Received June 6, 2020, accepted July 12, 2020, date of publication July 16, 2020, date of current version July 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009678

Multi-Granularity Modeling and Aggregation of Design Resources in Cloud Manufacturing

SHUHUI DING[1,2], JINGLIANG HAN[1], XIAOJUN MENG[3], AND FAI MA[2]

[1]College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94709, USA
[3]School of Mechanical Engineering, Taishan University, Tai'an 271000, China

Corresponding author: Xiaojun Meng (mxj931@tsu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51974170, and in part by the China National Coal Association Foundation under Grant MTKJ2016-282.

ABSTRACT In most Cloud Manufacturing (CMfg) systems, Design Resource (DR) is encapsulated into cloud service under a fine-grained condition. However, due to the small granularity of DRs provided by cloud provider, it is difficult for the cloud services to match with design tasks if there is no initiative resource. For example, because of the lack of initiative perception capabilities, it is difficult for design software to match with design tasks directly. A method of DR multi-granularity modeling with two-stage aggregation is proposed, by which the resource granularity is increased and dynamic design capability is formed. In the proposed DR multi-granularity model, DRs are classified into three granularities: Static Physical Resource (SPR), Dynamic Capacity Resource (DCR), and Cross-functional Design Unit (CDU). Their ontology models are set up to represent the basic function, structure and component of DRs. In the two-stage aggregation of DRs, two strategies are proposed to increase the granularity of DRs. The first is DCR aggregation strategy based on auxiliary resources actively pushing, and the second is CDU aggregation strategy based on meta task and meta capability matching. Using the operation parameters of DRs and the associated evaluation matrix, a method of DCR and CDU evaluation is proposed to optimize the searched DRs. With the help of the preceding multi-granularity DR modeling and the two-stage access strategy, DR granularity is enlarged and initiative design capability is formed, which solves the problem of DRs matching with design tasks because of small resource granularity.

INDEX TERMS Cloud manufacturing, multi-granularity design resource, ontology modeling, resource aggregation, resource evaluation.

I. INTRODUCTION

The Design Resources (DRs) in Cloud Manufacturing (CMfg) system [1], [2] refer to all the involved elements within the whole product life cycle. The perception and access of DRs [3], [4] is the foundation and premise of its virtualization and servitization [5]. Because of the difference among DRs at the attributes of existing form, transmission medium and usage mode, DRs are typically heterogeneous resources. Meanwhile, resource status will present dynamic evolution characteristics along with the carrying out of design process. The characteristics of heterogeneous and dynamic of DRs determine its complexity of modeling and accessing. Except some hardwares which are similar with Manufacturing Resources (MRs), Most of the DRs are wisdom resources, such as software, human resource, intelligent agent and so on. There are prominent difference in the way of perception and access between DRs and MRs. Most DRs are accessed with fixed I/O interface or human-computer interaction, except some hardwares such as modeling device and testing equipment which can be accessed by the methods of Internet of Things (IoT) [6], Cyber Physical System (CPS) [7], etc.

For the cloud provider [8] provided atomic DRs, if they are not aggregated with initiative resources, it is hard to access to the design process submitted by cloud requester. Only through reasonable resource aggregation [9], forming dynamic design capability or design units with specific function, can DRs be easily matched with specific design tasks. Therefore, all kinds of atomic level DRs, provided by cloud

The associate editor coordinating the review of this manuscript and approving it for publication was Ayaz Ahmad.

provider, need to be reasonably aggregated and form combinations with certain design capabilities before servitization encapsulation [10] and utilization. Meanwhile, reasonable DRs aggregation will help to reduce the later overhead of resource management and service composition.

The DRs in CMfg system are multiple granularity in terms of existence and usage [11]. They can be static hardware, software, knowledge, field, material and other resources, which are the smallest unit that cannot be decomposed in the process of use. They can also be dynamic capability resources formed with the assistance of static resources such as human resource, hardware, knowledge etc., which are aggregated centered on human resource. Also, they can be Cross-functional Design Unit having the capabilities of multidisciplinary design, which are aggregated with multiple capability resources. Because of the multi-granularity of DRs in terms of existence and usage, multi-granular resource modeling method is required, which will meet the requirement of resources I/O, virtualization, utilization, evaluation, etc., [12].

Due to the lack of flexibility in the granularity division of DRs, the massive design resources in CMfg system will not meet the requirements of multi-granularity reuse in the design process. However, most of the current research on CMfg system centers on the combination of virtualized service-oriented resources [13], [14], which focuses on the combination optimization of service resources [15], [16], but less on the aggregation of physical resources. Therefore, it is necessary to establish an appropriate granularity of hierarchy of DRs to enhance its design flexibility and agility in CMfg, to build a kind of aggregation and evaluation method of DRs to compose and optimize aggregated resources.

This paper proposes to address the problems of multiple-granularity DRs modeling, aggregation and its evaluation. First, a multi-granularity resource model is presented to express the three granularities of DRs: Static Physical Resource (SPR), Dynamic Capacity Resource (DCR) and Cross-functional Design Unit (CDU), and then their ontology models are defined. Secondly, in order to enlarge resource granularity, two resource aggregation strategies are presented, which accomplish the construction of DCR and CDU respectively. Finally, comprehensive evaluation matrixes are built, which is used to optimize the alternative DCR and CDU.

The remainder of the paper is organized as follows. Section II reviews the literature relating to resource classification, resource granularity, resource modeling and resource accessing. In Section III, multi-granularity resource model is defined. In Section IV, the ontology models of SPR, DCR and CDU are created separately. The aggregation strategies of DCR and CDU are proposed in Section V. In Section VI, a kind of aggregation resource evaluation method is designed and comprehensive evaluation matrixes are built to optimize DCR and CDU. The proposed methods are validated through a case study in Section VII, followed by conclusion in Section VIII.

II. LITERATURE REVIEW

Scholars have carried out many research studies on resource classification. Zhang *et al.* [17] classified resources to MRs and manufacturing capabilities, so that different virtualization methods can be used to diverse resources. Zhang *et al.* [18] classified MRs to machines and machining unit according to the different size of MRs granularity, and created the capability model of machine and the information model of machining unit. Zheng *et al.* [19] created MRs ontology tree and classify MRs to finance, service, equipment, manpower, software and logistics. Zhu [20] classified MRs to manpower, manufacturing equipment, software, service, supplies, computing, manufacturing knowledge and other resources. According to the existing form of DRs and their relationship, Kong [21] classified DRs to physical DRs and invisible design capabilities. Physical DRs contain tools, intelligent and knowledge and invisible design capabilities include all the required capabilities at the product design stage, such as requirement analysis, scheme selection, concept design, structure design and entity design. In the preceding MRs classification method, resources and capabilities are at one level, and the relationship between them have not been expressed, which is against the later resource modeling.

In the aspect of the relationship among MRs, manufacturing capabilities and manufacturing services, significant research had been carried out. Wang and Xu [22] proposed that it is the function of equipment, not the physical device itself, that should be serious considered. Guo [23] proposed a logical model of MRs, manufacturing capabilities and manufacturing services, which considered that manufacturing capabilities were formed by set of MRs, and manufacturing services were encapsulated from manufacturing capabilities. This model represented the hierarchical relationship among MRs, manufacturing capabilities and manufacturing services. In the preceding discussion about MRs, manufacturing capabilities were separated from MRs and not all kinds of MRs were expressed completely.

In the matter of resource granularity classification, research had been done. In [24], user tasks were classified to single resource service task and multiple resources service task. Different granularity of manufacturing cloud service was constructed, which generated different granularity of manufacturing capabilities. Zhu [20] classified MRs to multiple granularities. From the view of the task scale and manufacturing target, he proposed that resource model must support multiple granularities of resource composition. Single physical MRs exist independently and its granularity is small, whereas manufacturing target is a complex manufacturing task that is achieved through a collaboration of multiple single MRs.

Some scholars also researched the multi-granularity characteristics about service evaluation and design task. Deng *et al.* [25] built a kind of multi-granularity service evaluation model for service node. When evaluating service, the coarse-granularity global evaluation achieved through the creation of individual experience based multi-granularity trust model. And also, aiming at visitors' different interests and preferences, single element evaluation can be checked using one declaration of multi-tuple, which improved the granularity of trust evaluation. Guo [23] divided manufacturing services into two granularities, which are process level and part level. According to the different requirements of process level manufacturing and part level manufacturing, multi-granularity manufacturing task and manufacturing service model was proposed to construct optimized manufacturing service in process level or part level.

On the aspect of DR model, ontology modeling [26] is popular currently. Aiming at specific domain, ontology modeling achieves the capture of domain knowledge though making common terminologies and specifications. At the same time, formal definition of representation is proposed to realize the knowledge sharing across industries. Most scholars use n-tuple to express ontology. For example, Gao *et al.* [27] use two-tuple, Li *et al.* [28] use five-tuple and Kang *et al.* [29] use eight-tuple. Finite objects are used to define the elementary component of ontology in the definition of n-tuple. For example, Wei [30] defined a five-tuple ontology, using the formula: $O=(C, P, R, A, I)$, the objects of which stands for class, attribute, relationship, axiom and instance separately.

In most papers, fine-grained resources were encapsulated for servitization directly, and then the generated services were utilized. Zhu *et al.* [31] proposed a manufacturing service composition scheme named as Multi-Module Subtasks Collaborative Execution for Cloud Manufacturing Service Composition (MMSCE-CMSC), which was used to select services from service pool to generate a composite service with the given objective optimization. Li *et al.* [32] proposed a two-level multi-task scheduling model, which was used to find a better resource schedule. In this paper, distributed resources were encapsulated into cloud services directly. Aiming at the problems of multi objective CMfg service composition optimization during the collocation between services and tasks, a novel service composition optimization approach was proposed by Li *et al.* [33], which was called improved genetic algorithm based on entropy (LGABE). But the manufacturing resources were provided in the form of cloud services directly and fine-grained simple cloud services were combined into coarse-grained complex cloud services through service composition, to meet the requirement of task. In the preceding research, all the resources were directly virtualized to cloud services at the condition of minimum granularity, and then form specific design capability or manufacturing capability to meet the resource requester's requirement. In these paradigms, resources and services maintained a strict one-to-one correspondence relationship. But because of the small granularity of accessed atomic MRs, the generated fine-granularity cloud services were difficult to find and match with task in cloud service platform, which could not meet the requester's requirement for resource granularity in general.

Some research about resource aggregation had been done, which composited the fine-granularity resource and increased the resource searching efficiency. Cao *et al.* [34] studied many kinds of aggregation methods for different DRs. According to different accessed DR, two aggregating methods, modularization aggregation and virtual design unit aggregation, were put forward to organize the knowledge DR and organize the substantive DR respectively. Guo and Ma [35] proposed a kind of resource aggregation mechanism driven by functional requirement to achieve the virtual service resource build of single resource. Gao *et al.* [36] created a CMfg service resource modeling method based on multi-domain, and put forward a virtual service resource composition method, which achieved the virtual service resource building of single resource. Zhu *et al.* [37] proposed that resource in Cloud Terminal consisted of Single Resource (SR) and Complex Resource (CR) and CRs were formed by composition of SR based on functionality. Liu *et al.* [38] established a multi-granularity resource virtualization model, in which resource aggregation functions are constructed from attribute, activity and process levels. Multi-granularity resource virtualization strategies were proposed, in which attribute, activity and process oriented resource were mapped into virtualized resources based on interrelated resource aggregation functions. For the question of Manufacturing Resource Combinatorial Optimization (MRCO), Wang *et al.* [39] proposed a manufacturing resource selection strategy based on an improved Distributed Genetic Algorithm (DGA). All the preceding researches about resource aggregate are carried out for the purpose of resource application and there are seldom studies about initiative aggregation.

III. MULTI-GRANULARITY DESIGN RESOURCE MODEL

A multi-granularity DRs model is proposed to classify and define MRs from different levels, which divides MRs into three granularities. Shown in Fig.1, the three granularities are SPR, DCR and CDU.

In the multi-granularity DR model, SPR layer is the foundation, and DCR layer is the aggregation and embodiment of SPRs. SPR cannot be virtualized as cloud service directly, and it can be utilized only after aggregated to DCR or CDU. CDU is a combination of multiple DCRs and SPRs, which is used to complete specific design tasks and is dynamically aggregated according to design tasks.

(1) SPR. It refers to the relatively stable resource in design process, which exists on the basis of entities or software normally. Because of its static characteristic, SPR is easy to be mined, sorted, indexed and applied, which is indecomposable and the minimum resource element in the design process. SPRs mainly include intellectual resource, tool resource, knowledge resource, field resource, material resource and logistics resource. Intellectual resource includes all kinds of manpower resource and intelligent application used in the design process, such as administrative staff, domain engineer, product design engineer, test engineer, marketing engineer and expert system which can simulate human expert's behavior to solve domain problem. From the view of resource modeling, manpower resource has static attributes,

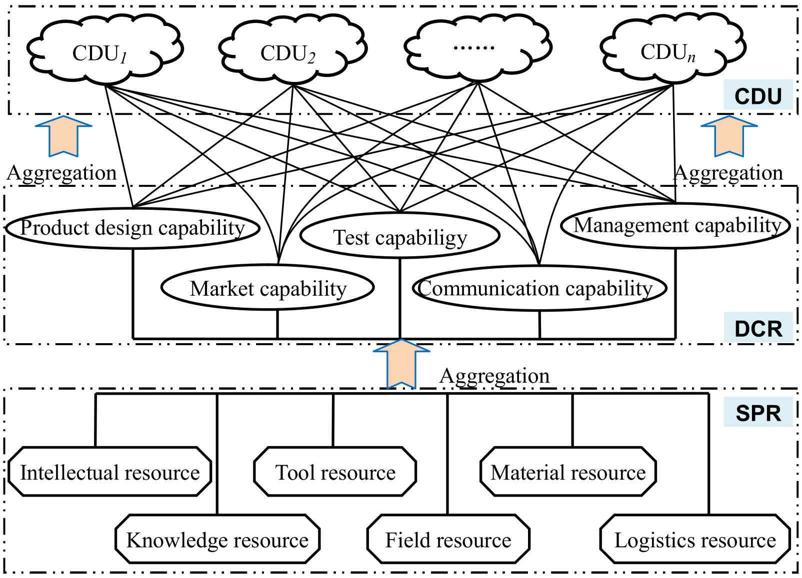


FIGURE 1. **Multi-granularity DR model.**

such as age, gender, job and specialization, and it belongs to static resource.

(2) DCR. It refers to the resource combination centered on intellectual resources for the purpose of achieving a certain expected objective. This combination is accomplished by selecting and matching SPRs. DCR is the combination of specific DRs with complete logical structure, aiming at specific design tasks, centering on human resources or intelligent computer program, assisted by equipments, software, knowledge resources and so on, which represents the ability to complete a specific design task, such as product design capability, analysis capability, test capability, management capability, etc.

DCR aggregation aims at the capability of requirements in the design process. For example, Product design capability refers to the ability to complete product design work using intellectual resources as the main resources, with the help of knowledge resources, software resources and etc. According to the design process, product design capability is divided into scheme demonstration capability, conceptual design capability and structural design capability, which are respectively used to complete the early scheme demonstration, product appearance design, and later detailed structure and tolerance design of the product. When building DCR, resources meeting the requirement are selected, in which intellectual resources with the required capabilities are as the center.

(3) CDU. With the aim of accomplishing product design or a certain stage task of it, CDU is a kind of resource aggregation with multiple design capabilities needed in the product design process, which is composed of SPRs and DCRs. CDU is put forward based on the concept of Integrated Product Team (IPT) [40], Virtual Manufacturing Cell (VMC) [41], Virtual Manufacturing Cell in CMfg (VMC in CMfg) [42] and Virtual Design Cell (VDC) [34], which is a design service unit that optimizes and aggregates the available SPRs and DCRs in CMfg system. CDU is used to complete the requirements of resource requester for specific product design.

In CMfg environment, CDU is actively aggregated by resource providers according to typical design tasks and design processes in the industry, which is a logical reconstruction of the design resources involved. In essence, CDU is a combination of resources and capabilities for the design tasks proposed by resource requester, as well as an information description of resources and their relationships.

CDU is different from VMC. VMC is a logical transient reconstruction of physical resources, which is a kind of resource combination based on some specific production tasks, and is generated with a specific production task and disbanded with the end of the production task [43]. While CDU is a multi-resource and multi-functional heterogeneous group established under the background of current industry segmentation, aiming at design tasks in specific fields. CDU is a relatively stable combination, and only when its evaluation values fail to meet the specified threshold, CDU needs to be decomposed and recombined. VMC in CMfg is a kind of minimum manufacturing resource aggregate that can run in CMfg environment and complete a certain process level manufacturing task of a product [42], which consists of four parts: basic information, resource structure information, manufacturing activity weight information and aggregate manufacturing capacity information. Compared with CDU, it lacks supports of knowledge resources, such as domain knowledge, design cases, national standards, etc.

CDU is also different from IPT. Only the optimal combination of designers is considered in IPT, and the organization and aggregation of other DRs are not involved [44]. Compared with the mode of IPT, VDU is a collection of hardware devices such as designers and computers that can independently complete a design activity in the distributed design environment [34], but it also lacks the support of knowledge resources compared with CDU.

IV. DESIGN RESOURCE ONTOLOGY MODELING

Due to the powerful resource expression ability and the advantage in the aspect of semantic matching, ontology modeling technology [45] has become the mainstream method for resource modeling. According to the requirement of multi-granularity DR modeling, a kind of DR universal ontology modeling method is created, with which all kinds of resource ontology model are generated.

A. DR GENERAL ONTOLOGY MODEL

Through extracting general rules of things, ontology expresses object with "class" and defines the relationship among classes by "property". Therefore, as a modeling method of DR, ontology needs to express the following two aspects.

(1) DR classification, resource elements and relationships with other DRs. For example, DRs have three levels: SPR, DCR and CDU. SPR consists of intellectual resource, knowledge resource, tool resource and so on. Intellectual resource can be divided into administrative staff, design engineer, domain engineer, and technician. Tool resource can be divided into computing resource, storage resource, software resource, etc. Design engineer has a relationship of "use" with software resource. Tool resource has a relationship of "is…subclass" with software resource.

(2) DR parameters and its relationship. For example, the functional parameters of design resource include the capability of modeling, design, simulation, sheet metal design, mold design and NC manufacturing. Design software has a relationship of "has…capability" with modeling capability.

According to the preceding definition of ontology, all kinds of resources and their parameters can be expressed by "class", and the relationship between resources can be expressed by "relationship". Therefore, to construct DR ontology, there are three parts needed to express as follows.

(1) Ontology. It is the object to be normalized represented, which can be used to express an object or its parameter. For example, "ontology" can be used to express all the DRs, a certain granularity level of DRs, one kind of DRs, one specific DR or some parameters of DR.

(2) Class. It is the collection of individuals of an object, which can be used to express everything about the object. For example, use "class" to express the constituent individuals of an object, or the involved function, strategy, behavior, etc.

(3) Property. It is the relationship among classes, which is used to express the relationship between two classes, class and words, or class and numerical value.

1) TWO-TUPLE FORMALIZED DEFINITION OF GENERAL ONTOLOGY MODEL

Base on the preceding requirement analysis, a kind of formalized definition is proposed to be used as the general ontology model of DR, which is described as a two-tuple. The definition is as follows.

$$O = (C, P^C) \tag{1}$$

where O is the ontology model, which can express the whole resources, resource individuals or parameters; C represents the classes of ontology, which is used to express DR or its parameters; P^C represents the property of classes, which is used to express the relationship between two classes.

In general, the definition of ontology should express class, property and the relationship among classes [46]. In (1), P^C represents both property and the relationship among classes. The definition of P^C includes domain of definition and domain of value. Domain of definition is one class, and the domain of value can be one class or a specific value. If it is one class, the instance of relationship can be expressed as $r_i(c_a, c_b)$, which represent the binary relationship between class c_a and c_b. If the domain of value is a specific value, then the relationship degenerates into data property between class and value.

When modeling one specific DR, firstly define one ontology model as template of DRs with common attributes, and then classes and properties of the template are described to define the attributes and relationship between DR classes. Generate an instance from the ontology template, and then initialize its classes and properties by the information offered by cloud provider. The instance with the initialized classes and properties is the ontology model of the specific DR.

2) REPRESENTATION OF ONTOLOGY MODEL COMPONENT OBJECTS

In (1), a two-tuple ontology model is used to express two component objects, which are C and P^C. For different kinds of ontology resource, it is necessary to create specific class set and property set, which are used to express the information of different DR, as follows.

$$\begin{cases} C = \{C_1, C_{2,}, C_3 \ldots, C_i\} \quad (i = n) \\ P^C = \{P^C(C_1), \ldots, P^C(C_i), P^C(C_1, C_2), \ldots, \\ \qquad\qquad P^C(C_{i-1}, C_i)\} \quad (i = n) \end{cases} \tag{2}$$

where C represents the class set and P^C represents the property set; i is the quantity of classes in the ontology; c_i represents the No.i class; $P^C(C_i)$ represents the property of No.i class; $P^C(C_i, C_j)$ represents the relationship between the No.i class and the No.j class.

Any DR ontology model can be created using (1) and (2), and the procedure is as follows: (1) create the ontology template of this kind of DR, (2) instantiate the template with resource information.

For example, one service provider wants to create the model of Romax software. Its classes include ID, name, version, cost, owner, address, function, available time, price, QoS and service records. The ontology modeling process is as follows. (1) Create design software ontology model as the template

According to the preceding method, create design software ontology template as follows.

$$\begin{cases} O_{mf} = (C_{mf}, P_{mf}^{C}) \\ C_{mf} = \{ID, Nam, Ver, Cos, Onr, Add, Fun, VaT, \\ \qquad Pri, QoS, SerRec\} \\ P_{mf}^{C} = \{P_{mf}^{C}(ID) = hasID, P_{ms}^{C}(Nam) \\ \qquad = hasNam \ldots, P_{ms}^{C}(SerRec) = hasSerRec\} \end{cases} \tag{3}$$

The above template can be expressed by design software ontology structure diagram, shown as Fig.2. In the diagram, *ModelSofw* stands for design software ontology model. It has 11 classes, which are *ID*, *Name*, *Version*, *Cost*, *Owner*, *Address*, *Function*, *ValidTime*, *Price*, *QoS* and *SerRec*. The first 9 classes are expressed by string. *QoS* represents the customer's evaluation, which is expressed with value. *SerRec* represents the records of resource, which include four components: the number of service times, finished service times, success service times and service duration.

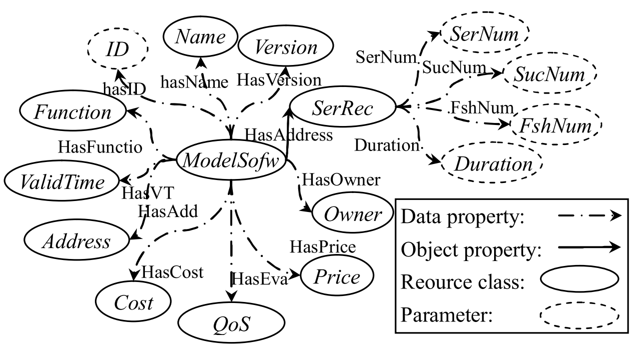


FIGURE 2. Ontology structure of DR.

(2) Instantiate the template into the specific software

Use the information offered by resource provider to instantiate the above template, and get the Romax model as follows.

$$\begin{cases} O_{mf} = (C_{mf}, P_{mf}^{C}) \\ C_{mf} = \{0952, Romax, 17, 150k, XX\ Inc., 85\ Marin\ St, \\ \qquad analysis, 20200301\ later, 200, (88, 92.0\%, 95.7), \\ \qquad (3, 3, 2, [2, 3, 3.2])\} \\ P_{mf}^{C} = \{P_{mf}^{C}(ID) = 0952, \ldots, P_{mf}^{C}(SerRec) \\ \qquad = (3, 3, 2, [2, 3, 3.2])\} \end{cases} \tag{4}$$

Formula (4) represents the software of Romax provided by XX Inc., in which the software's properties represents by the set of C_{mf}.

B. SPR ONTOLOGY MODELING

1) ONTOLOGY STRUCTURE OF SPR

Based on the requirement of resource modeling, SPRs are divided into eight kinds, which are human resource, artificial intelligent resource, software tool resource, hardware tool resource, knowledge resource, material resource, field resource and logistics resource. The ontology model structure of SPRs is shown in Fig.3.

Fig.3 shows the class composition of each ontology model, as well as the relationship between various resources and the inheritance relationship between parent class and subclass. Among the structure, "IntRes" represents intellectual resource, which includes human and artificial intelligent resources. "TolRes" represents tool resource, which includes software and hardware resource. "SerInf" respresents service information of each resource, which includes valid time, hour price, QoS, the number of service times, finished service times, success service times and service duration.

2) INSTANCE OF SPR ONTOLOGY MODEL

Taking manpower resource for example construct DR template. And taking one design engineer for instance explain the resource instantiation process.

According to formula (1), the two-tuple ontology model of human resource is constructed as follows.

$$O_{HumRes} = (C_{HumRes}, P_{HumRes}^{C}) \tag{5}$$

where O_{HumRes} represents human resource ontology model, and it is the template of all human resource instance; C_{HumRes} represents the class set of human resource; P_{HumRes}^{C} represents the property of human resource.

There are many kinds of human resources involved in the process of product design. It is necessary to refine the human objects layer by layer, so as to reflect the personal information of designers gradually.

At first, human resource classes and properties are listed as follows.

$$C_{HumRes} = \{ID, BasInf, Rol, Spy, ValTim, Pri, QoS, SerNr, \\ FshNr, SucNr, Duration\} \tag{6}$$

$$P_{HumRes}^{C} = \{P_{HumRes}^{C}(ID) = hasID, P_{HumRes}^{C}(BasInf) \\ = hasInf \ldots, P_{HumRes}^{C}(Duration) \\ = hasDuration\} \tag{7}$$

In (6), C_{HumRes} represents the class set of human resource template, and its values in turn represents ID of human resource, basic information, work position, specialty, available time, labor cost, QoS, service times, finished service times, success times and service duration time. In (7), P_{HumRes}^{C} represents the properties of classes. Because most of the classes in human resources have no relationship with each other, the property set of class is relatively simple, and the objects in the set are self attributes with their own information. Later, for the description of ontology resource, if there is no special case, the property set will not be listed.

In (6), *ID* represents the unique resource identifier in the system, expressed by a string. *BasInf* represents the basic information of human resources, expressed by the set shown in (8). *Rol* represents the position of individuals, expressed by the set shown in (9). *Spy* represents specialty, which refers

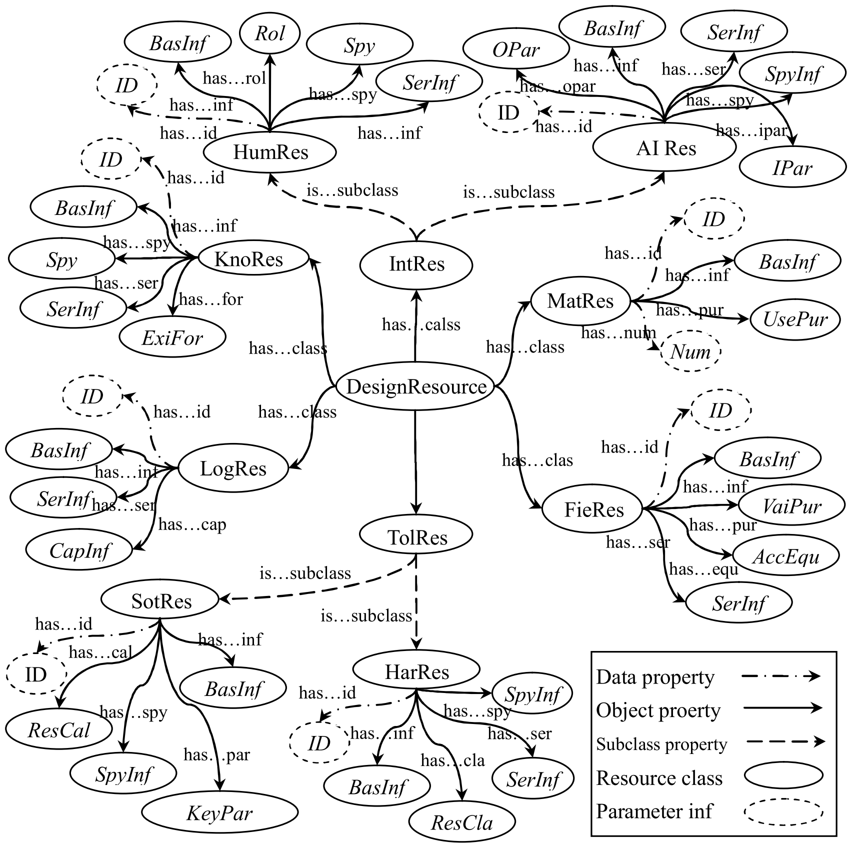


FIGURE 3. Ontology structure of design resource.

to the work proficiency of designers, expressed by the set shown in (10). *ValTim* represents available time, expressed by a period of time. *Pri* represents labor cost, expressed in numbers. *QoS* represents customer's evaluation, expressed with value. *SerNr*, *FshNr* and *SucNr* represent the number of service times, finished service times, success service times respectively, all of which are represented by numbers. *Duration* represents service duration time, expressed in arrays.

$$C_{BasInf} = \{Nam, Sex, BirDay, Col, EduBg, PstTil, Add, Tel, Eml\} \tag{8}$$

$$C_{Rol} = \{DsnEgr, DomEgr, TstEgr, MktEgr, Mgr, Tcn\} \tag{9}$$

$$C_{Spy} = \{Mdl, Dom, Mkt, Tst, Mng, Tec\} \tag{10}$$

In (8), the elements of *BasInf* set represent the designer's name, gender, date of birth, school, education background, title, address, telephone and email respectively. In (9), the elements of *Rol* set represent design engineer, domain engineer, test engineer, market engineer, organization manager and technician respectively. In (10), the elements of *Spy* set represent modeling, domain knowledge, market, testing, management and technical capability respectively. Among them, modeling, market and technical capability can be divided into many kinds, as shown in (11).

$$\begin{cases} C_{Mdl} = \{IndDsg, SodMdl, SufMdl, AniDsg, SimAly, \\ \qquad\quad ShtDsg, MolDsg\} \\ C_{Mkt} = \{Neg, PreMkt, AftSalSer\} \\ C_{Tec} = \{Mch, HotWrk, HetTrt, AsmDeg, BenWrk, Wed\} \end{cases} \tag{11}$$

In (11), the elements of *Mdl* set represent industrial design, solid modeling, surface modeling, animation design, simulation analysis, sheet metal design and die design respectively. The elements of *Mkt* set represent business negotiation, pre-sales consultation and after-sales service respectively. The elements of *Tec* set represent machining, heat processing, heat treatment, assembly and debugging, maintenance and welding respectively.

C. DCR ONTOLOGY MODELING

1) DEFINITIONS AND RULES

Definition 1 (Meta Resource): Only having static characteristics, it is the minimum resource that can be invoked by individuals or intelligent agents with initiative. In this paper, the SPRs that have not been aggregated are meta resources.

Definition 2 (Meta Capability): It is the smallest unit with the ability to solve specific problems in a certain field, which takes human resources with initiative or intelligent software with self driving ability as the main body, and other related resources as the auxiliary. Meta capability is the core of DCR aggregation.

Definition 3 (Master Resource): It refers to the human resource or intelligent software, which can initiate design activities when receiving design task. It is a kind of initiative resource.

Basing on their initiative, human resources can carry out design activities according to the received task and parameters with the help of other resources. Also, intelligent software, including expert system and other intelligent computer program having fixed I/O interface, will automatically complete specific design tasks when they receive the start command and input parameters.

For example, in the design process of high-speed and heavy-duty gear transmission mechanism, in order to keep a reasonable contact area of gear surface for the deformed gear under load, it is necessary to make a proper surface modification to the gear, which can ensure a uniform pressure distribution on the gear. As an indecomposable minimum element in the process of gear design, the ability of gear surface modification is called one meta capability. The engineer who completed the modification is the master resource. All kinds of software tools and experimental equipments used in the process of gear modification design are meta resources.

Definition 4 (Auxiliary Resource): It refer to the resources that can not initiate design activities, which are used to support or assist design activities and only can accept aggregation passively.

According to the role in design process, SPRs are divided into master resources and auxiliary resources. Intellectual resources, including human resources and intelligent software, belong to master resources. Other SPRs, such as tool resources, knowledge resources, site resources, material resources and logistics resources, are all auxiliary resources.

2) DCR ONTOLOGY MODELING

DCR are aggregated with meta capability as the center. The establishment of DCR is generally initiated by master resources and then combined with auxiliary resources. The expression of DCR requires elements as follows.

(1) Meta capability. It is the center of DCR, by which the main function of DCR is accomplished. A DCR has only one meta capability. If there is other capability, another DCR can be built with it. The later searching and usage of DCR is also based on meta capabilities.

(2) Master resource. The meta capability of DCR comes from individual or intelligent agent with initiative. Therefore, master resource is the sole source of DCR meta capability. When establishing DCR, the first task after determining the meta capability is to determine the designer or the intelligent software with this capability. For example, gear modification DCR takes surface modification as its meta capability. So when constructing gear modification DCR, it is crucial to search for designers with this capability as the master resource.

(3) Auxiliary resource. Master resource is the key to construct DCR, and the implementation of meta capability needs the support of other auxiliary resources, which mainly include design, modeling and analysis software, computer, experimental apparatus, etc. For example, besides the initiative of engineer, the implementation of gear modification DCR meta capability also needs the support of modeling and analysis software, as well as the corresponding computer hardware system.

(4) I/O and status information. The results of DCR design activities are shown in two aspects: data change and state transformation. Data change refers to the transition from input data to output data caused by design activities, which are expressed by I/O data. For example, because of the design activities of gear design DCR, the input parameters are transformed into gear parameters as output result. State transformation refers to the change of task state caused by design activities. For example, because of the design activities of DCR, the task state is transformed from ‘‘start’’ to ‘‘finish’’.

(5) Service information. It refers to the service attributes of DCR ontology model, as well as customer evaluation and history information after service. Service attributes mainly refer to the available service time and service price. The former is the important basis for searching available resources and the latter is the main reference for optimization. Customer evaluation refers to the subjective evaluation given by users as customers after the design services, expressed by QoS. The history information records the number of service times, finished service times, success service time of the DCR since its establishment.

Customer evaluation and history information include two parts. The first is called member information, which is accumulated from the customer evaluation and historical records of the master resource and auxiliary resources before the establishment of DCR. The second is the attributes information of DCR itself, which is the historical record of the design activities after the establishment of DCR and the customer's evaluation of DCR. The member information is calculated according to each component resource, which is an important basis for SPR selection when building DCR. When resource requester searches for DCR, the member information and DCR attributes information are used as the basis of DCR optimization, and the weight of the two parts is determined by weight coefficient.

(6) Other information. It includes ID, name, and the time recorded by the system.

Based on the two-tuple ontology model shown in (1), define the DCR ontology model as follows.

$$O_{DCR} = (C_{DCR}, P_{DCR}^{C}) \tag{12}$$

where O_{DCR} represents DCR ontology model, which is the universal template of DRC instance; C_{DCR} represents the class set of DCR, which expresses the main resource features; P_{DCR}^{C} represents the attribute set of DCR.

According to the expression of DCR, the class set of DCR is shown as follows.

$$C_{DCR} = \{ID, MetaCap, MSPR, L_{ASPR}, L_{Ipt}, L_{Opt}, L_{\Pr e}, L_{Rst}, SerNr, FshNr, SucNr, [LoT], ValTim, Pri, QoS\} \tag{13}$$

The class set shown in (13) represents the main components of DCR, where ID, MetaCap and MSPR represent resource number, meta capability and master resource respectively. LASPR L_{ASPR} represents its auxiliary resources, listed as an array. L_{Ipt}, L_{Opt}, L_{Pre}, and L_{Rst} represent input parameter list, output parameter list, precondition list and result list respectively. *SerNr*, *FshNr*, *SucNr* and [*LoT*] are the service information of DCR, which respectively represent the number of service times, finished service times, success service times and duration time of service. *ValTim* and *Pri* represent available time and labor cost respectively. *QoS* represents the customer's evaluation information set.

D. CDU ONTOLOGY MODELING

With the meta capabilities of multiple DCRs as the center and the relevant SPR as the auxiliary, larger granularity of DR combination can be built, which is called CDU. It is a multiple design capabilities aggregation aiming at the typical product design in a specific field. CDU is a multi-disciplinary and multi-resource product design team, which takes multiple DCRs as the core to build product design process, supplemented by relevant knowledge, computer, and other resources.

For example, in order to complete the design of a high-speed and heavy-duty transmission system, it is necessary to organize human resources or expert systems with various meta capabilities, such as gear design, gear modification, dynamic analysis and rotor dynamic balance analysis. CDU is constructed to complete the design task, by searching DCRs organized around these meta capabilities, or searching the relevant member resources such as domain specialist, high-performance computer, dynamic gearbox testbed, gear design domain knowledge package to build DCR firstly.

According to the preceding analysis of CDU construction, the required information to express CDU is as follows.

(1) Meta capability set. It provides basis for the construction of CDU. Based on the design task, meta capability set is established, which is achieved by searching the related DCRs.

(2) DCR set. It is the main resources for the construction of product design process.

(3) Auxiliary resource set. It is used to assist DCR to complete design task.

(4) I/O and status information. The invoking of CDU can produce data change and state transformation. Data change refers to the change between the input of the first DCR and the output of the last DCR. State transformation refers to the transition from the preconditions before the invoking of CDU to the result.

(5) Service information. It refers to CDU's service attributes, customer evaluation and history information. Service attributes mainly refer to available time and service price. Customer evaluation is the subjective evaluation given by resource requesters who invoke the resource, which is expressed by QoS. History information records the number of service times provided by CDU since its establishment, finished service times, success service times and the service duration of each task.

CDU customer evaluation and history information include two parts: member information and operation information. Member information records the customer evaluation and history of all member resources (including DCRs and all SPRs). Operation information is the historical record and customer evaluation information generated from design activities during CDU operation. Member information is calculated according to each component resource, which is an important basis for resource selection when establishing CDU. Operation information is an important optimization basis for resource requesters to search and optimize service resources. The importance of the two parts is determined by weight coefficient.

(6) Other information. It includes ID, resource named and establishment time recorded by the system.

Based on the above description of CDU, the general ontology model of CDU is as follows

$$O_{CDU} = (C_{CDU}, P_{CDU}^{C}) \tag{14}$$

where O_{CDU} represents CDU ontology model, which is the template of CDU instance; C_{CDU} represents class set of CDU template, which express the main attributes of resource; P_{CDU}^{C} represents the property set of CDU template.

According to the preceding expression of CDU, its class set is shown as follows.

$$\begin{cases} C_{CDU} = \{ID, S_{MetaCap}, S_{DCR}, S_{Aux}, L_{Ipt}, L_{Opt}, L_{Pre}, L_{Rst}, L_{SerInf}\} \\ L_{SerInf} = \{SerNr, FshNr, SucNr, [LoT], ValTim, Pri, QoS\} \end{cases} \tag{15}$$

The class set shown in formula (15) represents the main components of CDU. Among them, *ID*, $S_{MetaCap}$, S_{DCR}, S_{Aux}, L_{Ipt}, L_{Opt}, L_{Pre} and L_{Rst} represent resource number, Meta Capacity set, DCR set, auxiliary resource set, input parameter list, output parameter list, precondition list and result list respectively. L_{SerInf} represents service information set, where *SerNr*, *FshNr* and *SucNr* represent the number of service times, finished service times and success service times respectively. [*LoT*] is a time list, recording the service duration of each design task. *ValTim*, Pri and *QoS* represent available time list, labor cost and customer's evaluation respectively.

V. RESOURCE AGGREGATION STRATEGY AND ITS ALGORITHM

The multi-granularity model of DRs expresses the logical relationship among all kinds of DRs. Task units of different granularity need to be accomplished by the aggregated resources which are composed of many types of capability resources and static resources. Therefore, the aggregation of multi-granularity DRs is the key to the matching of multi-granularity task units and service resources.

A. RULES AND DEFINITIONS

Rule 1: The original resources provided by resource provider are fine-grained atomic resources, which only have static characteristics.

Both DCR and CDU are multiple resources combinations formed in the process of aggregation. Only SPR is directly provided by resource provider.

Rule 2: Meta resources need to be aggregated before they are invoked by resource requester.

Definition 5 (Resource Aggregation): It refers to the aggregation of fine-grained resources, which is used to increase the granularity of resources and reduce the complexity of resource invoking during resource utilization. Resource aggregation is conducive to efficient resource management.

Definition 6 (Capability Aggregation): The resource aggregation for the purpose of constructing DCR is called capability aggregation, which aims to form the smallest design unit with meta capability. Capability aggregation takes intellectual resource as master resources, focus on meta ability, and complete aggregation with the help of auxiliary resources.

Definition 7 (Meta Task): Meta task is the smallest task unit that cannot be decomposed in the design process, which is obtained by decomposing design task.

Definition 8 (Unit Aggregation): The resource aggregation for the purpose of constructing CDU is called unit aggregation, which aims to accomplish a meta tasks list. During unit aggregation, CDU is constructed with multiple DCRs as the core and auxiliary resources as assistance.

B. RESOURCE AGGREGATION STRATEGY

Through virtual description and servitization encapsulation of DRs and design capabilities, cloud provider provides resource services for cloud requester. As part of resource access, resource description is the important premise for resource servitization. Larger granularity of resources are good for reflecting design capability and easy to use. Resource aggregation is a way to increase resource granularity.

When providing resources, aggregation is essential to form design capability. According to the preceding multi-granularity DR model, there are two kinds of resource aggregation. The first is to create DCR with the aim at meta capability. The second is to create CDU aiming at typical design tasks. The aggregated large-scale DR can be used by resource requester.

Only after aggregation, forming a design combination with a certain granularity, can resources have the ability of initiative product design. This capability is generated by the aggregation of initiative designers, or intelligent software, and corresponding auxiliary resources. The aggregation process of design resources is shown in Fig.4.

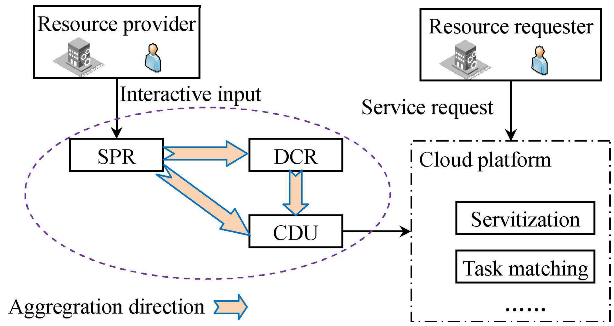


FIGURE 4. The aggregation process of DRs.

C. AUXILIARY RESOURCE ACTIVELY PUSHING BASED DCR AGGREGATION STRATEGY

In (13), the class set of ontology model defines the meta capability of DCR, the accomplishment of which requires a number of auxiliary resources as assistance. First, based on the semantic information of meta capability extracted from master resource, retrieve and match the class of "spy" in auxiliary SPR ontology model, and find the list of alternative auxiliary resources, which meet the functional requirements of meta capability. And then, based on the actively pushing of alternative auxiliary resource, establish the auxiliary resource list of DCR. In this way, DCR is established with meta capability, master resource and auxiliary resources list as components.

According to the preceding aggregation strategy, the composition of DCR is expressed as follows.

$$DCR = MetaCap + MastSpr + \sum_{i=1}^{n} AuxiSpr_i \tag{16}$$

where *MetaCap* and *MastSpr* represents meta capability and master resource respectively; $AuxiSpr_i$ represents the No.i auxiliary SPR.

According to the preceding aggregation method and (16), auxiliary resource actively pushing based DCR aggregation strategy is shown in Fig.5, and the main steps are as follows.

(1) Resource classification. According to the classification of SPRs, human resources and intelligence software are selected as master resources. Other SPRs are used as alternative auxiliary resources.

(2) Meta capability extraction. Extract the value of "spy" class from master resource ontology model as the meta capability of DCR. Meta capability set is established, with each of which one DCR will be created.

(3) Creation of alternative auxiliary resource set. Based on the semantic relation between auxiliary resources and meta capabilities, retrieve auxiliary resource set to find the auxiliary resources matching meta capabilities, which are alternative auxiliary resource set and are actively pushed to resource provider for selection.

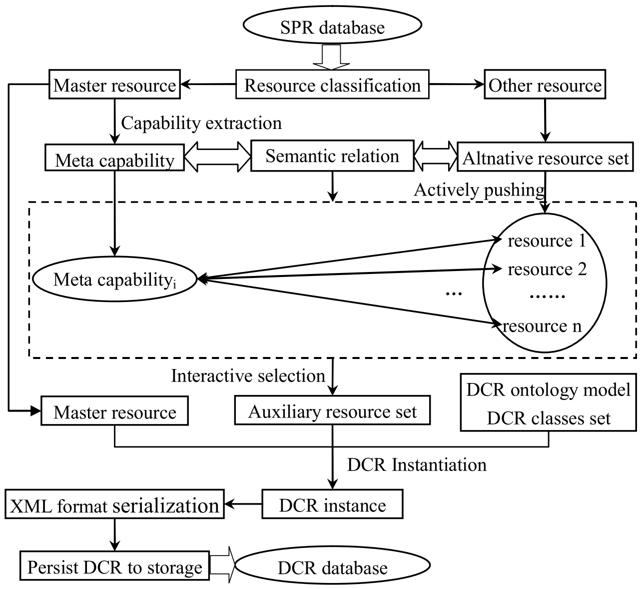


FIGURE 5. Auxiliary resource active push based DCR aggregation strategy.

(4) Interactive selection of auxiliary resources. According to the requirements of meta capability, resource provider selects appropriate resources as auxiliary resources in an interactive way from the list of alternative auxiliary resources actively pushed by the system.

(5) Creation of DCR instance. Master resource and auxiliary resources are input into DCR ontology model and other parameters are initialized, by which one DCR instance is created. For each one in meta capability set, one DCR will be created.

(6) Store DCR instance. Serialize DCR instance to XML data file and store XML data to database.

According to the preceding aggregation process, the DCR aggregation procedure is depicted in **Algorithm 1**, where n represents the number of SPRs and m represents the number of aggregated DCRs.

D. MATCH OF META TASK AND META CAPABILITY BASED CDU AGGREGATION PROCESS

First, decompose typical design tasks into corresponding meta task sequences. And then match meta tasks with the meta capabilities extracted from DCRs one by one, and the DCR sequence that meet the meta task sequences is obtained as the master resource list. So, take the master resource list as the main line, add auxiliary resources and initialize other parameters to complete the establishment of CDU instances.

According to the preceding aggregation strategy, the composition of CDU is expressed as follows.

$$CDU = \sum_{i=1}^{n}(MetaCap_i + MastRes_i + \sum_{j=1}^{m} AuxiRes_j) \quad (17)$$

Algorithm 1 DCR Aggregation Procedure

Input: SPR set $SPR = \{spr_1, spr_2, \ldots spr_n\}$
Output: Aggregated DCR set $DCR = \{dcr_1, dcr_2, \ldots dcr_m\}$
$MetaCap = \{\}$ //define a empty meta capability set
$MastRes = \{\}$ //define a empty master resource set
$AuxiRes = \{\}$ //define a empty auxiliary resource set
$AltAuxiRes = \{\}$ //define a empty alternative auxiliary resource set
$DcrAuxiRes = \{\}$ //define a empty DCR auxiliary resource set
$i \leftarrow 1, j \leftarrow 1$
/*resource classification. Create meta capability and master resource set*/
for ($i \leq$ n) **do**
  if (spr_i == master resource) **then**
    $MetaCap \leftarrow MetaCap \cup spr_i.Spy$ //extract Spy from spr_i and add to $MetaCap$
    $MastRes \leftarrow MastRes \cup spr_i$ //add spr_i to $MastRes$ set
  else
    $AuxiRes \leftarrow AuxiRes \cup spr_i$ //add spr_i to $AuxiRes$ set
  end if
end for
/*create m DCRs*/
m $\leftarrow$ number of $MetaCap$ set, h $\leftarrow$ number of $AuxiRes$ set
for ($j \leq$ m) **do**
  $k \leftarrow 1$
  for ($k \leq$ h) **do** //traverse $AuxiRes$ set
  if ($MetaCap_j == AuxiRes_k.Spy$) **then** //judge if there is semantic relation between the No.j meta capability and the Spy of No.k auxiliary SPR
    $AltAuxiRes = AltAuxiRes \cup AuxiRes_k$ //add $AuxiRes_k$ to $AltAuxiRes$
  end if
end for
/*interactive selection of auxiliary resources*/
$x \leftarrow 1$, z $\leftarrow$ the resource number of $AltAuxiRes$ set
for ($x \leq$ z) **do**
 Interactive selection of auxiliary resources
 if (resource selected) **then**
   $DcrAuxiRes = DcrAuxiRes \cup AuxiRes_x$ // add $AuxiRes_k$ to $AltAuxiRes$
 end if
end for
$dcr_j \leftarrow MetaCap_j \cup MastRes_j \cup DcrAuxiRes$ //define the No.j DCR
Initialize other parameters of dcr_j //Initialize the No.j DCR
end for
XML format serialization of Ontology resource //transform to XML format
Store DCR to storage
return the set of $DCR = \{dcr_1, dcr_2, \ldots dcr_m\}$

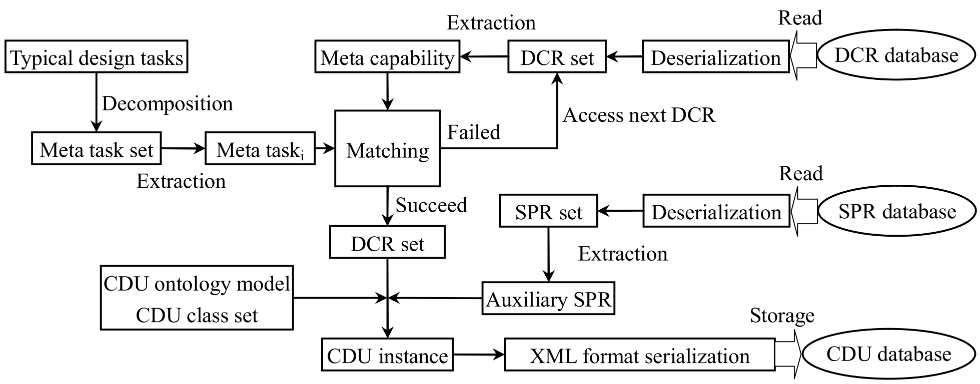


FIGURE 6. Match of meta task and meta capability based CDU aggregation process.

where *MetaCapi* and *MastSpri* represents the meta capability and master resource of the No.i CDU respectively; $AuxiRes_j$ represents the No.j auxiliary resource of the No.i DCR.

The CDU aggregation process is shown in Fig.6, and the main steps are as follows.

(1) Establish meta task sequence. Typical design tasks are analyzed and decomposed into meta task sequences.

(2) Extract meta capabilities from DCR database. Read XML format DCRs described in OWL language in DCR database, then establish DCR instances through deserialization operation, and extract the value of MetaCap class from its class set, which is the meta capability of this DCR.

(3) Match of task and capability, and establish DCR list. The meta tasks in (1) are semantically matched with the meta capabilities in (2) one by one to get the DCR list.

(4) Build CDU instance. Taking the DCR list obtained in (3) as the key resource to complete the design task sequence, search for relevant auxiliary resources, and build CDU instance with CDU ontology model as the template.

(5) Store CDU instance. Serialize CDU instance to XML data file and store XML data to database.

According to the preceding aggregation process, the CDU aggregation procedure is depicted in Algorithm 2.

VI. AGGREGATED RESOURCE EVALUATION METHOD

In order to find the optimal resource, DCRs and CDUs should be evaluated before use. First, the index matrix of DRs is established by calculating the evaluation variables and indexes of each DR, and the member evaluation matrix is established by operating with weight matrix. Secondly, the operation evaluation matrix of DCR or CDU is established according to the operation parameters of resources, and the comprehensive evaluation matrix of DCR or CDU is established on the base of comprehensive evaluation weight matrix. Finally, DRs are optimized according to the final results of comprehensive evaluation matrixes.

A. DR EVALUATION VARIABLES AND INDEX MATRIX

In CMfg system, after a period of activities, design history data and customer evaluation information of DRs will be stored in the database, which are the real refection of DRs' design ability. The key to the comprehensive evaluation of DRs is to determine the reasonable evaluation variables and extract the information reflecting the design ability of resources from the massive data. Design history data, such as the number of service times, finished service times, success times and the service duration of each task, as well as the customer evaluation, are selected as the basis for establishing evaluation variables. Evaluation variables are established as follows.

1) DESIGN MATURITY COEFFICIENT α

As the prime condition to judge whether the design can meet the requirements, the completion rate of design task reflects the maturity of DRs. Maturity coefficient is established as follows.

$$\alpha = \frac{m_f}{m} \tag{18}$$

where m represents times of tasks performed by DR and m_f represents the times of design completed.

In order to avoid the phenomenon of zero as divisor, the initial value of m and m_f in (18) is assigned as 1, so the initial value of α is 1. In the initial stage of resource participation in design, α may decrease rapidly because of the occasional uncompleted task. However, with the increase of m, α will gradually increase, and finally will be infinitely close to 1. The trend of α is shown in Fig.7.

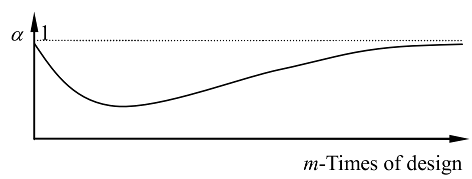


FIGURE 7. The transformation tendency of α.

2) DESIGN SUCCESS RATE COEFFICIENT β

Times of successful design also reflects DR's design ability, which is an important part of the evaluation of DR. Design success rate coefficient is shown as follows.

$$\beta = \frac{m_s}{m} \tag{19}$$

Algorithm 2 CDU Aggregation Procedure

Input: DCR set $DCR = \{dcr_1, dcr_2, \ldots dcr_n\}$ //DCR set extracted from DCR database, which includes n DCRs
Meta task set $TSK = \{tsk_1, tsk_2, \ldots tsk_m\}$ //Meta task set decomposed from typical design task, which includes m tasks
SPR set $SPR = \{spr_1, spr_2, \ldots spr_u\}$ //SPR set extracted auxiliary resource set from SPR database, which includes u SPRs
Output: $CDU = \sum_{i=1}^{m} (MetaCap_i + MastRes_i + \sum_{j=1}^{w} AuxiRes_j)$ //w is the number of auxililary resource of $MastRes_i$
$MetaCap = \{\}$ //define a empty meta capability set
$AuxiRes = \{\}$ //define a empty auxiliary resource set
/*list all the meta capability*/
$i \leftarrow 1$
for ($i \leq$ n) **do** //extract meta capability from DCR set
$MetaCap \leftarrow MetaCap \cup dcr_i.MetaCap$ //extract the class of $MetaCap$ from dcr_i and add it to meta capability set
end for
/*search auxiliary resource*/
$x \leftarrow 1$
for ($x \leq$ u) **do** //traverse SPR set to search all the auxiliary resource
  if (spr_x == auxiliary resource) **then**
    $AuxiRes \leftarrow AuxiRes \cup spr_x$ //add spr_x to $AuxiRes$ set
  end if
end for
/*construct CDU set */
$j \leftarrow 1$, h $\leftarrow$ number of $AuxiRes$
for ($j \leq$ m) **do**
  $k \leftarrow 1$
  for($k \leq$ n) **do** //for each meta task, find the matching DCR
    if ($MetaCap_k == tsk_j$) **then**
      $MastRes \leftarrow dcr_i$ //add the searched DCR to master resource
    end if
  end for
  /*for $task_j$, match auxiliary resource*/
  $z \leftarrow 1$
  $CduAuxiRes = \{\}$ //define a empty array for matched auxiliary to $task_j$
  for ($z \leq h$) **do** //traverse $AuxiRes$ set and search auxiliary resource for $task_j$
    if ($MetaCap_j == AuxiRes_z.Spy$) **then** //judge if there is semantic relation between the No.j meta capability and the Spy of No.k auxiliary SPR
      $CduAuxiRes = CduAuxiRes \cup AuxiRes_z$ //add $AuxiRes_k$ to $BackSpr$
    end if
  end for
  $CDU = CDU + MetaCap_j \cup MetaRes \cup CduAuxiRes$ //add the new created CDU
end for

Algorithm 2 *(Continued.)* CDU Aggregation Procedure

Initialize other parameters of cdu
XML format serialization of Ontology resource //transform ontology to XML
Store CDU to database
return $CDU = \sum_{i=1}^{m} (MetaCap_i + MastRes_i + \sum_{j=1}^{w} AuxiRes_j)$

where m represents times of tasks performed by DR and m_s represents times of successful design.

Similar to (18), the initial value of β is 1. In the early stage of participating in the design process, β may decrease rapidly due to design failures. However, with the increase of m, β will gradually increase, and finally will be infinitely close to 1. The trend of β is shown in Fig.8.

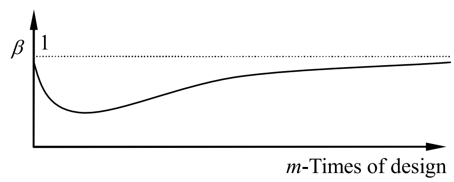


FIGURE 8. The transformation tendency of β.

3) DESIGN STABILITY COEFFICIENT γ

Due to the limitation of function, the design tasks accepted by the same DR are generally relatively stable, and each time the design tasks are similar. Under this premise, the closer the time spent on each design task is, the higher the stability of the DR is. When requester invokes this DR, the more likely the design task will be accomplished on time

In order to describe the design stability, σ is established firstly, as shown in equation (20):

$$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\frac{x_i - \mu}{\mu})^2} \tag{20}$$

where m represents the times of tasks performed by DR; x_i represents the time used during the No.i task. μ represents the average time of m tasks.

σ reflects the degree of difference in design time when performing multiple tasks. The ratio of design time difference to design time is used to reflect the proportion difference in design time. The greater the difference is, the more unstable the design process is. Under normal circumstances, because the difference of design time will be far less than the mean value of design time, the calculated value of σ should be far less than 1.

In order to reflect design stability more reasonably, design stability coefficient γ is established as follows.

$$\gamma = 1 - \sigma \tag{21}$$

To avoid the phenomenon of zero as divisor, when the resource is invoked for the first time, the initial values of (20)

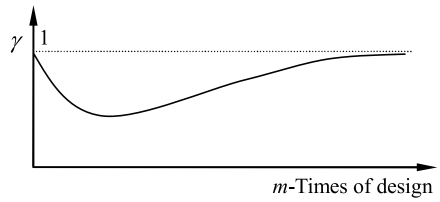


FIGURE 9. The transformation tendency of γ.

is: m=1, $x_i = \mu \neq 0$. So, the initial value of σ is 0 and corresponding the value of γ is 1. In the initial period, design stability parameters γ will decrease rapidly. With the increase of m, γ will be infinitely close to 1. The trend is shown in Fig.9.

4) EXPERIENTIAL COEFFICIENT ε

The total number of times that DR participates in design reflects the experience degree of DR. Experiential coefficient ε is shown in formula (22):

$$\varepsilon = 1 - \frac{1}{\sqrt{m}} \tag{22}$$

The purpose of using the expression shown in (22) is to make ε approach to 1 infinitely with the increase of m, but not too fast.

Similar to (18), the initial value of ε is 0, which indicates that this DR has not participated in design before. With the increase of m, ε will be infinitely close to 1. The trend is shown in Fig.10.

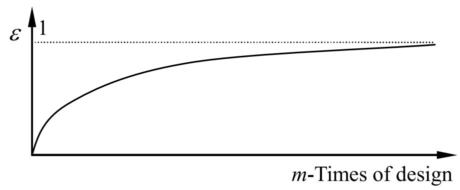


FIGURE 10. The transformation tendency of ε.

5) CAPABILITY INDEX A

The above four coefficients, design maturity coefficient α, design success rate coefficient β, design stability coefficient γ and experiential coefficient ε, are all related to the design capability of DR, and they record the information related to their design ability through their own design trajectory. DR capability index A is constructed from the above four coefficients, as shown in equation (23):

$$A = \alpha \times \beta \times \gamma \times \varepsilon \tag{23}$$

where A represents DR capability index; α represents design maturity coefficient; β represents design success rate coefficient; γ represents design stability coefficient ε represents experiential coefficient.

It can be seen from the value range of the four evaluation variables that the initial value of DR capability index A is 0. With the increase of design times, its value gradually increases, and finally it is infinitely close to 1. Its change range is shown in Fig.11.

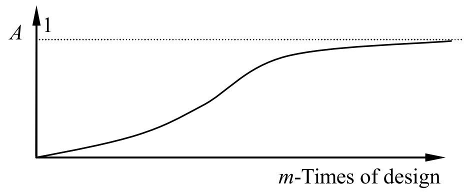


FIGURE 11. The transformation tendency of A.

6) EVALUATION INDEX E

The mean value of user's evaluation score is used as the evaluation index E of DR, as shown in equation (24).

$$E = \frac{\sum_{i=1}^{m} Eva_i}{m} \tag{24}$$

where m represents the times of tasks performed by DR. Eva_i represents the No.i evaluation score provided by user.

User evaluates DR with a full score of 100 each time. As times of design increases, the value of each DR will gradually stabilizes at a value n, which reflects the real capability of resource. The trend of evaluation index E is shown in Fig.12.

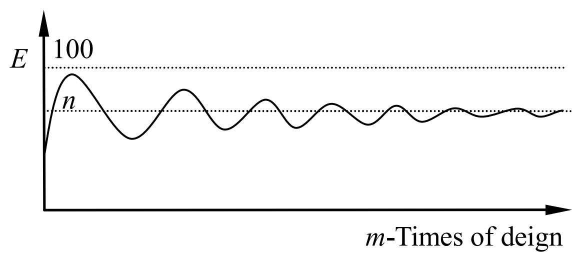


FIGURE 12. The transformation tendency of E.

7) COST INDEX C

Design cost is reflected in labor price and duration time. DR cost index C is established as follows.

$$C = \frac{\sum_{i=1}^{m} (t_i \times Pri_i)}{m} \tag{25}$$

where m represents the times of tasks performed by DR. t_i represents the duration of the No.i design. Pri_i represents the labor cost of the No.i design.

Assuming that labor price of each design is the same, DR cost index will be proportional to design duration. Therefore, as the times of design tasks increases, DR cost index will be stable at a certain value. The trend of cost index C is shown in Fig.13.

According to the preceding analysis, capacity index A, evaluation index E and cost index C are used to evaluate the comprehensive performance of DR. A reflects the design capacity of DR, which is regarded as the first evaluation index of resource evaluation. E reflects the customer's satisfaction

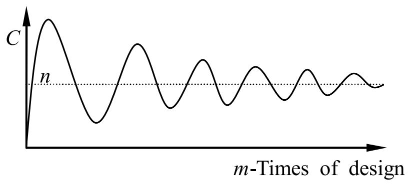


FIGURE 13. The transformation tendency of *C*.

with the use of DR, which is regarded as the second evaluation index. C reflects the price information of DR, which is regarded as the third evaluation index. Based on the above three indexes, DR index matrix is established as follows.

$$M = \begin{bmatrix} A \\ E \\ C \end{bmatrix} \tag{26}$$

where M represents DR index matrix; A represents DR capability index; E represents DR evaluation index; C represents DR cost index.

The DR index matrix can reflect the comprehensive performance of DR from three aspects: capability, evaluation and cost. It can be used for the evaluation of SPR when DCR is established, or as the basis for resource requesters to search and optimize DCR.

B. DCR EVALUATION METHOD AND DCR COMPREHENSIVE EVALUATION MATRIX

DCR evaluation includes two aspects: DCR member evaluation and DCR operation evaluation.

1) DCR MEMBER EVALUATION MATRIX M_{Mbr}

Suppose a DCR is composed of m SPRs, and the master resource and auxiliary resources of DCR are evaluated from three aspects: capability index A, evaluation index E and cost index C. Establish DCR member index matrix, as follows.

$$M_{SPR} = \begin{bmatrix} A_1 & A_2 & \ldots & A_i & \ldots & A_m \\ E_1 & E_2 & \ldots & E_i & \ldots & E_m \\ C_1 & C_2 & \ldots & C_i & \ldots & C_m \end{bmatrix} \tag{27}$$

where M_{SPR} represents DCR member index matrix; A_i represents the No.i capability index; E_i represents the No.i evaluation index; C_i represents the No.i cost index; m represents that DCR is composed by m member resource, including master resource and auxiliary resources.

For the m resources that compose DCR, because of their different contributions to the formation of DCR, the weight of each index to evaluation matrix is not equal when building DCR. In order to reflect the different contribution of each resource, member weight matrix is introduced, as shown in (28):

$$\omega_{SPR} = \begin{bmatrix} \omega_1 & \omega_2 & \ldots & \omega_i & \ldots & \omega_m \end{bmatrix} \quad \left(\sum_{i=1}^{m} \omega_i = 1\right) \tag{28}$$

where ω_i represents the No.i weight coefficient; m represents that DCR is composed by m member resources.

According to (27) and (28), the calculation formula of DCR member evaluation matrix is constructed as follows.

$$M_{Mbr} = \begin{bmatrix} A_1 & A_2 & \ldots & A_i & \ldots & A_m \\ E_1 & E_2 & \ldots & E_i & \ldots & E_m \\ C_1 & C_2 & \ldots & C_i & \ldots & C_m \end{bmatrix} \times \begin{bmatrix} \omega_1 \\ \omega_2 \\ \ldots \\ \omega_i \\ \ldots \\ \omega_m \end{bmatrix} \tag{29}$$

where M_{Mbr} represents DCR member evaluation matrix; M_{SPR} represents DCR member index matrix; ω_{SPR} represents the transposed matrix of member resource weight matrix.

Formula (29) reflects the comprehensive performance of each DCR member in capacity, evaluation and cost. So, the comprehensive capacity index, comprehensive evaluation index and comprehensive cost index of each member resource of DCR are shown in (30), (31), and (32) respectively.

$$A_{Mbr} = A_1 \times \omega_1 + A_2 \times \omega_2 + \ldots + A_i \times \omega_i + \ldots + A_m \times \omega_m \tag{30}$$

$$E_{Mbr} = E_1 \times \omega_1 + E_2 \times \omega_2 + \ldots + E_i \times \omega_i + \ldots + E_m \times \omega_m \tag{31}$$

$$C_{Mbr} = C_1 \times \omega_1 + C_2 \times \omega_2 + \ldots + C_i \times \omega_i + \ldots + C_m \times \omega_m \tag{32}$$

The preceding indexes in (30) - (32) reflect the contribution of each member to DCR, and the weight of each member's contribution to DCR is also reflected by the weight coefficient.

2) DCR OPERATION EVALUATION MATRIX M_{RUN}

When available DCRs being searched, they can be evaluated based on their usage history. Alternative DCRs that fail to meet the specified threshold will be discarded or decomposed. DCR operation evaluation matrix M_{RUN} is used to express its historical operation, shown as follows.

$$M_{Run} = \begin{bmatrix} A_{Run} \\ E_{Run} \\ C_{Run} \end{bmatrix} \tag{33}$$

where A_{RUN}, E_{RUN}, and C_{RUN} represent DCR's comprehensive capacity index, comprehensive evaluation index and comprehensive cost index respectively. Similar with (18) - (32), the values of the above index can be calculated according to the historical data of DCRs.

3) DCR COMPREHENSIVE EVALUATION MATRIX M_{Cph}

According the preceding DCR member evaluation matrix and operation evaluation matrix, DCR comprehensive evaluation matrix M_{Cph} is constructed, as shown in (34), which is an important evaluation indicator in the process of DCR optimization.

$$M_{Cph} = \begin{bmatrix} M_{Mbr} & M_{Run} \end{bmatrix} \times \omega_{Cph}^{T} \tag{34}$$

where $[M_{Mbr}\ \ M_{Run}]$ represents DCR member evaluation matrix and operation evaluation matrix; ω_{Cph} represents the transposed matrix of comprehensive evaluation weight matrix, and the definition of ω_{Cph} is as follows.

$$\omega_{\mathrm{Cph}} = [\,\omega_{Mbr}\quad \omega_{Run}\,]\,(\omega_{Mbr} + \omega_{Run} = 1) \tag{35}$$

where ω_{Mbr} and ω_{Run} represent the weight of DCR member evaluation matrix and operation evaluation matrix respectively, and the sum of the two is equal to 1.

According to the above parameters, the result of (34) is shown in (36), as shown at the bottom of the next page.

From (36), it can be inferred that DCR comprehensive evaluation matrix M_{Cph} can reflect not only the design history data and customer evaluation of each member resource and DCR, but also the weight proportion of member resource and DCR running data. DCR M_{Cph} can objectively reflect the performance of DCR in design capability, customer evaluation and cost index.

C. CDU EVALUATION METHOD AND ITS COMPREHENSIVE EVALUATION MATRIX

The comprehensive evaluation of CDU also includes two parts: member evaluation and operation evaluation.

1) CDU MEMBER EVALUATION MATRIX M_{SPRDCR}

CDU member evaluation is a comprehensive evaluation of the member resources of CDU at three aspects: capability index, evaluation index and cost index, in order to test whether its composition can reach the required threshold.

Assuming that the creation of CDU involves m DCRs and n independent SPRs, establish CDU member index matrix M_{SPRDCR}, as shown in (37), as shown at the bottom of the next page.

Where A_{mi}, E_{mi}, and C_{mi} represent the capability index, evaluation index and cost index of the No.i SPR respectively; A_{nj}, E_{nj}, and C_{nj} represent the capability index, evaluation index and cost index of the No.j DCR respectively; m and n represents the number of SPR and DCR that make up CDU.

Due to the different capability of each member resource, their contributions to the construction of CDU vary. Therefore, member weight matrix ω_{SPRDCR} are introduced, as shown in (38), as shown at the bottom of the next page, to express the importance of each member resource. where ω_{mi} and ω_{nj} represent the weight of the No.i DCR member resource and the No.j SPR member resource; m and n represent the number of DCR and SPRs respectively.

CDU member evaluation matrix is constructed by multiplying CDU member index matrix and member weight matrix, as shown in (39).

$$M_{CDUMbr} = M_{SPRDCR} \times \omega_{SPRDCR}^{T} \tag{39}$$

According to (37) and (38), CDU member evaluation matrix is as follows.

The first row in (40), as shown at the bottom of the next page, reflects the comprehensive performance of each member resource of the newly established CDU in terms of design capability, the second row reflects customer evaluation, and the last row reflects cost.

2) CDU OPERATION EVALUATION MATRIX M_{CDURun}

When searching for optimal CDU, resource requester evaluates the operation status of CDU after its establishment to check whether its design capability, customer evaluation and design cost meet the requirements.

CDU operation evaluation matrix M_{CDURun} consists of historical record information and customer evaluation information during CDU operation, as shown in (41):

$$M_{CDURun} = \begin{bmatrix} A_{CDURun} \\ E_{CDURun} \\ C_{CDURun} \end{bmatrix} \tag{41}$$

where A_{CDURun}, E_{CDURun}, and C_{CDURun} represent operation capability index, operation evaluation index and cost index of CDU respectively. Refer to the formulas of each index in Section 6.1, as well as the design history data and customer evaluation information generated during the operation of CDU, the value of A_{CDURun}, E_{CDURun}, and C_{CDURun} can be calculated.

3) CDU COMPREHENSIVE EVALUATION MATRIX M_{CDUCph}

CDU comprehensive evaluation matrix M_{CDUCph} consists of two parts: resource member matrix data and resource operation matrix data. Considering the different importance of these two parts, comprehensive evaluation weight coefficient matrix is introduced, as shown in (42).

$$\omega_{CDUCph} = [\,\omega_{CDUMbr}\quad \omega_{CDURun}\,] \times (\omega_{CDUMbr} + \omega_{CDURun} = 1) \tag{42}$$

According to the preceding member matrix and operation matrix, as well as weight coefficient matrix shown in (42), CDU comprehensive evaluation matrix M_{CDUCph} is constructed as follows.

$$M_{CDUCph} = [\,M_{CDUMbr}\quad M_{CDURun}\,] \times \omega_{CDUCph}^{T} \tag{43}$$

According to the preceding parameters, the result of CDU comprehensive evaluation matrix is shown in (44), as shown at the bottom of the next page.

M_{CDUCph} can comprehensively reflect the historical data and customer evaluation data of CDU operation, as well as the relevant information of all its members. At the same time, different weight coefficients can be set according to different components and the importance of each member resource, which truly reflect the operation status of CDU.

VII. CASE STUDY

To validate the proposed approach, a case study is performed in collaboration with a machinery enterprise. This enterprise is specialized in the design and processing of large-scale gearbox and other mechanical products, with professional designers and production equipments. However, when their orders fluctuate, it needs to outsource part of their design work, or undertake some design tasks from other enterprises, which requires that enterprises can output design services to other factories or receive external design services. Resource modeling and aggregation is the premise of resource virtualization and servitization and the key to resource output or access. Based on such an application scenario, a CMfg prototype system is developed, which accomplishes the work of modeling, aggregation and evaluation of design resource.

Fig.14 shows an application screenshot of the design resource modeling and aggregation prototype that mainly consists of DR access, aggregation, evaluation, design task decomposition, matching of task and cloud service. Its architecture is shown in Fig.15.

Resource access module completes the access of SPRs. Figure 16 shows the human resources access interface. After access, data serialization and storage program are invoked to serialize SPR ontology model into XML format and store it in database. OWL language is used to serialize DR ontology. Fig.17 and 18 show the code of class and class instance of human resource instance.

Resource provider can view the list of SPRs provided by himself, as shown in Fig.19. Each record in the figure shows the overview of one SPR. Click the link of "click and check" of each SPR to display its details, as shown in Fig.20, which not only shows the basic information of resource input by resource provider, but also shows service history information and customer evaluation accumulated during resource operation.

The aggregation and storage of DCR are accomplished in the aggregation module. According to the master resources retrieved from SPR database, its meta capabilities are extracted. And then the association with auxiliary resources is establishes. DCR is generated, which is serialized and stored in database, shown in Fig.21. DCR list is shown in Fig.22, which shows the name of DCR, its master resource, function description and auxiliary resources list.

Resource evaluation module is used for quantitative evaluation and optimization of DCR or CDU. By calculating evaluation variables, establishing evaluation matrix and matrix operation, capability index, evaluation index and cost index of candidate aggregated resource can be calculated accurately, which provides quantitative basis for optimization. Taking DCR searching and optimization of high-speed and heavy-duty gearbox design, a common design task, as an example, this paper illustrates the aggregated resource evaluation algorithm. The needed member resources mainly include: gear profile modification engineer, gearbox structure design

$$M_{Cph} = \begin{bmatrix} (A_1 \times \omega_1 + A_2 \times \omega_2 + \ldots + A_m \times \omega_m) \times \omega_{Mbr} + A_{Run} \times \omega_{Run} \\ (E_1 \times \omega_1 + E_2 \times \omega_2 + \ldots + E_m \times \omega_m) \times \omega_{Mbr} + E_{Run} \times \omega_{Run} \\ (C_1 \times \omega_1 + C_2 \times \omega_2 + \ldots + C_m \times \omega_m) \times \omega_{Mbr} + C_{Run} \times \omega_{Run} \end{bmatrix} \tag{36}$$

$$M_{SPRDCR} = \begin{bmatrix} A_{m1} & A_{m2} & \ldots & A_{mi} & \ldots & A_{mm} & A_{n1} & A_{n2} & \ldots & A_{nj} & \ldots & A_{nn} \\ E_{m1} & E_{m2} & \ldots & E_{mi} & \ldots & E_{mn} & E_{n1} & E_{n2} & \ldots & E_{nj} & \ldots & E_{nn} \\ C_{m1} & C_{m2} & \ldots & C_{mi} & \ldots & C_{mn} & C_{n1} & C_{n2} & \ldots & C_{nj} & \ldots & C_{nn} \end{bmatrix} \tag{37}$$

$$\omega_{SPRDCR} = \begin{bmatrix} \omega_{m1} & \omega_{m2} & \ldots & \omega_{mi} & \ldots & \omega_{mm} & \omega_{n1} & \omega_{n2} & \ldots & \omega_{nj} & \ldots & \omega_{nn} \end{bmatrix} \quad (\sum_{i=1}^{m} \omega_{mi} + \sum_{j=1}^{n} \omega_{nj} = 1) \tag{38}$$

$$M_{CDUMbr} = \begin{bmatrix} A_{m1} \times \omega_{m1} + \ldots + A_{mi} \times \omega_{mi} + \ldots + A_{mm} \times \omega_{mm} + A_{n1} \times \omega_{n1} + \ldots + A_{nj} \times \omega_{nj} + \ldots + A_{nn} \times \omega_{nn} \\ E_{m1} \times \omega_{m1} + \ldots + E_{mi} \times \omega_{mi} + \ldots + E_{mm} \times \omega_{mm} + E_{n1} \times \omega_{n1} + \ldots + E_{nj} \times \omega_{nj} + \ldots + E_{nn} \times \omega_{nn} \\ C_{m1} \times \omega_{m1} + \ldots + C_{mi} \times \omega_{mi} + \ldots + C_{mm} \times \omega_{mm} + C_{n1} \times \omega_{n1} + \ldots + C_{nj} \times \omega_{nj} + \ldots + C_{nn} \times \omega_{nn} \end{bmatrix} \tag{40}$$

$$M_{CDUCph} = \begin{bmatrix} (\sum_{i=1}^{m} (A_{mi} \times \omega_{mi}) + \sum_{j=1}^{n} (A_{nj} \times \omega_{nj})) \times \omega_{CDUMbr} + A_{CDURun} \times \omega_{CDURun} \\ (\sum_{i=1}^{m} (E_{mi} \times \omega_{mi}) + \sum_{j=1}^{n} (E_{nj} \times \omega_{nj})) \times \omega_{CDUMbr} + E_{CDURun} \times \omega_{CDURun} \\ (\sum_{i=1}^{m} (C_{mi} \times \omega_{mi}) + \sum_{j=1}^{n} (C_{nj} \times \omega_{nj})) \times \omega_{CDUMbr} + C_{CDURun} \times \omega_{CDURun} \end{bmatrix} \tag{44}$$

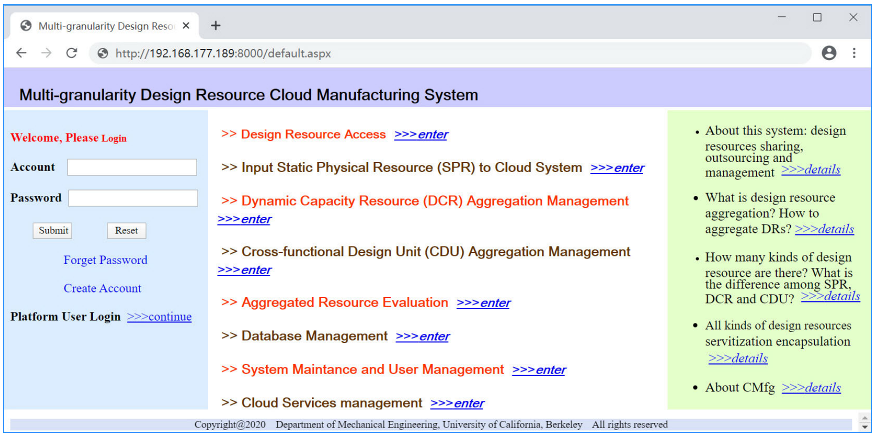


FIGURE 14. Interface view of DR modeling and aggregation system.

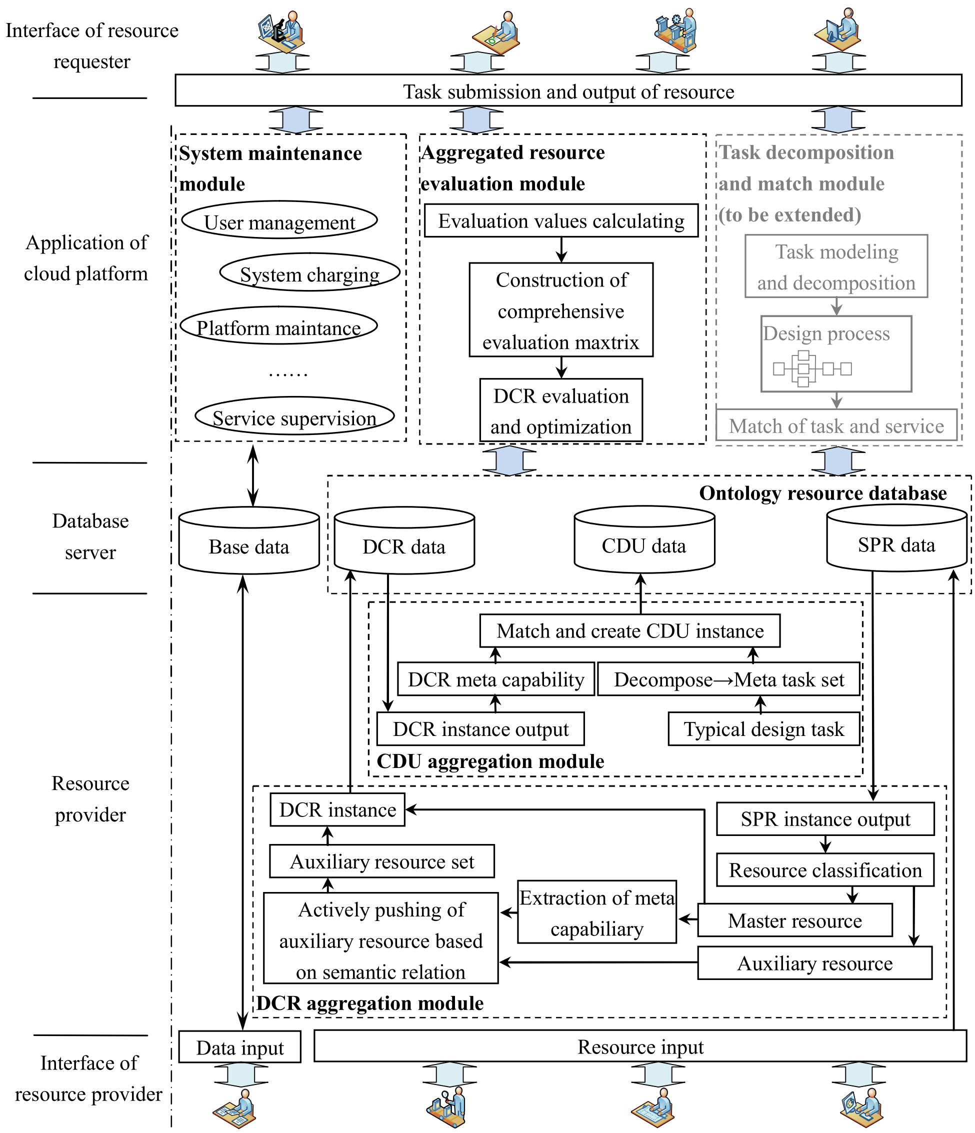


FIGURE 15. Architecture of DR CMfg system.

engineer, gearbox dynamic analysis software, finite element analysis software and computer workstation, etc.

According to the requirements of the above gearbox design DCR, when searching in the CMfg system, it is assumed

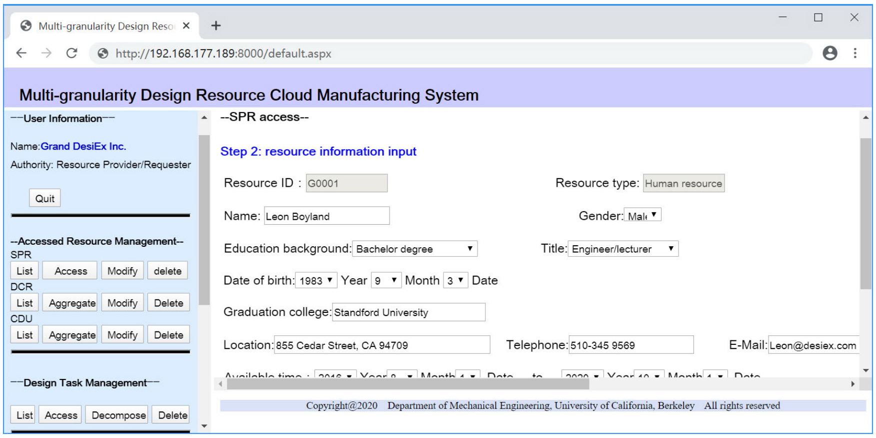


FIGURE 16. The interface of human resource access.

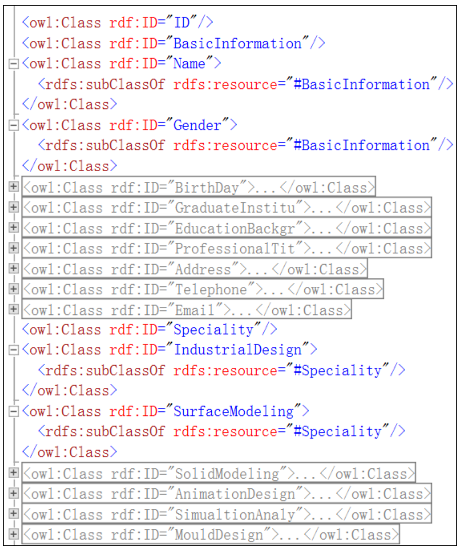


FIGURE 17. The class definition of DR ontology.

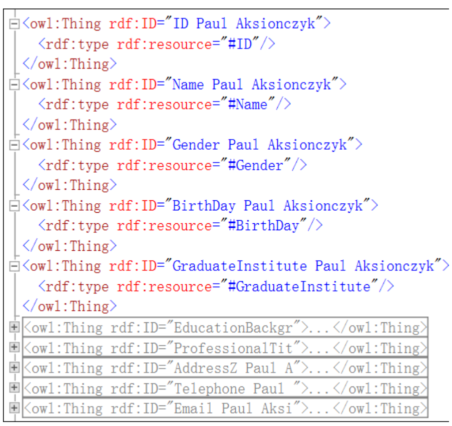


FIGURE 18. The building of DR instance of class.

that two DCRs meeting the requirements are found. The first DCR is named "ROMAX gearbox analysis working group". Its member resources mainly include: Paul, which is gear surface modification engineer, Boris, which is gear

TABLE 1. Member resource attribute information list of "ROMAX gear box analysis group."

	SerNr	*FshNr*	*SucNr*	*[LoT]*	*[Eva]*
Paul	20	18	14	10 8 12 15 6 9 10 11 10 9 11 12 8 14 9 10 9 11 13 18	70 89 71 74 70 80 82 78 89 70 90 72 69 75 72 89 77 85 84 83
Boris	15	14	10	8 11 9 10 10 9 11 12 13 7 8 10 12 13 7	72 75 72 90 69 70 78 62 90 86 72 71 89 74 72
ROMAX 14.5	16	9	9	11 6 7 8 10 9 10 12 13 9 7 8 7 8 7 11	80 82 78 89 70 90 72 69 75 70 78 62 90 86 85 76
ANSYS 16.0	10	8	8	9 8 10 11 12 7 9 13 10 11	89 70 90 72 69 75 70 86 76 88
HP Z640	8	6	5	12 10 9 11 8 15 10 14	78 62 90 86 85 76 85 69

box structure design engineer, ROMAX 14.5, ANSYS 16.0, and HP Z640. Its attributes are shown in Table 1, and the attributes of DCR are shown in Table 2. The second DCR is named "MASTA gearbox analysis working group". Its member resources mainly include: Leon, which is gear surface modification engineer, Eric, which is gearbox structure design engineer, MASTA 5.0, ANSYS 15.0, and Dell T7910. Its attributes are shown in Table 3, and the attribute of DCR is shown in Table 4. Where, SerNr refers to the times that the resource performs design tasks, FshNr refers to the times that the resource completes design tasks, SucNr refers to the times that the resource successfully completes design tasks, [LoT] refers to array of time that the resource completes design tasks, and [Eva] refers to the array of evaluation score of the resource.

The above two candidate resources are quantitatively optimized by resource evaluation method. According to (18) - (22), design maturity coefficient α, design success rate coefficient β, design stability coefficient γ and experiential coefficient ε of two DCR member resources are

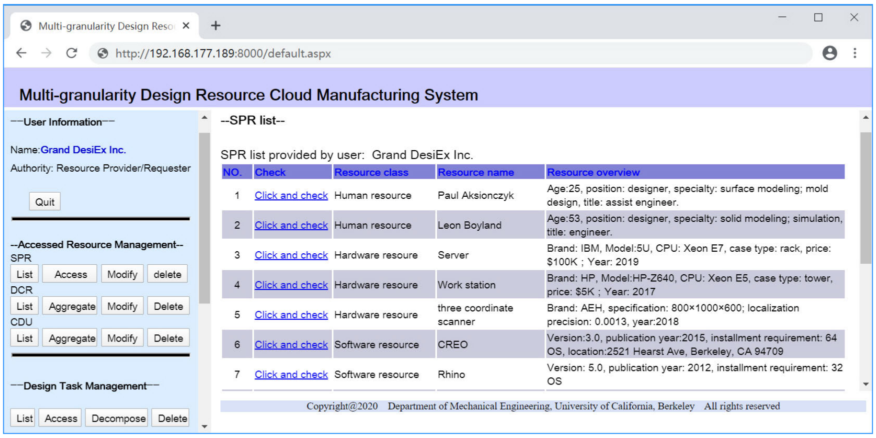


FIGURE 19. List of SPRs.

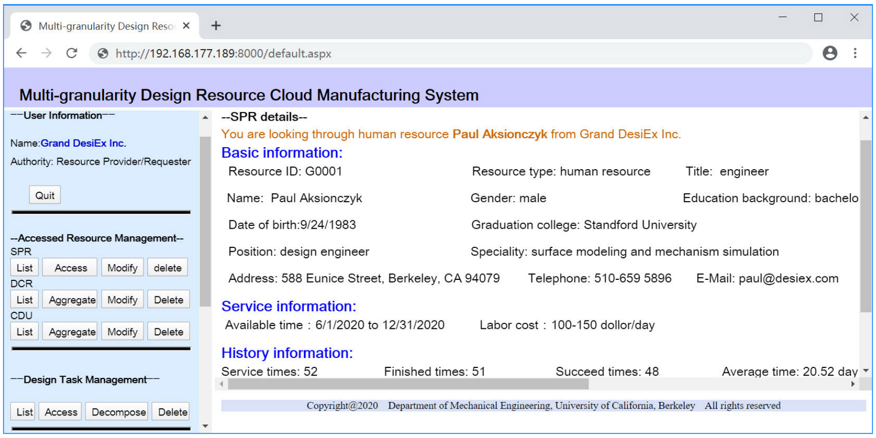


FIGURE 20. Details about SPR.

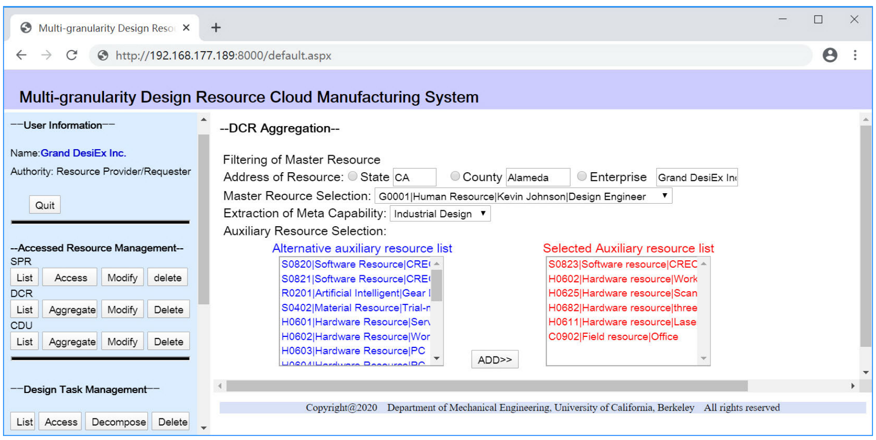


FIGURE 21. Process of DCR aggregation.

calculated respectively. According to (23) - (25), capability index A, evaluation index E and cost index C of the two DCR member resources are calculated, which is shown as in Table 5 and 6.

The weight matrix of the two DCRs is represented as $\omega_{SPR} = [0.4\ 0.2\ 0.2\ 0.1\ 0.1]$. Then, member resource indexes of the two DCRs are calculated as shown in Table 7.

TABLE 2. Attribute information list of "ROMAX gear box analysis group."

	SerNr	FshNr	SucNr	[LoT]	[Eva]
ROMAX	4	3	3	7 9 13 10	89 70 90 72

According to Table 2 and 4, design maturity coefficient α, design success rate coefficient β, design stability coefficient

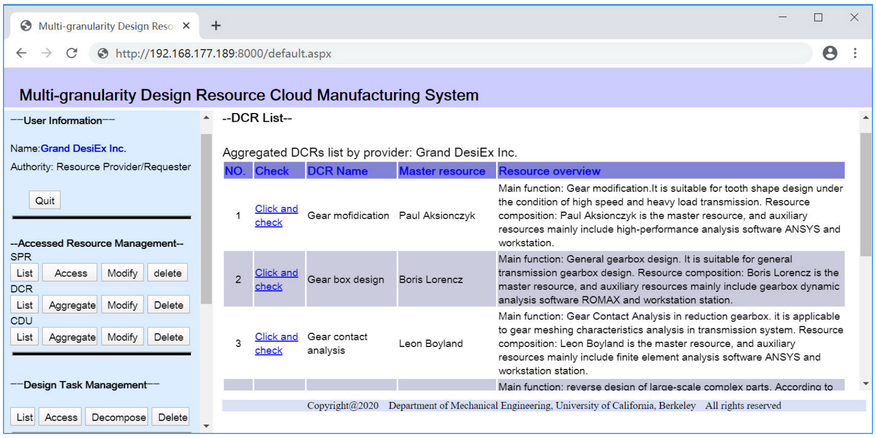


FIGURE 22. DCR list.

TABLE 3. Member resource attribute information list of "MASTA gear box analysis group."

	SerNr	FshNr	SucNr	[LoT]	[Eva]
Leon	15	13	13	12 6 9 10 8 10 11 10 9 11 12 8 14 9 10	80 82 78 89 70 89 71 74 70 70 90 72 69 75 85
Eric	12	10	8	8 11 9 10 10 9 11 12 13 7 8 10	69 70 78 72 75 72 80 62 90 86 72 71
MASTA 5.0	10	9	7	11 6 7 8 10 9 10 12 13 9	72 69 80 82 78 89 70 85 75 70
ANSYS 15.0	8	6	6	11 12 9 8 10 7 9 13	70 90 72 69 70 86 76 88
Dell T7910	10	8	6	12 10 9 11 8 15 10 14 7 8	78 62 90 86 85 76 85 69 75 72

TABLE 4. Service information list of the second DCR.

	SerNr	FshNr	SucNr	[LoT]	[Eva]
MASTA	6	6	5	7 9 13 10 8 9	89 70 90 72 85 89

TABLE 5. Member parameter of the first DCR.

	A	E	C
Paul	0.37	78.45	10.75
Boris	0.38	71.38	10.00
ROMAX 14.5	0.20	78.25	8.94
ANSYS 16.0	0.36	78.50	10.00
HP Z640	0.22	78.88	11.13

γ and experiential coefficient ε of two DCRs are calculated respectively, which are shown in Table 8. According to (23) - (25), capability index A, evaluation index E and cost index C of two DCRs are calculated, which is shown as in Table 9.

The weight matrix of DCR comprehensive evaluation is represented as $\omega_{Cmp} = [0.7\ 0.3]$. Two DCR comprehensive evaluation indexes are calculated, shown in Table 10.

TABLE 6. Member parameter of the second DCR.

	A	E	C
Leon	0.45	77.60	9.93
Eric	0.33	74.75	9.83
MASTA 5.0	0.34	77.00	9.5
ANSYS 15.0	0.30	77.63	9.88
Dell T7910	0.25	77.80	10.4

TABLE 7. Member resource parameter of the two DCRs.

	A	E	C
Member resource of No.1 DCR	0.32	77.04	10.20
Member resource of No.2 DCR	0.37	76.93	9.87

TABLE 8. Coefficient of DCRs.

	α	β	γ	ε
No.1 DCR	0.75	0.75	0.78	0.50
No.2 DCR	1	0.83	0.80	0.59

TABLE 9. Parameter of DCRs.

	A	E	C
No.1 DCR	0.22	80.25	9.75
No.2 DCR	0.39	82.5	9.33

TABLE 10. Compare between two DCRs.

	A	E	C
No.1 DCR	0.29	78.00	10.07
No.2 DCR	0.38	78.60	9.71

It can be seen from Table 10 that the comprehensive capability index A and evaluation index E of the second DCR are higher than that of the first DCR, and the comprehensive cost index C is also better than that of the first DCR. Therefore, it can be concluded that the second DCR is better than the first.

VIII. CONCLUSIONS

In summary, aiming at the service sharing problem of DRs in CMfg system, a novel method of resource granularity division, resource modeling, resource aggregation and evaluation is proposed, which provides the underlying foundation for resource service sharing. Based on the study of the demand of CMfg system for coarse granularity DRs, a multi-granularity DR model is proposed, which is convenient for resource management and utilization. The ontology model of each granularity resource is defined, in which the abstract description of resource is realized by using formal expression method. The aggregation strategies of DCR and CDU are established, which accomplish the capability aggregation of DCR and the unit aggregation of CDU. Resource evaluation method is proposed, which is used to optimize the aggregated resources. Finally, a prototype system is established to accomplish the function of DRs multi-granularity modeling and aggregation, and the example of aggregated resource evaluation verifies the effectiveness of resource evaluation method.

Our future work will focus on the service-oriented research of aggregation resources, and the matching of aggregation service resources and coarse granularity design tasks. In addition, the prototype system needs to be improved to accomplish the CMfg application of DRs.

REFERENCES

[1] J. Siderska and K. S. Jadaan, "Cloud manufacturing: A service-oriented manufacturing paradigm. A review paper," *Eng. Manage. Prod. Services*, vol. 10, no. 1, pp. 22–31, Mar. 2018.

[2] B. Li, "Cloud manufacturing: A new service-oriented networked manufacturing model," *Comput. Integr. Manuf. Syst.*, vol. 16, no. 1, pp. 1–7, 2010.

[3] Y. Y. Wang, L. E. Zhang, and L. Zhao, "Cloud design resource scheduling mechanism and algorithm based on negative feedback," *Comput. Integr. Manuf. Syst.*, vol. 23, no. 6, pp. 1196–1204, Jun. 2017.

[4] R. S. G. Yan org, B. He, Y. C. Hua, F. F. Li, L. X. Lu, S. X. Li, and L. M. Li, "Cloud service-based collaborative design resource Management," *Adv. Mech. Des.*, vol. 2017, pp. 539–552, Oct. 2017.

[5] Y. Zhang, G. Zhang, Y. Liu, and D. Hu, "Research on services encapsulation and virtualization access model of machine for cloud manufacturing," *J. Intell. Manuf.*, vol. 28, no. 5, pp. 1109–1123, Jun. 2017.

[6] R. Y. Zhong, L. Wang, and X. Xu, "An IoT-enabled real-time machine status monitoring approach for cloud manufacturing," *Procedia CIRP*, vol. 63, pp. 709–714, 2017.

[7] Y. Liu, L. Wang, X. V. Wang, X. Xu, and P. Jiang, "Cloud manufacturing: Key issues and future perspectives," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 9, pp. 858–874, Sep. 2019.

[8] L. Ren, L. Zhang, L. Wang, F. Tao, and X. Chai, "Cloud manufacturing: Key characteristics and applications," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 6, pp. 501–515, Jun. 2017.

[9] X. B. Cao, "Study on key technologies of complicated product integrated design service platform based on design resource's multiple agglomeration methods," Ph.D. dissertation, Dept. Aerosp. Eng., Beijing Inst. Technol., Beijing, China, 2015.

[10] K. Li, "Research on operation mechanism and key techniques of cloud manufacturing system for small and dedium-sized enterprise," Ph.D. dissertation, Dept. Mechatron. Eng., Xidian Univ., Xi'an, Shanxi, China, 2018.

[11] N. Liu and X. Li, "Granulation-based resource classification in cloud manufacturing," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 229, no. 7, pp. 1258–1270, Jul. 2015.

[12] C. Zhao, L. H. Wang, and X. S. Zhang, "Service agent networks in cloud manufacturing: Modeling and evaluation based on set-pair analysis," *Robot. Cim.-Int. Manuf.*, vol. 65, pp. 1–9, 2020.

[13] Y. X. Li and X. F. Yao, "Cloud manufacturing service composition and formal verification based on extended process calculus," *Adv. Mech. Eng.*, vol. 10, no. 6, pp. 1–16, 2018.

[14] Y. Wang, Z. Dai, W. Zhang, S. Zhang, Y. Xu, and Q. Chen, "Urgent task-aware cloud manufacturing service composition using two-stage biogeography-based optimisation," *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 10, pp. 1034–1047, Oct. 2018.

[15] Y. Que, W. Zhong, H. Chen, X. Chen, and X. Ji, "Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 9–12, pp. 4455–4465, Mar. 2018.

[16] M. H. Yuan, Z. Zhou, X. X. Cai, C. Sun, and W. B. Gu, "Service composition model and method in cloud manufacturing," *Robot. Cim.-Int. Manuf.*, vol. 61, Feb. 2020, Art. no. 101840.

[17] L. Zhang, Y. L. Luo, F. Tao, L. Ren, and H. Guo, "Key technologies for the construction of manufacturing cloud," *Comput. Integr. Manuf. Syst.*, vol. 16, no. 11, pp. 2510–2520, Nov. 2010.

[18] Y. Zhang, D. Xi, R. Li, and S. Sun, "Task-driven manufacturing cloud service proactive discovery and optimal configuration method," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 1–4, pp. 29–45, Apr. 2016.

[19] L. B. Zheng, J. N. Gu, and Y. R. Dai, "Modeling of Manufacturing Resources Based on Ontology," *Mach. Des. Res.*, vol. 25, no. 5, pp. 61–63, Oct. 2009.

[20] L. N. Zhu, "Research on Resource Modeling and Selection in Cloud Manufacturing," Ph.D. dissertation, Dept. Mech. Eng., Zhejiang Univ. Technol., Hangzhou, Zhejiang, China, 2014.

[21] L. J. Kong, "Research on servitization method of design resources in cloud manufacturing environment," Ph.D. dissertation, School Mech. Electron. Control. Eng., Beijing Jiaotong Univ., Beijing, China, 2013.

[22] X. Vincent Wang and X. W. Xu, "An interoperable solution for cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 4, pp. 232–247, Aug. 2013.

[23] L. Guo, "Research on cloud manufacturing service platform key technology for machining," Ph.D. dissertation, Dept. Elect. Eng., Chongqing Univ., Chongqing, China, 2014.

[24] F. Tao, L. Zhang, H. Guo, Y. L. Luo, and L. Ren, "Typical characteristics of cloud manufacturing and several key issues of cloud service composition," *Comput. Integr. Manuf. Syst.*, vol. 17, no. 3, pp. 477–486, 2011.

[25] Z. J. Deng, S. J. Wang, X. F. Zheng, Y. F. Suo, and Z. Yu, "Multi-granularity trust model based on individual experience," *Comput. Sci.*, vol. 37, no. 4, pp. 91–105, Apr. 2010.

[26] A. Sarkar and D. Šormaz, "Ontology model for process level capabilities of manufacturing resources," *Procedia Manuf.*, vol. 39, pp. 1889–1898, 2019.

[27] J. Gao, T. J. Wang, D. Q. Yang, and S. W. Tang, "Ontology-based two-phase semi-automatic Web extracting," *Chin. J. Comput.*, vol. 27, no. 3, pp. 310–318, Mar. 2004.

[28] M. Li, D. Z. Wang, and X. Y. Du, "Dynamic composition of Web services based on domain ontology," *Chin. J. Comput.*, vol. 28, no. 4, pp. 644–650, Apr. 2005.

[29] D. Z. Kang, B. W. Xu, J. J. Lu, P. Wang, and Y. H. Li, "Extracting sub-ontology from multiple ontologies," *Lect. Notes. Comput. Sci.*, vol. 2, pp. 731–740, Oct. 2004.

[30] J. Y. Wei, "Research on key technologies of manufacturing resource sharing system supporting for SMEs based on Web services," Ph.D. dissertation, Dept. Mech. Eng. Electron., Shandong Univ. Sci. Technol., Qingdao, Shandong, 2011.

[31] L. N. Zhu, P. H. Li, G. J. Shen, and Z. Liu, "A novel service composition algorithm for cloud-based manufacturing environment," *IEEE Access*, vol. 8, pp. 39148–39164, 2020.

[32] F. Li, T. W. Liao, and L. Zhang, "Two-level multi-task scheduling in a cloud manufacturing environment," *Robot. Cim.-Int. Manuf.*, vol. 56, pp. 127–139, Oct. 2019.

[33] Y. Li, X. Yao, and M. Liu, "Cloud manufacturing service composition optimization with improved genetic algorithm," *Math. Problems Eng.*, vol. 2019, Jul. 2019, Art. no. 7194258.

[34] X. B. Cao, C. D. Xu, C. S. Hu, and Z. Y. Liu, "Design resources agglomeration methods based on design ability," *Comput. Integr. Manuf. Syst.*, vol. 21, no. 9, pp. 2296–2311, Sep. 2015.

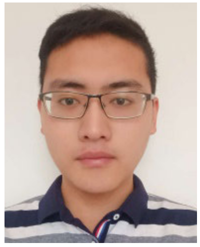
[35] H. M. Guo and S. L. Ma, "Functional-driven resource aggregation: Method and its implementing," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 34, no. 5, pp. 276–579, May 2008.

[36] Y. C. Gao, Y. X. Feng, J. R. Tan, H. Zheng, and Z. Wei, "Composition method of manufacturing service resources based on mutli-domain recursion for multi-disciplinary design," *Int. J. Comput. Integr. Manuf. Syst.*, vol. 18, no. 7, pp. 1406–1414, 2012.

[37] L. Zhu, Y. Zhao, and W. Wang, "A bilayer resource model for cloud manufacturing services," *Math. Problems Eng.*, vol. 2013, Mar. 2013, Art. no. 607582.
[38] N. Liu, X. Li, and W. Shen, "Multi-granularity resource virtualization and sharing strategies in cloud manufacturing," *J. Netw. Comput. Appl.*, vol. 46, pp. 72–82, Nov. 2014.
[39] L. Wang, S. Guo, X. Li, B. Du, and W. Xu, "Distributed manufacturing resource selection strategy in cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 9–12, pp. 3375–3388, Feb. 2018.
[40] C. B. Li and S. H. Hu, "The analysis and design of integrated product team," *Ind. Eng. Manag.*, vol. 7, no. 6, pp. 37–41, Jun. 2002.
[41] S. E. Kesen and Z. Güngör, "Job scheduling in virtual manufacturing cells with lot-streaming strategy: A new mathematical model formulation and a genetic algorithm approach," *J. Oper. Res. Soc.*, vol. 63, no. 5, pp. 683–695, May 2012.
[42] X. B. Cao, C. D. Xu, and C. S. Hu, "Virtual manufacturing unit in cloud manufacturing," *Comput. Integr. Manuf. Syst.*, vol. 18, no. 7, pp. 1415–1425, Jul. 2012.
[43] F. J. Liang and R. X. Ning, "Method of virtual manufacturing cell formation based on similarity science in reconfigurable manufacturing system," *Comput. Integr. Manuf. Syst.*, vol. 10, no. 11, pp. 1370–1376, Nov. 2004.
[44] X. L. Yang and Q. Li, "Applied research of cross-enterprise manufacturing unit build strategy based on multi-objective multi-constraint," *Manuf. Autom.*, vol. 38, no. 11, pp. 99–103, Nov. 2016.
[45] V. E. Fortineau, J. C. Huet, and S. Lamouri, "Ontology for cloud manufacturing based product lifecycle management," *J. Intell. Manuf.*, vol. 30, pp. 2171–2192, Feb. 2019.
[46] X. Li, P. Zhuang, and C. Yin, "A metadata based manufacturing resource ontology modeling in cloud manufacturing systems," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 3, pp. 1039–1047, Mar. 2019.

SHUHUI DING was born in Shandong, China, in 1977. He received the Ph.D. degree in mechanical engineering from the Shandong University of Science and Technology, in 2017.

Since 2012, he has been working as an Associate Professor with the Mechanical Manufacturing Department, Shandong University of Science and Technology. Since 2020, he has also been a Visiting Scholar with the Department of Mechanical Engineering, University of California at Berkeley. He is the author of eight books, five inventions, and more than ten articles. His research interests include cloud manufacturing, digital design, mechanical CAD, intelligent manufacturing, and industrial robot integrated systems.



JINGLIANG HAN was born in Shandong, China, in 1996. He is currently pursuing the master's degree in mechanical engineering with the College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, China.

His research interests include cloud manufacturing system development and digital design.

XIAOJUN MENG was born in Hebei, China, in 1972. He received the M.S. degree in mechanical and electronic engineering from the Shandong University of Science and Technology, in 2002, and the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, in 2008.

Since 2020, he has been a Professor with the Mechanical Engineering Department, Taishan University, Shandong, China. He is the author of one book and more than 20 articles. His research interests include computer integrated manufacturing, cloud manufacturing, computer aided design, and product data management.

FAI MA received the B.S. degree from the University of Hong Kong, in 1977, and the Ph.D. degree from the California Institute of Technology, in 1981. He was formerly a Research Engineer with the IBM Thomas J. Watson Research Center and the Standard Oil Company. For more than 30 years, he has been involved in research in the areas of vibration, nonlinear damping, and system uncertainties. He is currently a Professor of applied mechanics with the Department of Mechanical Engineering, University of California at Berkeley. He is the author or coauthor of more than 190 technical publications. He is a Fellow of the American Society of Mechanical Engineers.