought is changing the Horn clause semantics of the object constraints matching algorithm (include Horn clause) into a constraint satisfaction problem (CSP) [46]. In the transformation process, make full use of the semantic relations between concepts of domain ontology as the semantic heuristic information, reasoning logic try to simplify the complexity of CSP, and then use a variety of optimization algorithms of CSP field to determine compatibility between constraints.

The matching algorithm of object-type constraints is mainly divided into three steps:

First, this algorithm is normalized in a constraint clause; namely, the constraints are exhumed into the semantic information of the expression contains and then exhumed into OWL ontology knowledge and description language (DL) reasoning machine. The semantic information normalization is associated with specific DL. Generally speaking, the more powerful the DL language, the more complicated the normalization process of semantic information; simply, we mainly consider the class, property, and the implied semantic information of the property domain and range of the OWL language in the current semantic normalization.

Second, the semantic θ (contains problems between conjunctive normal form clauses) is transformed into a CSP problem. It also exist a range of values of the variables which predicate containment relationship between the class of the OWL as variable substitution, and the existence of binary predicate contains relations (OWL property) between variables to as constraint conditions; thus, this put the replacement of variables into CSP problem. By using semantic information of domain ontology, matching algorithm is combined with the method of node. The consistency and Arc, consistency of the CSP field, reasoning

Table 1 Matching algorithm of numerical constraints

Term Provider2N (numerical constraints type)

Degree of match		Equal (相等)			LargeOrEqual (大于等于)			LessEqual (小于等于)		
(provider, requester)		≥	≤	=	≥	≤	=	≥	≤	=
Term provider1	Exact matching (term requester1)	1.0	1.0	1.0	1.0	0.8	0	0.8	1.0	0
	Compatible matching (term requester1)	0.7	0.7	0.7	0.7	0.5	0	0.5	0.7	0
	Contains matching (term requester1)	0.4	0.4	0.4	0.4	0.2	0	0.2	0.4	0
	Mismatching (term requester1)	0	0	0	0	0	0	0	0	0

logic is reducing to the complexity of the corresponding CSP problem. Lastly, this method can greatly reduce the search area of the variable displacement.

Third, the rules adopted some optimization algorithms in the CSP field. Within a range of the predicate form, search to satisfy all the constraints from the binary predicate variable replacement; then, DL reasoning machine reused clauses between text semantic reasoning.

3.5 The linguistic match of intelligent matching engine

Criteria semantic matching (linguistic match) is based on the above reasoning judgment out of service resources, to evaluate them in the linguistic sense of the word meaning similarity. Therefore, the establishment of the criteria semantic matching is a practical significance in the name of the semantic concepts. Semantic matching algorithm has used a variety of names and a string of matching methods, including the use of common conceptual similarity judgment NG algorithm, using the Word Net Dictionary to synonym judgment, and calling the abbreviation for judging the thesaurus. Alternatively, you can make essential pretreatment when those entries that match a particular query string, when you use the matching algorithm, such as conversion of punctuation and uppercase and lowercase letters; filtering out the stop words, stem extract, or substring; removing the affix; and so on. After the pretreatment methods and various matching algorithm executions, if any kind of algorithm to get the match exactly results (matching degree is 1), the two concepts are considered as exact match [18, 47]; if all the algorithms do not match the results (matching degree is 0), then the two concepts do not match; otherwise, it is a non-zero average value as the matching degree of matching concepts. Through the analysis, the distance between two main concepts X, Y can be calculated by the distances between the nodes:

$$LingMS(wc,oc)=\begin{cases}1 & \text{if}\left(ms_1\vee ms_2\vee\ldots\vee ms_n=1\right)\\ avg(ms_i) & \text{if}\left(ms_i\neq 0, i\in I\right)\wedge\left(ms_j=0, j\notin I\right)\\ 0 & \text{if}\left(ms_1=ms_2=\ldots=ms_n=0\right)\end{cases} \quad (13)$$

where Wc(WSDL concept) is on behalf of WSDL concepts, Oc (ontology concept) is on behalf of the ontology concepts. Wc and Oc's match scored are obtained for different matching algorithms.

4 Case study

In this article, in order to match supply and demand initiative, related concept synonym service resources dictionary is built to service resources including car/motorcycle lamps, mirrors, plastic parts and general machinery parts, and so on. We took a verification application in the automobile headlight.

4.1 Background

Auto parts' whole process of new product development relates to software resources such as sample data scanning, image ware point cloud processing, UG reverse design, UGCAM machining, NC code programming, etc. and a large number of hardware resources including scanning equipment, machine tools, molds, etc. [48]. At present, these service resources are scattered in different places and each point, and most of these companies can offer only partial auto parts product development services. But, new product development of auto parts is a continuous, consistent process from product reverse designing, researching, and designing to manufacturing.

4.2 Query expansion process

This paper will establish a new service resources ontology library of the automobile and motorcycle spare parts product research and development, developed by Stanford University's protégé.

To verify the Smart Matching of CMfg service and feasibility and practicality of process traceability, we manufacture in the early development of the motorcycle parts companies leaked processing support system, combining with the above-mentioned research ideas and related outcomes [49–51]. This paper makes local improvement and updates the modeling, description, registration, publication, cloud search, service capabilities, and other aspects of cloud system in automobile and motorcycle parts new product resources and improves the system to have attribute analysis, resources modeling, semantic description, and service of packaging and other function. Therefore, the system overcomes this kind of automobile and motorcycle parts itself functional new product development complexity and, meanwhile, initially meets the automobile and motorcycle parts smart match of demand and supply of new products manufactured under a cloud environment, polymerization of process knowledge, and other requirements of application.

Automobile lamp ontology includes car lamps class, mirrors class, plastic parts class, and general machinery parts class, etc., and the car lamp class also includes lighting lamp and signal lamp two subclasses: lighting lamp subclass also includes headlamp, reversing lamp, door lamp, etc, and signal lamp subclass includes turn signal lamp, operating turn signal lamp, turn signal lamp, etc. When establishing a subclass, we want to create restrain, object properties, and data properties. At the same time, the subclass' instance will be created. As shown in Fig. 5, it is the automobile lamps instance Jambalaya diagram of CMfg service resources:

In the figure, "Flashlight" and "Signal lamp" are in the same class and belong to a car light subclasses, "Flashlight" and "Signal lamp" subclasses' semantic value has a weight of 2. The second bottom of the "Head light" and "Reversing lamp" are respectively corresponding to the subclasses of "Flashlight" and "Signal lamp." The semantic distance between the two concepts is far away from the root and can be set to 3.

The input, output, precondition, and effect (IOPE) matching model of headlamp service providers provider (abbreviated P) and demanders requester (abbreviated R) is R=<RInput, ROutput, RPrecondition, REffect>. In the automotive headlamp parameters corresponding to each lamp attribute as an example, RInput={Head light, Three-- dimensional reverse engineering service, Three-dimensional optical scanner+Image ware software, Dimension error, Angle deviation}, ROutput={Class A Engineering}, RPrecondition={Chongqing city}, PEffect={Dimension error≤0.01, Angle deviation≤0.5}[29]

Headlamp service providers' IOPE matching model is P= <PInput, POutput, Precondition, PEffect>, where the parameters corresponding to each attribute are as follows: service release parameters are Input={Retrofit Design service, Three-dimensional optical scanner, UG, Institute for Manufacturing Engineering, Dimension error, Angle deviation}, POutput={Class A Engineering}, Precondition={Chongqing city}, Effect = ={Dimension error≤0.01, Angle deviation≤0.5};

According to the using of index knowledge, synonyms, and semantic contain query retrieval algorithm, the model

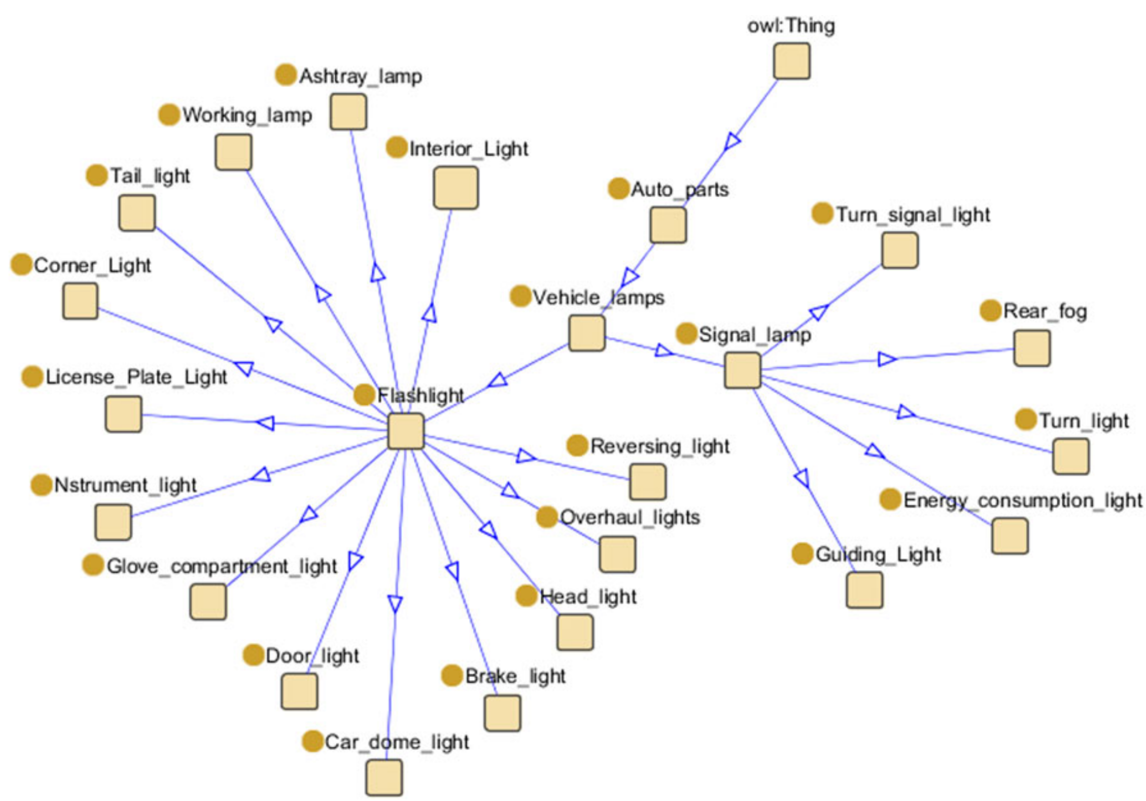


Fig. 5 Car lamp body Jambalaya diagram

Table 2 Matching degree and I/O property matching degree of ROutput and P

Property	Matching condition	Matching degree	I/O property matching degree
	(Automobile tail light, automobile reverse)	0.96	0.90
	(Three-dimensional optical scanner, articulated arm measuring instrument)	0.92	
Input	(Image ware, UG)	0.90	
	(CATIA, UG)	0.90	
	(Dimension error, dimension error)	1.00	
	(Angle deviation, angle deviation)	1.00	
Output	(Class A engineering, class A engineering)	1.00	1.00

IOPE Input, Output, Precondition, Effect four parameters are calculated and calculate its data of 0.9,1,0.9,1.

And, automobile headlamps providers' and requesters' comprehensive matching degree: matchingmaking=0.9+1+0.9+1=3.8, namely=3.8.

In view of the above parameter values by formula (5), that is,

$$Sim(\text{“Head light”},\ \text{“Lighting lamp”}) = \frac{3\times 3.8\times\beta}{5\times(2+\beta)} \quad (14)$$

$$Sim(\text{“Head light”}, \text{“Reversing light”}) = \frac{1\times 3.8\times\beta}{3\times(2+\beta)} \quad (15)$$

Because the α and β are positive, then *Sim*("Head light" and "Lighting lamp")>*Sim*("Head light," "Reversing lamp"), then the depth of reaction has effect on similarity (Table 2).

In the formula, we import keywords into the cloud searching box to calculating the similarity algorithm. If the value is less than a certain numerical value, we can deduce all relevant service resources in the formula.

4.3 Process of intelligent matching for supply and demand

In the development of CloudManu System, developers can also adopt the MVC hierarchical model, because a rational system architecture design hierarchy can not only guarantee the quality of development, but also help to improve the development life cycle, which is convenient for division of labor. Based on the project, this paper uses service resources network intelligence to create new models to achieve CMfg runs

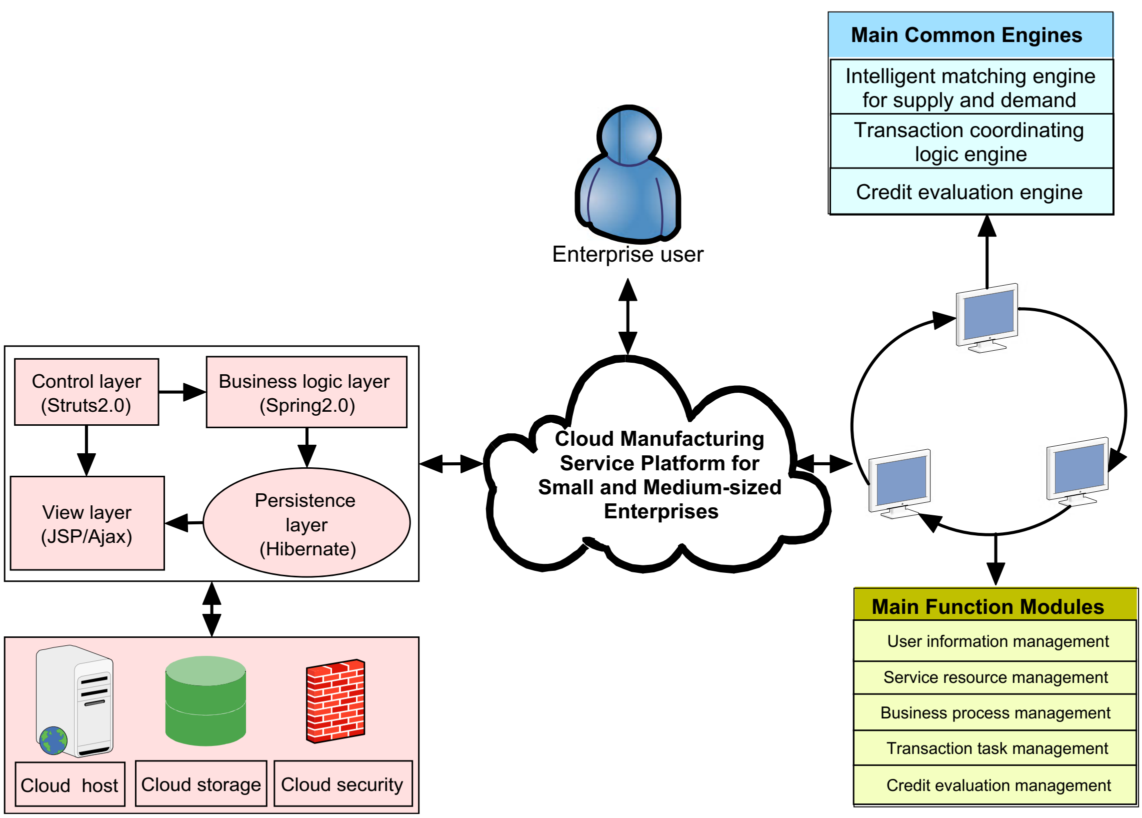


Fig. 6 Platform architecture diagram

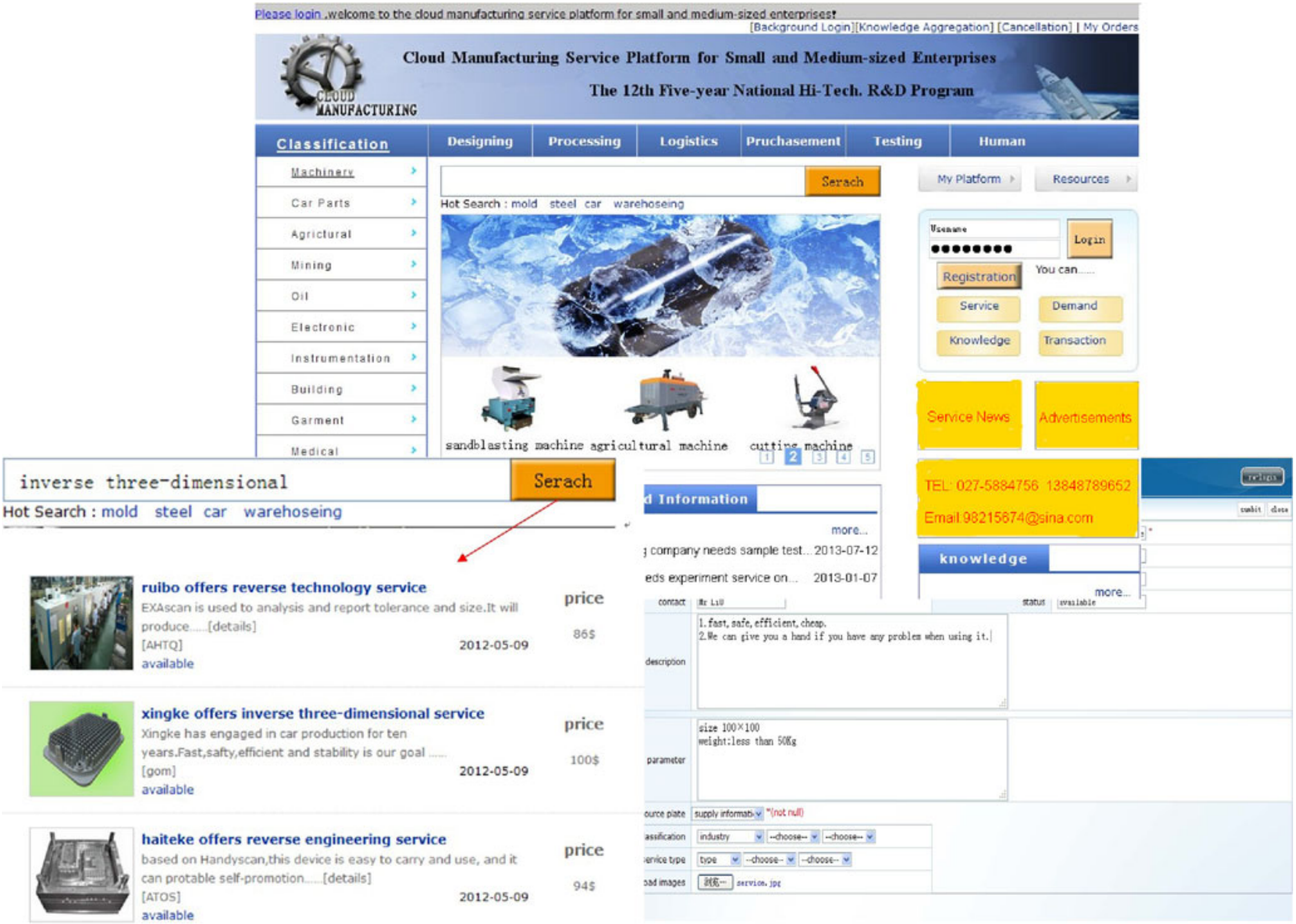
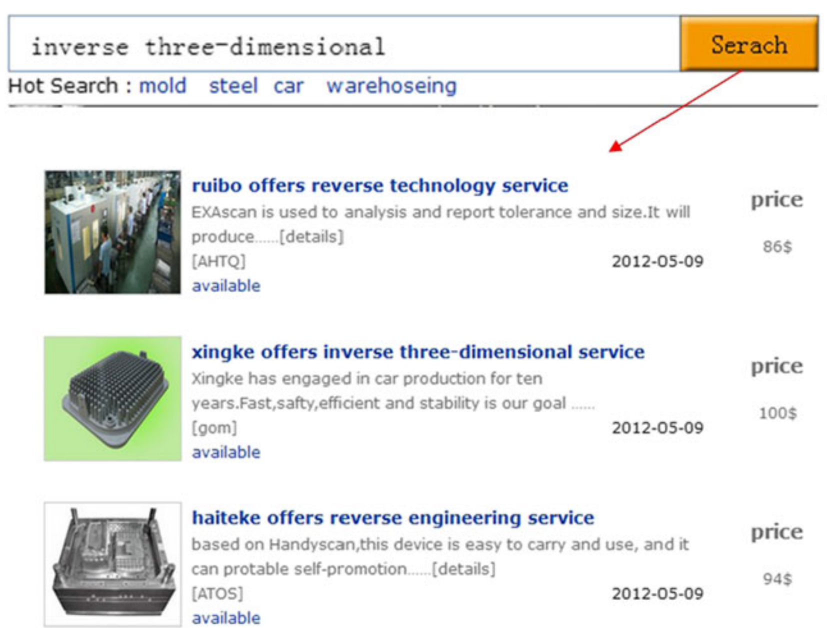


Fig. 7 Manufacturing resources cloud search diagram

of a number of manufacturing enterprises with more orders for collaboration [52–54]. However, it needs relevant modification if the existing system wants more enterprises and more orders. Using multi-collaboration can reduce the

Fig. 8 CMfg resources semantic matching diagram



risk of enterprises, and it needs to authorize the operation for business users, which is not realistic and very complex in the existing service platform. In order to let both sides of the supply and demand obtain the required orders quickly, we design platform framework (Fig. 6).

The keyword semantic matching will take function, data, quality, and execution into consideration, when users search service resources in the CloudManu system. With the matching algorithm of semantic similarity, those records according to the searching conditions will form a service recommendation list for users to choose. The manufacturing resources cloud search diagram may be described with Fig. 7.

When they enter in the CloudManu system, users from both supply and demand can input keywords of manufacturing service resources that they need into the cloud searching box, and then, they will get the best service resources after semantic intelligence computing and matching calculating by the platform itself (Fig. 8).

In the process of semantic matching, the similarity value from similar concept calculation can achieve the searching requirement. While, it is impossible to avoid the relationship extension between concepts based on the query expansion of relevance.

Therefore, based on the above calculation method, quantitative results of comprehensive relationship between concepts will store into the semantic matching process. When indexing data by cloud searching, it will acquire the query expansion by looking for similarities and relevance between concepts and send it to the searching engine eventually [55].

In the above ontology's, the query expansion of relevancy includes the node calculating between upper and lower relationship and relevant concepts.

The following is a part of OWL description about automobile parts ontology instance:

```
<rdf:Description>
    <rdf:Description rdf:about="# Service Type ">
      <rdfs:subPropertyOf rdf:resource="#  Representation property"/>
  <rdf:typerdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="# Flashlight ">
<rdfs:subClassOf rdf:resource="# Head light "/>
      <rdfs:subClassOf rdf:nodeID="A0"/>
         ……
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
```

5 Conclusions

CMfg is an important means for the transformation and upgrading of traditional manufacturing to service-oriented manufacturing. The Smart Matching of CMfg service is one of the key technologies that need to be broken through. The function of Smart Matching is powerfully determines whether the platform can search and automatic match the results that we expect conveniently and powerfully. Meanwhile, the Smart Matching of CMfg service also shows great influence on the popularity's function experience. However, the study of CMfg just starts the research of search and matching in the CMfg services remains to be enriched and improved. Besides, the CMfg service's search and word segmentation, description of making manufacturing resources synonym to service resources dictionary, the construct of domain ontology library, and the semantic similarity's matching algorithm and optimization of semantic reasoning and other aspects need to further study. The paper studied the matching process' diversity, heterogeneity, and multi-constraints between the intelligent matching engine and cloud manufacturing (CMfg) service, then established an intelligent searching engine of CMfg service in SMEs which is based on OWL-S, and analyzed its matching degree quantization problems of matching process between ontology concept parameters and constraint parameters.

In the next stage, we will further improve the manufacturing service resources ontology library, focusing on studying related intelligent matching similarity algorithm. We will devote time to promoting the cloud manufacturing service model, so as to provide better services for small- and medium-sized enterprises.

Acknowledgments This work is partially supported by the Chinese National Hi-Tech R&D Program under grant 2011AA040504. Within this group project community, several PIs and co-PIs give valuable suggestions on this work. Authors would transfer special thanks to Professors Tingxin Song, Biqing Huang, Chao Yin, and Fei Tao.

References

1. Huang BQ, Li CH, Yin C, Zhao XP (2013) Cloud manufacturing service platform for small and medium-sized enterprises. Int J Adv Manuf Technol 65(12):1261–1272
2. Li BH, Zhang L, Chai XD (2010) Introduction t o cloud manufacturing. ZTE Communications 16(4):5–8
3. Ma CX, Ren L, Teng DX (2012) Ubiquitous human-computer interaction in cloud manufacturing system. Comput Integr Manuf Syst 17(3):504–510
4. Tao F, Zhang L, Venkatesh VC, Luo Y, Cheng Y (2011) Cloud manufacturing: a computing and service-oriented manufacturing model. J Eng Manuf 225(10):1969–1976
5. Tao F, Zhang L, Liu YK, Cheng Y, Wang LH, Xu X (2015) Manufacturing service management in cloud manufacturing: overview and future research directions, J Manuf Eng Trans ASME 137(4):040912-1-040912-11
6. Wu J, Li SB (2014) The hardware resource virtual encapsulation in cloud manufacturing. Mach Des Manuf 7(7):112–115
7. Tao F, Cheng Y, Zhang L, Nee AYC (2015) Advanced manufacturing systems: socialization characteristics and trends. J Intell Manuf. doi:10.1007/s10845-015-1042-8
8. Tao F, Zuo Y, Xu LD, Zhang L (2014) IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. IEEE Trans Ind Inf 10(2):1547–1557
9. Tao F, Zhao DM, Hu YF, Zhou ZD (2008) Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. IEEE Trans Ind Inf 4(4):315–327
10. Yin C, Xia Q, Li ZW (2012) Semantic matching technique of cloud manufacturing service based on OWL-S. Comput Integr Manuf Syst 7(7):1494–1502
11. Li XB, Yin C, YFN S (2014) Semantic description and characteristics of machine tool resources in cloud manufacturing environment. Comput Integr Manuf Syst 20(9):2064–2171
12. Tao F, Li Yuan J (2013) FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system. IEEE Trans Ind Inf 9(4):2023–2033
13. Zhang L, Luo YL, Tao F (2012) Cloud manufacturing: a new manufacturing paradigm. Enterp Inf Syst 8(2):167–187
14. LI BH, Zhang L, Ren L et al (2011) Further discussion on cloud manufacturing. Comput Integr Manuf Syst 17(3):449–457
15. KUNAL V, KAAPTHIK S, AMIT S et al (2005) METEOR-SWSDI: scalable P2 infrastructure registries for semantic publication and discovery of web services. J Inf Technol Manag 6(1):17–39
16. Peng H, ShiI ZZ, Qiu LR et al (2008) Matching algorithm of semantic Web service based on similarity of ontology concepts. Comput Eng 34(15):51–53
17. Song TX, Liu HM, Wei CM et al (2014) Common engines of cloud manufacturing service platform for SMEs. Int J Adv Manuf Technol 73(1):557–569
18. Li BH, Zhang L, Wang SL et al (2010) Cloud manufacturing: a new service -oriented networked manufacturing mode. Comput Integr Manuf Syst 16(1):1–7
19. Valentina D (2014) Description of a semantic-based navigation model to explore document collections in the maritime domain. Procedia Comput Sci 35:1155–1164
20. Wang RL (2010) SWOT analysis of the financing of SMMs. Mark Modernization 11(27):68–69
21. Song TX, Zhang CL, Huang BQ (2013) A cloud manufacturing service platform for small and medium enterprises. Comput Integr Manuf Syst 5(5):47–52
22. Wang XH, Yuan Y, Zhao ZC et al (2014) Patent literature search and its application to patent a static value calculation. Inf Stud Theory Appl 37(12):76–80
23. Kimura T, Tanaka Y (2010) Research on evaluation technique of patented invention using both technical value and economic value. Technology Management for Global Economic Growth(PICMET). 2010 Proceedings of PICMET'10. IEEE, 2010: 1-7
24. David Lichtenthal J, Eliaz S (2003) Internet integration in business marketing tactics. Ind Mark Manag 32(1):3–13
25. SONG TX, Zhang CL, Li CH et al (2013) A cloud manufacturing service platform for small and medium enterprises. Comput Integr Manuf Syst 31(6):1278–1283
26. Wei CM, Zhang CL, SONG TX, et al (2013) A cloud manufacturing service management model and its implementation. 2013 Int Conf Serv Sci (22):60-63
27. Meier H, Völker O, Funke B (2011) Industrial product-service system (IPS2) paradigm shift by mutually determined products and services. Int J Adv Manuf Technol 52(9):1175–1191
28. Chen K, Zheng WM (2009) Cloud computing: system instances and current research. J Softw 20(5):1337–1348
29. Erkoyuncu JA, Roy R, Shehab E, Cheruvu K (2011) Understanding service uncertainties in industrial product-service system cost estimation. Int J Adv Manuf Technol 52(9):1223–1238
30. Dai LL, LIU B, XIA YN et al (2008) Measuring semantic similarity between words using HowNet/ / ICCSIT'08: 2008 International Conference on Computer Science and Information Technology Washington, DC. IEEE Comput Soc 2008:601–605
31. Zhu ZY, Sun JHJ (2013) Improved vocabulary semantic similarity calculation based on How Net. J Comput Appl 33(8):2276–2279
32. Yu X, Liu JH, He M (2011) Design knowledge retrieval technology based on domain ontology for complex products. Comput Integr Manuf Syst 17(2):225–231
33. Zhang XW, Wang KQ, Tong SR (2014) Value modeling of customer needs to support value-driven design 20(10): 2340-2347
34. Keeney R (1993) Decisions with multiple objectives. Cambridge University Press, USA
35. Huang WD, Wang YY, Xie Q et al (2007) Research on management of design k now ledge based on ontology in the integrated product development. China Mech Eng 18(21):2566–2569
36. Tao F, Zhang L, Lu K, Zhao D (2012) Study on manufacturing grid resource service optimal-selection and composition framework. Enterp Inf Syst 6(2):237–264
37. Tao F, Zhang L, Nee AYC (2011) A review of the application of grid technology in manufacturing. Int J Prod Res 49(13):4119–4155
38. Deif AM (2011) A system model for green manufacturing. J Clean Prod 19(14):1553–1559

39. Eppler J (2014) Making knowledge visible through intranet knowledge maps: concept, elements cases. http://www.miter.org/news/theedge/april00/damore.html
40. Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A (2011) Cloud computing—the business perspective. Decis Support Syst 51(1):176–189
41. Fabrizio S (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):11–12, **32-33**
42. SALTON G, WONG A, YANG CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
43. Brandt SC, Morbach J, Michalis M (2008) A ontology-based approach to know ledge management in design processes. Comput Chem Eng 32(1):320–342
44. Yin S, Yin C, Liu F et al (2011) Outsourcing resources integration service mode and semantic description in cloud manufacturing environment. Comput Intergr Manuf Syst 17(3):526–532
45. Li HF, Zhang L, Jiang R (2014) Study of manufacturing cloud service matching algorithm based on OWL-S. 26th Chin Control Decis Conf (2014):4155-4160
46. Zhang BH (2011) A dissertation for the degree of M. eng research on matching algorithm in semantic web services based on group preference. Dissertation, Harbin Engineering University
47. Wang LH (2013) Machine availability monitoring and machining process planning towards Cloud manufacturing. CIRP J Manuf Sci 6(4):263–273
48. Kalpakjian S, Schmid S (2006) Manufacturing engineering and technology (5th ed.). Prentice Hall, Englewood Cliffs, p 1192
49. Li JR, Tao F, Cheng Y, Zhao LJ (2015) Big data in product lifecycle management. Int J Adv Manuf Technol 81:667–684
50. Zou XF, Tao F, Jiang PL, Gu SX, Qiao K, Xu L (2015) A new approach for data processing in supply chain network based on FPGA. Int J Adv Manuf Technol. doi: 10.1007/s00170-015-7803-x
51. Tao F, Zhang L, Lu K, Zhao D (2012) Research on manufacturing grid resource service optimal-selection and composition framework. Enterp Inf Syst 6(2):237–264
52. Tao F, Zhang L, Zhang ZH, Nee AYC (2010) A quantum multi-agent evolutionary algorithm for selection of partners in a virtual enterprise. CIRP Ann Manuf Technol 59(1):485–488
53. Tao F, Laili YJ, Zhang L, Zhang ZH, Nee AYC (2014) QMAEA: a quantum multi-agent evolutionary algorithm for multi-objective combinatorial optimization. SIMULATION: Transactions of The Society for Modeling and Simulation International 90(2):182–204
54. Tao F , Li C, Liao TW, Laili YJ (2015) BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing, IEEE Trans Serv Comput. doi: 10.1109/TSC.2015.2416928
55. Tao F , Zhao DM, Yefa Hu, Zude Zhou (2010) Correlation-aware resource service composition and optimal-selection in manufacturing grid. Eur J Oper Res 201(1):129–143