Int J Adv Manuf Technol (2009) 43:379–399
DOI 10.1007/s00170-008-1699-7

ORIGINAL ARTICLE

Study on resource service match and search in manufacturing grid system

Fei Tao · Yefa Hu · Dongming Zhao · Zude Zhou

Received: 21 April 2008 / Accepted: 5 August 2008 / Published online: 2 September 2008

Abstract Resource service match and search (RSMS) is the core to realize manufacturing grid (MGrid) resource scheduling. In order to realize effectively RSMS between resource demanders and providers, a RSMS framework is proposed and the key technologies to realize it are studied. The describing information of resource services are classified into four categories: (a) word concept information, (b) sentence information, (c) number information, including number interval and fuzzy number, and (d) entity class (or data structure) information. The similarity matching algorithms of each kind of describing information are investigated, respectively, including word matching algorithms, sentence matching algorithms, number matching algorithms, and entity class matching algorithms. Based on the proposed matching algorithms, the match and search processes of MGrid resource services are divided into four phases: *first*, matching the basic information of resource services, such as service name and service description, namely, basic matching; *second*, matching the inputs and outputs information of resource services, namely, I/O matching; *third*, matching the quality of service (QoS) information of resource services, namely QoS matching; *last*, combining the above three matching results and generating an integrated matching result, namely, integrated matching. The matching functions and algorithms of each phase are described in detail. A case study illustrates the application of proposed methods, and the accuracy and efficiency of the proposed method are measured.

Keywords Manufacturing grid · Resource service match · Resource service search · Word match · Sentence match · Number match · Entity class match

1 Introduction

Manufacturing grid (MGrid) has been widely accepted, researched, and paid more and more attention to by researchers, scholars, organization, etc. from all over the world [1, 2]. Existing works on MGrid primarily concentrate on its concept, architecture, application prototype system, and application foreground [3]. The application fields of MGrid involve virtual manufacturing, die and mold industry, aeronautical manufacturing, modern logistic, rapid manufacturing, equipments support, engineering simulation, etc [3]. The concept and connotation, including MGrid architecture, key technologies, research contents, technical driving forces, related works of MGrid, quality of service (QoS) modeling, and evaluation for MGrid resource services, have been detailedly described in the authors' previous works [3–5].

In MGrid system, there are primarily two kinds of users [3]: (a) resource enterprise or resource service provider (RSP) and (b) user enterprise or resource service demander (RSD), as shown in Fig. 1. The former, RSP, publishes its idle resource, product, manufacturing ability, etc. and provides manufacturing resource service to meet user's

F. Tao (✉)
School of Mechanical and Electronic Engineering,
Wuhan University of Technology,
Wuhan 430070, China
e-mail: taofei@whut.edu.cn

F. Tao · Y. Hu · Z. Zhou
Hubei Digital Manufacturing Key Laboratory,
Wuhan University of Technology,
Wuhan 430070, China

F. Tao · D. Zhao
Department of ECE, University of Michigan,
Dearborn, MI 48128-1491, USA

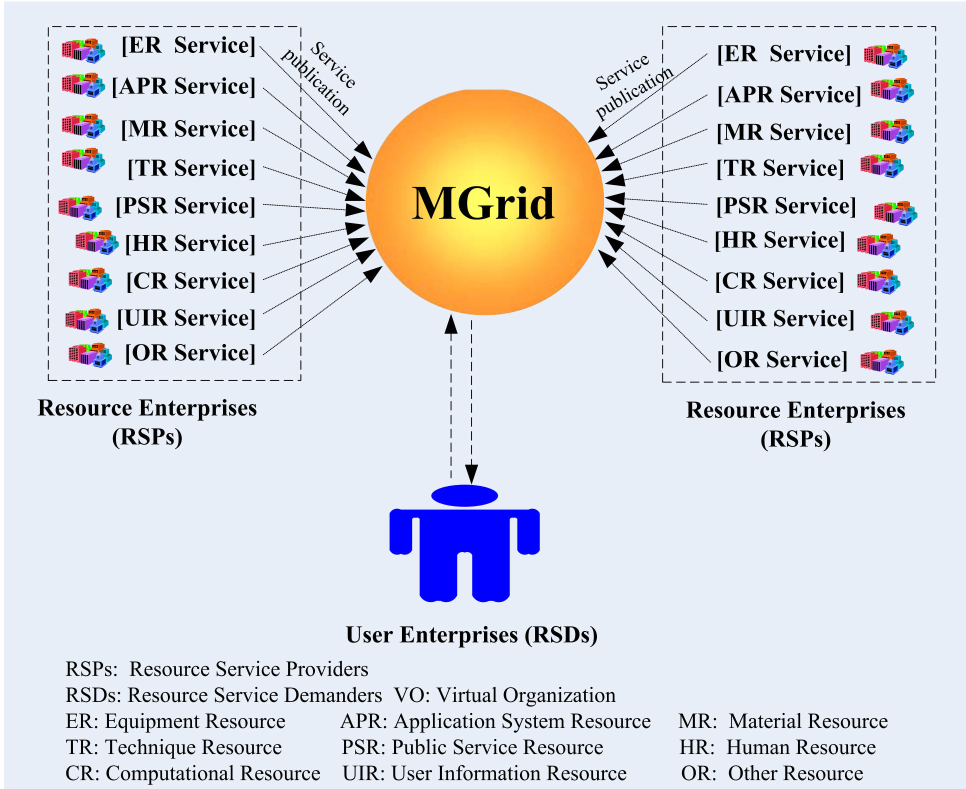


Fig. 1 The relationship of RSD, RSP, and MGrid

requirements. The latter, RSD, searches the optimal manufacturing resource and service required and selects the corresponding partner to establish a collaboration manufacturing net.

One of the key technologies to realize resource service exchange in MGrid is resource service match and search (RSMS). Many works on service search and scheduling are carried out in distributed system such as computing grid system. Some methods such as QoS-based service searching and scheduling, min-min scheduling, and genetic algorithms (GA)-based scheduling have been proposed and studied. But compared with computing grid system, MGrid resource services have the characteristics such as more flexible interaction, real time, long life cycle, multiparty cooperation, resources multiplicity, knowledge, functional complexity, data complexity, higher reliability requirements, online/offline of resources sharing and task submitting. Therefore, RSMS in MGrid cannot simply employ the methods used in computing grid.

Therefore, according to the special characteristics and requirements of MGrid, RSMS in MGrid is studied in this work. First, a RSMS framework is proposed, as well as its working flow. Second, the digital describing information of resource services are classified into four categories: (a) word concept information, (b) sentence information, (c) number information, including number interval and fuzzy number, and (d) entity class (or data structure) information. Then, the similarity matching algorithms (SMAs) of each kind describing information are investigated, including (a) word matching algorithms (WMAs), (b) sentence matching algorithms (SeMAs), (c) number matching algorithms (NMAs), and (d) entity class matching algorithms (ECMAs). A four-phase resource service match and search method is proposed based on the above proposed SMAs, including basic matching, I/O matching, QoS matching, and integrated matching.

The remainder of this paper is organized as follows: Section 2 investigates the related works, and the difference between this work and others are proposed. RSMS framework and its work flows are presented in Section 3. SMAs of four kinds of describing information about resource services are researched in Sections 4. A four-phase RSMS method is described in Section 5. A case study is given out in Section 6. The performance results of the proposed method are discussed in Section 7. Section 8 concludes the whole paper.

2 Related works

2.1 Service discovery in traditional distributed system

A variety of papers dealing with service discovery have recently been published. The most existing service discovery techniques aim at web service discovery simply based on inputs and outputs [6–8] of service. Some of them take into account preconditions, effects, and other parameters that describe services. In addition, some researchers investigate service discovery from the perspectives of behavior signature [9], service community [10], context of service [11–14], user experience [15], role-based interaction model [16], met interaction [17], market-oriented [18], UML [19], goal-based web service discovery with sophisticated semantic matchmaking [20], etc.

The above research works primarily concentrated on abstract service discovery mechanisms and methods. Although these papers proposed efficient and fast searching schemes for service discovery, their schemes do not consider practical SMAs between basic describing information of services. Without the supports of SMAs for basic describing information of services, the upper service discovery cannot be realized. Furthermore, the proposed mechanisms and methods were concerned on the computing resource services in distributed system such as computing grid system. Compared with traditional distributed systems, services in MGrid are more complicated and various and have the characteristics such as more flexible interaction, long life cycle, knowledge, and stronger reliability request. Hence, the existing service discovery methods cannot be simply and directly used in MGrid system. Therefore, in addition to proposing a match and search framework for MGrid resource services, this paper emphasizes on the matching algorithms of basic describing information of resource services.

2.2 Service match and discovery in distributed manufacturing system

Bianchini et al. [21] investigated ontology-based architecture for service discovery system. They classified services into concrete services, abstract services, and subject categories and studied five kinds of service matches, including exact match, plug-in match, subsume match, intersection match, and mismatch. Liu et al. [22] researched MGrid resource scheduling based on QoS and classified the process of resource scheduling into QoS-based resource search, QoS-based resource scheduling, dynamic negotiation, etc. Deng et al [23] and Chen et al. [24] proposed a resource characteristic-based scheduling strategy and applied it into the development of customized artificial joint. Zhang et al. [25] proposed a multi-objective optimization mathematical model. Based on the model and combined with user weights, they designed a GA-based MGrid resource scheduling process. Lv et al. [26] researched a marked equilibrium based MGrid resource optimal allocation methods. The scheduling methods they researched are based on the precondition that the advertising resource service have been searched from the database. Tan et al. [27] studied semantic-based service match and composition under networked manufacturing environment. Zhang et al. [28, 29] researched an ontology-based approach of automated service chaining for MGrid. In order to supplement the shortage that service discovery middle agents (e.g., matchmaker, broker, yellow page, blackboard, etc.) that do not guarantee efficient and rapid matching results, Lee et al. [30] proposed a new matching algorithm based on marriage matching algorithm of ATM network to improve middle agents' performance.

However, the describing information of services considered in their matching methods primarily is limited to ontology concept, inputs, and outputs. In fact, in addition to these, there are number describing information, sentence describing information, and data structure describing information which play a very important role in MGrid resource service description. Furthermore, the specific matching and searching algorithms are not given out in above researches. In order to supplement the above shortages and realize effective RSMS in MGrid system, SMAs for basic describing information of resource services are investigated in this works. A four-phase method for resource service match and search based on the proposed SMAs is presented.

3 Framework of match and search for MGrid resource service

Definition 1 Resource service matching type *Let A be a requested resource service (or a task requirements) and B an advertising resource service in a resource service information center (RSIC), Af_i be an arbitrary describing parameter of A, and Bf_j is the corresponding describing parameter of B. The matching types between A and B are defined as follows*:

(a) If $\forall i, \exists j, Af_i \equiv Bf_j$, then the match type between *A* and *B* are defined as *exact match*;
(b) If $\forall i, \exists j$, $Subs(Bf_j, Af_i)$ (i.e., Bf_j is a super-concept of Af_i), then the match type between *A* and *B* are defined as *plug-in match*. In this condition, *B* can satisfy *A*'s requirements and can be selected to execute the task;
(c) If $\forall i, \exists j$, $Subs(Af_i, Bf_j)$ (i.e., Bf_j is a sub-concept of Af_i), then the match type between *A* and *B* are defined as *subsume match*. In this condition, *B* can satisfy *A*'s requirements partly;
(d) Otherwise, *A* and *B* has no similarity, and *B* cannot satisfy *A*'s requirements at all, namely *mismatch*.

In fact, most advertising resource in RSIC cannot accord to the request of a task or a query 100%, but they are qualified to execute the task. If system only selects the *exact match* resource services as the candidates, as a result, there may be no match result or some qualified resource services are excluded. Consequently, the match and search accuracy is reduced. In order to overcome the above shortage, a relaxation matching strategy is used in the process of RSMS in MGrid. When the similarity between two compared resource services (i.e., advertising resource service and requested resource service) equate or exceed a scheduled threshold value, then the advertising resource service is considered as a qualified candidate. The proposed framework of RSMS is shown in Fig. 2. The brief working flows are described as follows:

(a) A user or RSD submits its request (i.e., a manufacturing task or a resource service request) to the MGrid task management system (MGTMS) via corresponding human–machine interface of MGrid.

(b) MGTMS decomposes the task into corresponding several subtasks that cannot be decomposed again and submits them to system parser.

(c) The system parser transfers the decomposed tasks' requirements into standard resource service describing information, including general information, inputs information, outputs information, QoS information, etc.

(d–e) The system selects the corresponding resource services information from RSIC, and the system parser transfers them into corresponding general information, inputs information, outputs information, QoS information, etc.

(f) The resource service matcher (RS-Matcher) matches the requested resource service information with each advertising resource service extracted from RSIC by invoking the component SMAs.

(g–h) The indices of the qualified resource services are recorded in the candidate resource service set. Then, resource service optimal selection and composition selects the optimal resource service.

In the proposed RSMS framework, the key components are SMAs and RS-Matcher. SMAs provides RS-Matcher with similarity matching algorithms of basic describing information about resource service such as WMAs, SeMAs, NMAs, and ECMAs. RS-Matcher is responsible for matching resource services and task, including basic matching, I/O matching, QoS matching, and integrated matching. The following sections design and describe the detailed working flow and algorithms of SMAs and RS-Matcher, respectively.

4 SMAs: similarity matching algorithms

Digital describing of resource service (DDORS) is the foundation to realize MGrid. It provides data and information supports for the operations involved in the implementation of

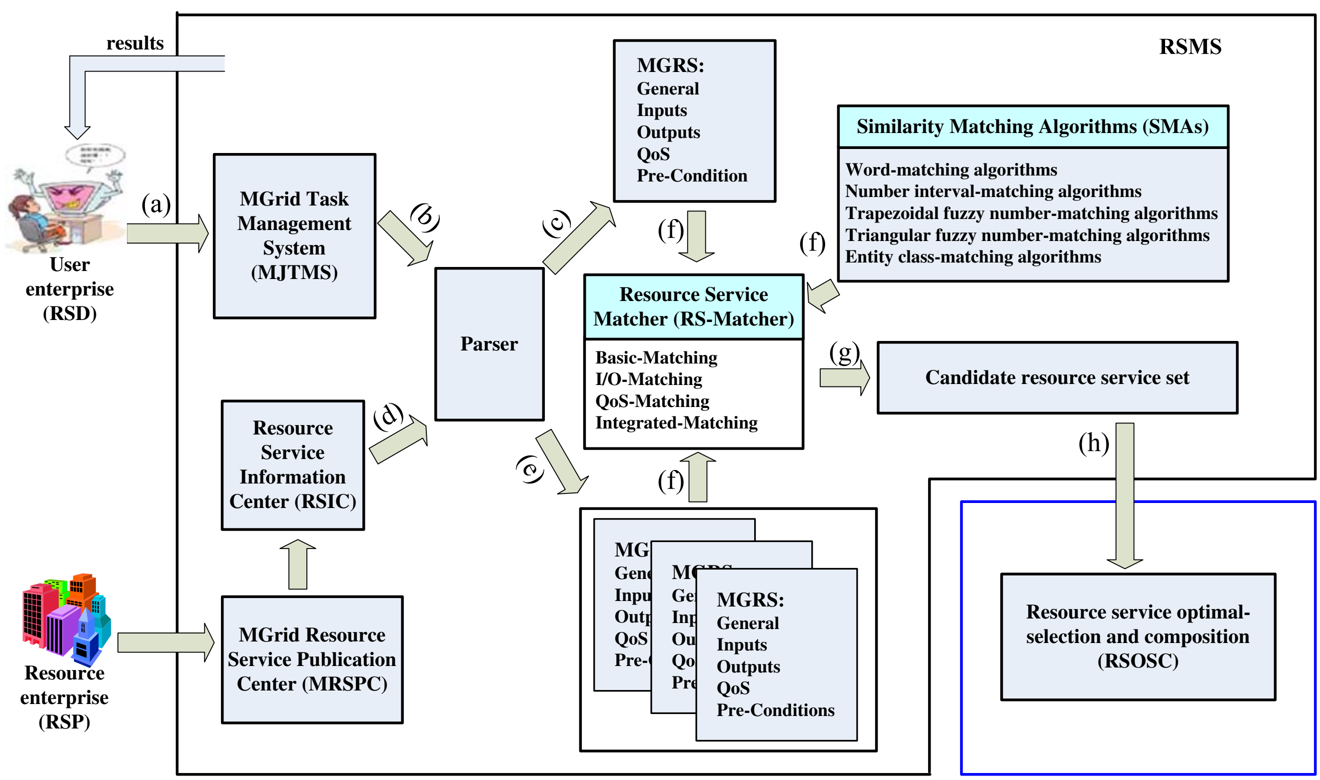


Fig. 2 Framework of match and search for MGrid resource service and its working flow chart

MGrid, including match and search of resource service, QoS modeling and evaluation of resource service, resource service optimal selection and composition, failure detection and recovery, etc. In the authors' research, digital describing of resource service is realized based on OWL-S and MGrid ontology. Each MGrid resource service (MGRS) is described as a five-tuple based on OWL-S and MGrid ontology as follows:

Definition 2: Digital description of resource service
MGRS = (Agent, MGSP, MGSM, MGSG, Res)

where:

- *Agent* is the resource service entity that can solve a manufacturing problem or a RSP. It has self-control ability to its owned resources and their states. The primary describing attributes of *agent* includes *agentName*, *agentAbility*, *agentAim*, *agentType*, etc.
- *MGSP* describes the function and service contents of each resource service. *MGSP* = {*general*, *functional*, *QoS*, *mfgCapability*, *connection*, *state*, *mfgResource*,...}.

  ➢*General* denotes the overall information, e.g., service name, brief service description, create time, and creator, etc. *General* = {*serviceName*, *serviceDescription*, *contactInformation*, *serviceCategory*, *CreateTime*, *Creator*,...}.

  ➢*Functional* is the functional representations of the service. It describes the functional information transforming (i.e., inputs and outputs) and the state changing (i.e., precondition and effect) of service execution. *Functional* = {*Inputs, Outputs, Precondition, Effect*}. *Inputs* denotes the input datum set required when the corresponding resource service is used, e.g., structural parameter and 3D model, etc. *Outputs* denotes the output datum set that the corresponding employed resource service can provide, e.g., a 3D model, an engineering drawing, a simulation results, and so on. *Precondition* denotes the action restriction of the resource service or the precondition of a resource service's executing. For example, pay cost fist before using a resource service. *Effect* denotes the service effect of a resource service.

  ➢*QoS* is the representations of QoS information for resource service. It describes no-functional (i.e., QoS) information of a resource service, including both performance QoS (such as time, reliability, maintainability, satisfaction, etc) and description QoS (such as trust-QoS, cost, etc). It provides data and information support when evaluating the quality of a service. QoS is an extensible vector and *QoS* = {*time*, *cost*, *trust-QoS*, *reliability*, *maintainability*, *functional similarity*, …}. The detailed QoS model associated with its evaluation method can be found in the authors' previous work [4, 5].

  ➢*mfgCapability* describes the manufacturing capability of a resource service, including the manufacturing capability a resource can provide and the requirements of an MGrid task.

  ➢*State* describes the current working or usability state of a resource service. *State* = {*idle*, *underMaintain*, *Load*, *fullLoad*, overload,…}.

  ➢*mfgResource* define the specific information of related nine kinds of resource service. They only fit the very kind resource service belonging to the same classification.
- *MGSM* describes the implementing process of a resource service.
- *MGSG* describes the way another entity (such as soft agent, resource service demander, and provider, etc.) access to a resource service. In MGrid, *MGSG* employs the WSDL and SOAP, etc. as the resource service accessing interface protocol.
- *Res* denotes the resources owned by the entity.

The establishing method for MGrid ontology and detailed method for DDORS are researched in the authors' another paper. In order to simplify the process of RSMS and enhance the efficiency, the describing information of MGrid resource service is classified into the following four categories:

- *Word* concept information, such as resource service name, resource service description, etc.
- *Sentence* information, such as resource service description, etc.
- *Number* information, including *fuzzy number* and *interval number*, e.g., cost of service, manufacturing precision, etc.
- *Entity class* (or *data structure*) information.

A different kind of describing information has different characters and should employ different matching functions when calculating the similarity between them. In the following sections, the SMAs of each kind describing information are investigated, respectively, including WMAs, SeMAs, NMAs, and EMAs.

4.1 Word matching algorithms

Provide that Aw_i is a word concept describing information of a requested resource service *A*, and Bw_j is that of an advertising resource service *B*. The matching degree of the corresponding word concepts describing information between *A* and *B* is defined as the similarity between Aw_i and Bw_j.

Many researchers have investigated the similarity measurement between word concepts in virtual system. Resnik

[31] proposed a method for measuring semantic similarity based on information theory. Jiang and Conrath [32] presented a corpus statistics and lexical-taxonomy-based approach for measuring word semantic similarity. But research on similarity measurement for word concept in manufacturing field is insufficient. Lin et al. [33] studied a measurement method for the process knowledge service ontology (PKSO) concepts. They employed the function $S_c(C_1, C_2)=1/(1+p)$ to evaluate the similarity of two different concepts, C_1 and C_2, in PKSO, where p denotes the shortest path length between C_1 and C_2. However, this method is imprecise if it is applied to calculate the similarity of two manufacturing concepts. For example, in Fig. 3, the shortest path length from *optical-shaft* to *step-shaft* and from *optical-shaft* to *straight-shaft* both are 2. According to the above method, the two pairs of concepts have the same similarity value 1/3, which is unpractical because they only considered the path length between two compared concepts, but the depth, overlap, and density of concepts are ignored. Hence, the calculating result is inauthentic.

In order to address the deficiency, more information should be considered to adjust the above method. In fact, the more of the same words Aw_i and Bw_j contain or share, the stronger is similarity between them. Therefore, the *overlap* of the pair of compared concepts in the hierarchy of ontology should be taken into account while calculating their similarity. Furthermore, concepts at upper layers of the hierarchy have more general semantics and weak similarity between them, while concepts at lower layers have more

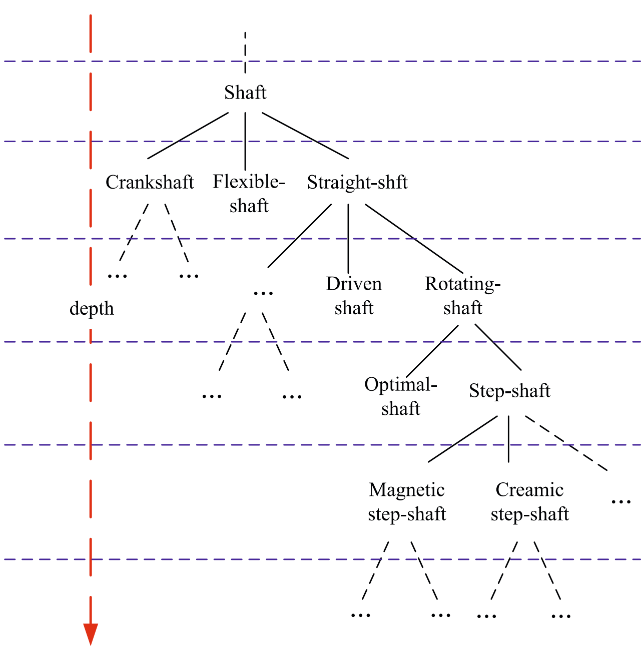


Fig. 3 Hierarchical relationship of shaft ontology. "…" indicates that some words were omitted to save space

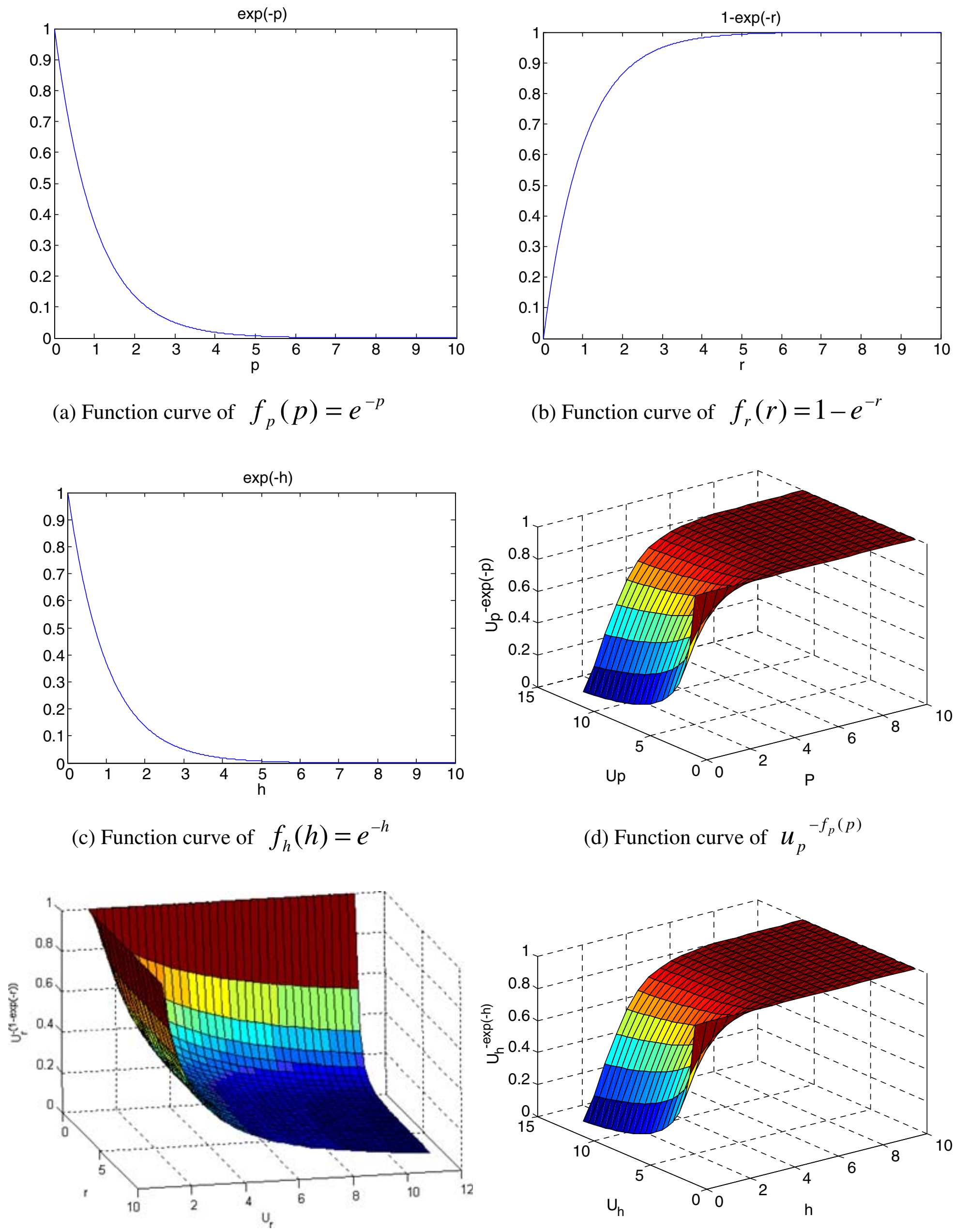
concrete semantics and stronger similarity. Therefore, the *depth* of concepts in the hierarchy is also a decisive fact to similarity. Hence, the similarity between a pair of ontology concepts is decided not only by *path length* but also by *depth* and *overlap* [34].

(a) **Path length (p) of two compared word concepts**. The *path length* of two compared concepts is the number of the border involved in the shortest path connecting the compared words, e.g., in the ISA hierarchy figure of shaft (see Fig. 3), the number of the border involved in the shortest path between *shaft* and *driven-shaft* is 2, so the path length between them is 2. The path length is an important factor that affects the similarity between two compared words. In general, the bigger the path length is, the weaker is the similarity. Contrarily, the smaller the path length is, the stronger is the similarity. Therefore, a monotonically decreasing transfer function, $f_p(p)$, is designed with respect to path length, p, and $f_p(p)=e^{-p}(0\leq p)$ [34]. The function curve of $f_p(p)$ is shown in Fig. 4a.

(b) **Overlap (r) of two compared words**. The *overlap* between two compared word concepts is the number of the same upper concepts they share. The bigger the overlap is, the stronger is the similarity between them, e.g., in Fig. 3, word concepts *rotating-shaft* and *optical-shaft* have three same upper concepts in the ISA hierarchy figures, so their overlap is 3. While the overlap between word concepts *rotating-shaft* and *driven-shaft* is only 2; therefore, the similarity between *rotating-shaft* and *optical-shaft* is stronger than that between *rotating-shaft* and *driven-shaft.* Let $f_r(r)$ denote the transfer function of overlap, r. Then, $f_r(r)$ is a monotonically increasing function with respect to r, and $f_r(r)=1-e^{-r}$. The function curve of $f_r(r)$ is shown in Fig. 4b.

(c) ***Depth* of two compared words (h)**. The *depth* of each compared word concept is derived by counting the levels from it to the top of the lexical hierarchy, e.g., in Fig. 3, the depth of *driven-shaft* is 2. When the compared words have the same path length, the similarity between them increases as the sum of their level numbers increases or decreases as the difference of their level numbers decreases, e.g., in Fig. 3, *optical-axis* and *step-shaft*, *rotating-shaft* and *driven-shaft* have the same path length, but the sum of the level numbers of the former is bigger than the later, so similarity of the former is bigger than of the later. Let h_i and h_j be the depth of two compared word concepts, Aw_i and Bw_j, respectively, and $h=|h_i-h_j|$. Let $f_h(h)$ denote the transfer function of depth, $f_h(h)$ is a monotonically decreasing function with respect to

Fig. 4 Function curves of word concept similarity



(a) Function curve of $f_p(p) = e^{-p}$

(b) Function curve of $f_r(r) = 1 - e^{-r}$

(c) Function curve of $f_h(h) = e^{-h}$

(d) Function curve of $u_p^{-f_p(p)}$

(e) Function curve of $u_r^{-f_r(r)}$

(f) Function curve of $u_h^{-f_h(h)}$

depth h, and $f_h(h)=e^{-h}=e^{-|hi-hj|}$. The function curve of $f_h(h)$ is shown in Fig. 4c.

Therefore, the similarity, $\mathrm{Match_w}(Aw_i, Bw_j)$, between Aw_i and Bw_j is a function of the attributes p, r, and h, as follows:

$$\begin{aligned}\mathrm{Match_w}\left(Aw_i, Bw_j\right) &= f(p,r,h) \\ &= u_\mathrm{p}^{-f_\mathrm{p}(p)} \times u_\mathrm{r}^{-f_\mathrm{r}(r)} \times u_\mathrm{h}^{-f_\mathrm{h}(h)}\end{aligned} \quad (1)$$

where:

$$\begin{cases} f_\mathrm{p}(p) = e^{-p} & (0 \le p) \\ f_\mathrm{r}(r) = 0 & (r = 0) \\ f_\mathrm{r}(r) = 1 - e^{-r} & (0 < r) \\ f_\mathrm{h}(h) = 1 & \left(h = \left|h_i - h_j\right| = 0\right) \\ f_\mathrm{h}(h) = e^{-h} = e^{-\left|h_i - h_j\right|} & \left(0 \le h = \left|h_i - h_j\right|\right) \\ \left(u_\mathrm{p}, u_\mathrm{r}, u_\mathrm{h} \in [1, \infty]\right) & \end{cases} \quad (2)$$

u_p, u_r, u_h are the scaling factors of $f_\mathrm{p}(p)$, $f_\gamma(r)$, and $f_\mathrm{h}(h)$, respectively. They are the decisive factors to the speed approaching to the 100% similarity 1. At the same time, u_p, u_r, u_h can be seen as the weights of $f_p(p)$, $f_r(r)$, and $f_h(h)$, respectively, occupy in the whole similarity between Aw_i and Bw_j, or the respective influencing degree to the whole similarity, e.g., the bigger the u_p is, the faster the value of $f_p(p)$ approaching to 1 (as shown in Fig. 4d) and the bigger the weight path length, p, occupies in the whole similarity. The same applies to $f_h(h)$ and *depth*, h, (as shown in Fig. 4f). Contrarily, the bigger the u_r, the slower the value of $f_r(r)$ approaching to 1 (as shown in Fig. 4e) and the smaller the weight overlap occupies in the whole similarity.

4.2 Sentence matching algorithms

Provided that At_i is a sentence describing information of a requested resource service and Bt_j is that of an advertising resource service. The similarity between At_i and Bt_j is decided by *key words*, *sentence length*, and *word order*. Set T_i and T_j are the words set of sentence At_i and Bt_j, respectively, and $W^{\cup} = T_i \cup T_j = \{w_1^{\circ}, w_2^{\circ}, \cdots, w_m^{\circ}\}(m = 1,\ 2,\ 3, \ldots)$ denotes the union set of T_i and T_j, $W^{\cap} = T_i \cap T_j \{w_1{}^*, w_2{}^*, \cdots, w_n{}^*\}(n = 1,\ 2,\ 3, \ldots)$ denotes the intersection set of T_i and T_j. Obviously, $m \geq n$. The respective similarity measuring method of key words, sentence length, and word order are described as follows.

(a) ***Key words* similarity**: Let $s_{i,k}$ and $s_{j,k}(0 \leq s_{i,k}, s_{j,k} \leq 1)$ denote the strongest similarity of each word $w_k^{\circ}(0 < k \leq m)$ in $W^{\cup}$ within the corresponding words in T_i and T_j, respectively. If w_k° appears in the sentence T_i (or T_j), $s_{i,k}$(or $s_{j,k}$) is set to 1. Otherwise, if w_k° is not contained in T_i(or T_j), a similarity score is calculated between w_k° and each word in sentence T_i(or T_j) using the method in Section 4.1. Then, $s_{i,k}$(or $s_{j,k}$) is the strongest similarity between the words in sentence T_i(or T_j) and w_k°. Therefore, the key word similarity, $S_1(At_i, Bt_j)$, between At_i and Bt_j can be formulated as [35]:

$$S_1(At_i, Bt_j) = 1 - \left| \sum_{k=1}^{m} s_{i,k} - \sum_{k=1}^{m} s_{j,k} \right| \Big/ \left(\sum_{k=1}^{m} s_{i,k} + \sum_{k=1}^{m} s_{j,k} \right). \tag{3}$$

(b) ***Word order* similarity**: A unique index number is assigned for each word in $W^{\cup}$. The index number is the corresponding order number in $W^{\cup}$. Set $O_{\mathrm{rder}} = \{1,2,3, \ldots, m\}$ denote the order number vector set. Correspondingly, the word order vector OT_i and OT_j are formulated to denote the word order in sentence T_i and T_j, respectively. Let $|OT_i|$ and $|OT_j|$ denote the absolute value of the sum of backward sequence numbers of every two neighboring order numbers in OT_i and OT_j, respectively, e.g., if $OT_i = \{1, 2, 4, 3, 3, 5\}$, then $|OT_i| = |1 + 2 + (-1) + 0 + 2| = 0$. Therefore, the word order similarity between At_i and Bt_j can be formulated as:

$$S_2(At_i, Bt_j) = 1 - \big||OT_i| - |OT_j|\big| \big/ \left(|OT_i| + |OT_j|\right). \tag{4}$$

(c) ***Sentence length* similarity**: *Sentence length* is defined as the number of the total words involved in a sentence. Let l_i and l_j denote the sentence length of At_i and Bt_j, respectively. Then, the sentence length similarity between At_i and Bt_j can be formulated as $S_3(At_i, Bt_j) = 1 - |l_i - l_j|/(l_i + l_j)$. In general, when the *key words* similarity and *word order* similarity are fixed, the bigger is the sentence length similarity value and the stronger is the similarity between two compared sentences.

During the practical sentence similarity evaluation process, *key words* similarity pays the highest role, and *word order* and *sentence length* play secondary function. Therefore, different weights should be assigned to key words similarity, word order similarity, and sentence length similarity. Let α, β, δ $(0 \leq \alpha, \beta, \delta \leq 1, \alpha + \beta + \delta = 1)$ be the corresponding weights, respectively. Therefore, the total similarity of sentences At_i and Bt_j is formulated as:

$$\begin{aligned}\mathrm{Match_T}(At_i, Bt_j) &= a \times S_1(At_i, Bt_j) + \beta \times S_2(At_i, Bt_j) \\ &\quad + \delta \times S_3(At_i, Bt_j) \\ &= \alpha \left(1 - \frac{\left| \sum_{k=1}^{m} s_{i,k} - \sum_{k=1}^{m} s_{j,k} \right|}{\sum_{k=1}^{m} s_{i,k} + \sum_{k=1}^{m} s_{j,k}} \right) + \beta \left(1 - \frac{\big||OT_i| - |OT_j|\big|}{|OT_i| + |OT_j|} \right) \\ &\quad + \delta \left(1 - \frac{|li - lj|}{li + lj} \right).\end{aligned} \tag{5}$$

4.3 Number matching algorithms

When matching two MGrid resource services, some number parameters cannot be avoided. Number describing information of MGrid resource service can be classified into two subclasses: (a) *numerical interval* and (b) *fuzzy number*. The former, *numerical interval*, is used to describe a specific value or value range, such as cost and date of delivery. The later, *fuzzy number*, is used to describe some uncertain parameters, such as fuzzy grades, fuzzy evaluations, etc. The specific similarity measuring method for numerical interval parameters and fuzzy number parameters are different.

1. Numerical interval similarity

The task of numerical interval matching is to calculate the matching degree of numerical interval parameters between requested and advertising resource service, e.g., a

RSD asks for a resource service within a cost during 100 dollars to 200 dollars, while the price of an advertising resource service (RSP) is higher than 120 dollars and lower than 200 dollars. So what is the matching degree between the RSD and RSP? Whether the RSD will accept the RSP to execute its task is decided by the similarity between their costs.

When calculating the similarity of number parameters, number information usually can be transformed into numerical interval, e.g., the aforementioned cost (i.e., during 100 dollars to 200 dollars) asked by the RSD can be denoted as interval [100,200], and the price (i.e., higher than 120 dollars and lower than 200 dollars) of the RSP can be transformed into interval [120,200]. Therefore, the number matching is transformed into numerical interval matching.

Provided that Ad_i is a number parameter describing information of a requested resource service A and Bd_j is that of an advertising resource service B. Let Ad and Bd denote the transformed corresponding numerical interval about Ad_i and Bd_j respectively. The similarity between Ad_i and Bd_j is formulated as:

$$\mathrm{Match}_D\left(Ad_i, Bd_j\right) = \begin{cases} 1 & (Ad \cap Bd = Ad, or, Ad \cap Bd = Bd) \\ \frac{|Ad \cap Bd|}{|Ad|} & (Ad \cap Bd \neq \Phi, and, Ad \cap Bd \neq Ad) \\ 0 & (Ad \cap Bd = \Phi) \end{cases} \quad (6)$$

where '| |'denotes the length of the corresponding numerical interval, e.g., |(10,20)|=10 and |(10,∞)|=∞.

2. Fuzzy numerical similarity

In real manufacturing activities, situations are very often uncertain. Fuzzy numbers and fuzzy expressions are often used to resolve the uncertain problems existing in manufacturing activities, such as linguistic, knowledge expression, control system, database, decision making system, failure diagnosis, etc., e.g., fuzzy evaluations to a certain index during manufacturing activities usually are described as {*bad*, *middle*, *general*, *good*, *excellent*} or {*poor*, *middle*, *high*}. All these linguistic variables or parameters cannot be read and understood by machine and computer, so they are often transformed into fuzzy number, e.g., the fuzzy evaluation {*poor*, *middle*, *high*} can be represented by fuzzy number {0.2, 0.6, 0.8}. In order to enable the description of uncertain factors and parameters to be easily read and understood by machine, in MGrid system, uncertain factor parameters are represented using fuzzy number, including *triangular fuzzy numbers* and *trapezoidal fuzzy numbers*. Hence, when the system searching and matching resource service according to user's requirements, the problem of fuzzy number matching must be addressed. In this work, triangular fuzzy numbers and trapezoidal fuzzy numbers are primarily used to describe the uncertain factors and parameters of resource service information. Therefore, the matching methods and algorithms between fuzzy numbers are investigated in the following sections, including triangular fuzzy numbers matching algorithms (TriFN-MAs) and trapezoidal fuzzy numbers matching algorithms (TraFNMAS).

Before describing the matching algorithms of fuzzy numbers, let us briefly introduce the concept of triangular fuzzy numbers and trapezoidal fuzzy number.

Definition 3: Triangular fuzzy number *Triangular fuzzy number is a fuzzy number represented with three points as follows [36]:*

$$\widetilde{A} = (m, \alpha, \beta)$$

This representation is interpreted as membership functions (Fig. 5).

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0 & , \quad x < m - \alpha \\ (x - m + \alpha)/\alpha & , \quad m - \alpha \le x < m \\ (m + \beta - x)/\beta & , \quad m \le x < m + \beta \\ 0 & , \quad x \ge m + \beta \end{cases} . \quad (7)$$

Definition 4: Trapezoidal fuzzy number *Trapezoidal fuzzy number is defined as [36]:*

$$\widetilde{A} = (m_1, m_2, \alpha, \beta).$$

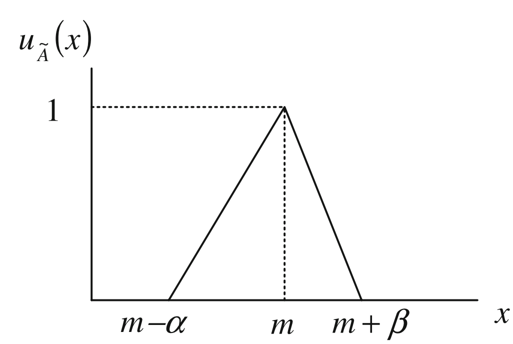


Fig. 5 Triangular fuzzy number $\widetilde{A} = (m, \alpha, \beta)$

The membership function of this fuzzy number is interpreted as follows (Fig. 6).

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0 & , \quad x < m_1 - \alpha \\ (x - m_1 + \alpha)/\alpha & , \quad m_1 - \alpha \le x < m_1 \\ 1 & , \quad m_1 \le x < m_2 \\ (m_2 + \beta - x)/\beta & , \quad m_2 \le x < m_2 + \beta \\ 0 & , \quad x \ge m_2 + \beta \end{cases} . \qquad (8)$$

Set $Ad_{\rm i}$ is a fuzzy number parameter describing information of a requested resource service A and Bd_j is that of an advertising resource service B; then, the similarity between Ad_i and Bd_j is formulated as [37]:

$$\mathrm{Match}_D\left(Ad_i, Bd_j\right) = \begin{cases} 1 & \left(Ad_i = Bd_j\right) \\ \exp\left(-d^2\left(Ad_i, Bd_j\right)/\sigma\right) & \left(Ad_i \neq Bd_j\right) \end{cases} \qquad (9)$$

where σ is a constant bigger than 0, and the corresponding variables in above function are as follows:

(a) If $Ad_{\rm i}$ and $Bd_{\rm j}$ are triangular fuzzy numbers, and Ad_i=(m_i,α_i,β_i), Ad_j=(m_j,α_j,β_j), then

$$\begin{cases} \ell = \int_0^1 \mathrm{L}^{-1}(\mathrm{w})\mathrm{dw} \\ r = \int_0^1 \mathrm{R}^{-1}(\mathrm{w})\mathrm{dw} \\ D_* = \left|\left(m_i - \ell\alpha_i\right) - \left(m_j - \ell\alpha_j\right)\right| \\ D^* = \left|\left(m_i + r\beta_i\right) - \left(m_j + r\beta_j\right)\right| \\ d^2 = \left(m_i - \alpha_j\right)^2 + \left(D_*\right)^2 + \left(D^*\right)^2 \\ \sigma = \left(D_* + D^*\right)\Big/2 + \left(\left|\left(m_i - \alpha_i\right) - \left(m_j - \alpha_j\right)\right| + \left|\left(m_i + \beta_i\right) - \left(m_j + \beta_j\right)\right|\right)/2^3 \end{cases} \qquad (10)$$

(b) If $Ad_{\rm i}$ and $Bd_{\rm j}$ are trapezoidal fuzzy numbers and Ad_i=($m_{1,i}$,$m_{2,i}$,α_i,β_i); Ad_j=($m_{1,j}$,$m_{2,j}$,α_j,β_j), then

$$\begin{cases} \ell = \int_0^1 \mathrm{L}^{-1}(\mathrm{w})\mathrm{dw} \\ r = \int_0^1 \mathrm{R}^{-1}(\mathrm{w})\mathrm{dw} \\ D_* = \left|\left(m_{1,i} - \ell\alpha_i\right) - \left(m_{1,j} - \ell\alpha_j\right)\right| \\ D^* = \left|\left(m_{2,i} + r\beta_{2,i}\right) - \left(m_{2,j} + r\beta_{2,j}\right)\right| \\ d^2 = \left(\left(m_{1,i} - m_{1,j}\right)^2 + \left(m_{2,i} - m_{2,j}\right)^2\right)\Big/2 + \left(D_*\right)^2 + \left(D^*\right)^2 \\ \sigma = \left(D_* + D^*\right)\Big/2 + \left(\left|\left(m_{1,i} - \alpha_i\right) - \left(m_{1,j} - \alpha_j\right)\right| + \left|\left(m_{2,i} + \beta_i\right) - \left(m_{2,j} + \beta_j\right)\right|\right)/2^4 \end{cases} . \qquad (11)$$

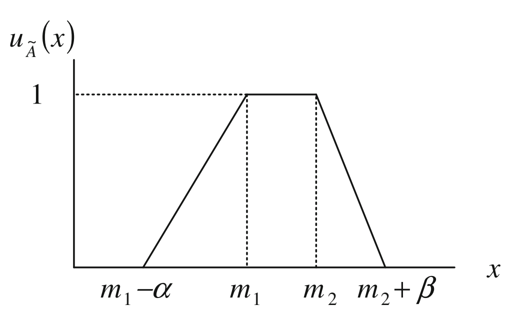


Fig. 6 Trapezoidal fuzzy number $\widetilde{A} = (m_1, m_2, \alpha, \beta)$

4.4 Entity class matching algorithms

Set Ac_i is an entity class parameter describing information of a requested resource service A and Bc_j is that of an advertising resource service B. Let Ac and Bc are the corresponding description sets (i.e., synonym sets, set of distinguishing features, and sets of the entity class in the semantic neighborhood) of Ac_i and Bc_j, respectively. The similarity between Ac_i and Bc_j is calculated according to

Tversky's model [38] and Andrea' similarity functions [39] combined with set theory.

$$\mathrm{Match_C}\left(Ac_i, Bc_j\right) = \frac{|Ac \cap Bc|}{|Ac \cap Bc| + \alpha\left(Ac_i, Bc_j\right)|Ac/Bc| + \left(1 - \alpha\left(Ac_i, Bc_j\right)|Bc/Ac|\right)} \tag{12}$$

where

$$\alpha\left(Ac_i, Bc_j\right) = \begin{cases} \frac{\mathrm{depth}(Ac_i)}{\mathrm{depth}(Ac_i)+\mathrm{depth}\left(Bc_j\right)} & \left(\mathrm{depth}(Ac_i) \le \mathrm{depth}\left(Bc_j\right)\right) \\ 1 - \frac{\mathrm{depth}(Ac_i)}{\mathrm{depth}(Ac_i)+\mathrm{depth}\left(Bc_j\right)} & \left(\mathrm{depth}(Ac_i) > \mathrm{depth}\left(Bc_j\right)\right) \end{cases} \tag{13}$$

'∩'is the intersection operation and '/' is the difference operation of set (e.g., $Ac/Bc=Ac-(Ac\cap Bc)$). '| |'is the cardinality of a set. α is a function that defines the relative importance of the non-common characteristics. The function depth() corresponds to the shortest path from the entity class point (e.g., the point of entity class Ac_i) to the upper intersection point of entity Ac_i and Bc_j [39]. If Ac_i and Bc_j are the same concepts in the ontology, then $\alpha(Ac_i, Bc_j)=0.5$.

5 RS-Matcher: Resource service matcher

Definition 5 Resource service matching model Set *A* is a requested resource service and *B* is an advertising resource service, and the matching model between *A* and *B* are defined as follows:

$$\mathrm{Match}(A, B) = \prod\left(\mathrm{Match_{bas}}(A, B), \mathrm{Match_{i/o}}(A, B), \mathrm{Match_{QoS}}(A, B)\right) \tag{14}$$

where:

- $\mathrm{Match_{bas}}(A,B)$ denotes *basic matching*. It is primarily responsible for matching the general information between *A* and *B*, such as *ServiceName* and *ServiceDescription*. The detailed methods and algorithms of *basic matching* are shown in Section 5.1.
- $\mathrm{Match_{i/0}}(A,B)$ is the *I/O matching* (i.e., inputs and outputs matching). It is primarily responsible for matching the inputs and outputs information between *A* and *B*. The detailed methods and algorithms of *I/O matching* are shown in Section 5.2.
- $\mathrm{Match_{QoS}}(A,B)$ is *QoS matching*. It is primarily responsible for matching the QoS information between *A* and *B*, such as time, cost, reliability, trust, maintainability, and satisfaction. The detailed methods and algorithms of *QoS matching* are shown in Section 5.3.
- Π() is a comprehensive processing function. It integrates the matching results of basic matching, I/O matching, and QoS matching and generates an integrated matching result.

The matching process of MGrid resource service is a stepwise precision process, and it primarily has four steps.

- First, matching the basic information, such as *serviceName* and *serviceDescription*, of *A* and *B* according to the basic matching methods and algorithms described in Section 5.1. If the result of basic matching, i.e., $\mathrm{Match_{bas}}(A,B)$, is lower than the basic threshold value, $\zeta_{bas}(0\le\zeta_{bas}\le1)$, which is set by user or system, then it illustrated that the advertising resource service cannot satisfy the general requirements of user. Then, the system stops further matching and returns the failed message to user.
- If basic matching result arrives ζ_{bas} and the user requires further precision matching, then the system carries out I/O matching according to the methods and algorithms described in Section 5.2. If the result of I/O matching, i.e., $\mathrm{Match_{i/0}}(A,B)$, is lower than the I/O threshold value, $\zeta_{i/o}(0\le\zeta_{i/o}\le1)$, which is set by user or system, then it illustrated that the advertising resource service is unqualified for user's I/O requirements. Then, the system stops further matching and returns the failed message to user.
- If the advertising resource service qualified for user's I/O requirements and further precision matching is required, then system carries out QoS matching according to the methods and algorithms described in Section 5.3. If the result of QoS matching, i.e., $\mathrm{Match_{QoS}}(A,B)$, is lower than the QoS threshold value, *$\zeta_{QoS}(0\le\zeta_{QoS}\le1)$, which is set by user or system, then it illustrated that the advertising resource service is*

unqualified for user's QoS requirements. Then, the system stops further matching and returns the failed message to user.

- If the advertising resource service is qualified for all the requirements (i.e., arrived all the threshold values of ζ_{bas}, $\zeta_{i/o}$, and ζ_{QoS}), then the system calculates the final overall matching according to the methods and algorithms described in Section 5.4. If the integrated matching result is equal or over the total threshold value, $\zeta(0 \le \zeta \le 1)$, then it concludes that the advertising resource service satisfied the requirements of user very well and it can be selected to execute the requested task.

The matching process flowchart is shown in Fig. 7. The basic matching, I/O matching, QoS matching can be used separately or together, which is decided by the practical requirements and conditions of user or system.

5.1 Basic matching

Basic matching is primarily responsible for matching the general information between resource services, such as *ServiceName* and *ServiceDescription*. Because *ServiceName* and *ServiceDescription* usually are described with words or sentences, the system primarily employs the proposed WMAs and SeMAs to calculate basic matching as follows:

$$\mathrm{Match_{bas}}(A,B) = \begin{pmatrix} \omega_1 \times \mathrm{Match_w}(A.ServiceName, B.ServiceName) + \\ \omega_2 \times \mathrm{Match_T}(A.ServiceDescription, B.ServiceDescription) \end{pmatrix}. \quad (15)$$

For $0 \le \omega_1, \omega_2 \le 1$, and $\omega_1 + \omega_2 = 1$. ω_1 and ω_2 are the weights of *ServiceName* and *ServiceDescription*, respectively.

Set *A. General* is the general describing information set of requested resource service *A*, and *B. General* is that of advertising resource service *B*. Let ζ_{bas} be the basic matching threshold value declared by *A* (i.e., if an advertising resource service want to execute the task of *A*, the basic matching value between it and *A* must equal or more than ζ_{bas}, otherwise it will be an also-ran). According to expression 15, the pseudo-codes of algorithms for basic matching between *A* and *B* are as follows:

Inputs: *A. General*; *B. Genera*; ζ_{bas}
Outputs: resource service basic-matching results, $\mathrm{Match_{bas}}(A,B)$
1 Calculating $\mathrm{Match_w}$ (*A.ServiceName*, *B. ServiceName*) according to expression 1
2 Calculating $\mathrm{Match_T}$(*A.ServiceDescription*, *B. ServiceDescription*) according to expression 5
3 Calculating $\mathrm{Match_{bas}}(A,B)$ according to expression 15
4 If ($\mathrm{Match_{bas}}(A,B) < \zeta_{bas}$)
5 *B* is unqualified for *A*'s general requirements, return failed matching message
6 End if
7 *B* is qualified for *A*, return successful message and matching result $\mathrm{Match_{bas}}(A,B)$

5.2 I/O matching

As stated above, I/O matching is primarily responsible for matching the input and output information between requested and advertising resource services. It is assumed that the parameter set of outputs and inputs of a resource service is D∪C∪W where D,C,W denote the subset of number parameters, entity classes parameters, word concept parameters, respectively. The matching function of I/O matching is defined as follows:

$$\mathrm{Match_{i/o}}(A,B) = \begin{pmatrix} \sum_{i=1}^{N_d} \omega_{d_i} \mathrm{Match_D}(A.d_i, B.d_i) + \\ \sum_{j=1}^{N_c} \omega_{c_j} \mathrm{Match_C}(A.c_j, B.c_j) + \\ \sum_{k=1}^{N_k} \omega_{w_k} \mathrm{Match_W}(A.w_k, B.w_k) \end{pmatrix} \quad (16)$$

where $d_i(d_i \in D), c_j(c_j \in C), w_k(w_k \in W)$ denotes the $i^{th}(i=1,2,3,\ldots,N_d)$, $j^{th}(j=1,2,3,\ldots,N_c)$, $k^{th}(k=1,2,3,\ldots,N_k)$ parameter in D,C,W, respectively. $\omega_{d_i}, \omega_{c_j}, \omega_{w_k}$ are the corresponding weights of d_i, c_j, w_k, respectively, $0 \le \omega_{d_i}, \omega_{c_j}, \omega_{w_k} \le 1$ and $\sum_{i=1}^{N_d} \omega_{d_i} + \sum_{j=1}^{N_c} \omega_{c_j} + \sum_{k=1}^{N_k} \omega_{w_k} = 1$.

Set *A.Inputs* and *A.Outputs* are the input and output describing information sets of requested resource service *A*. I/O parameter set of *A* can be classified into three different subsets W_A, D_A, C_A (the intersection of any two of W_A, D_A, C_A is empty), which denote the subsets of word concept parameters, number parameters, and entity class parameters, respectively. Accordingly, Let W_B, D_B, C_B (the intersection of any two of W_B, D_B, C_B is empty too) be the subsets of word concept parameters, number parameters, and entity class parameters of *B*'s I/O parameters set. Let $\zeta_{i/o}$ be the I/O matching threshold value declared by *A*. According to

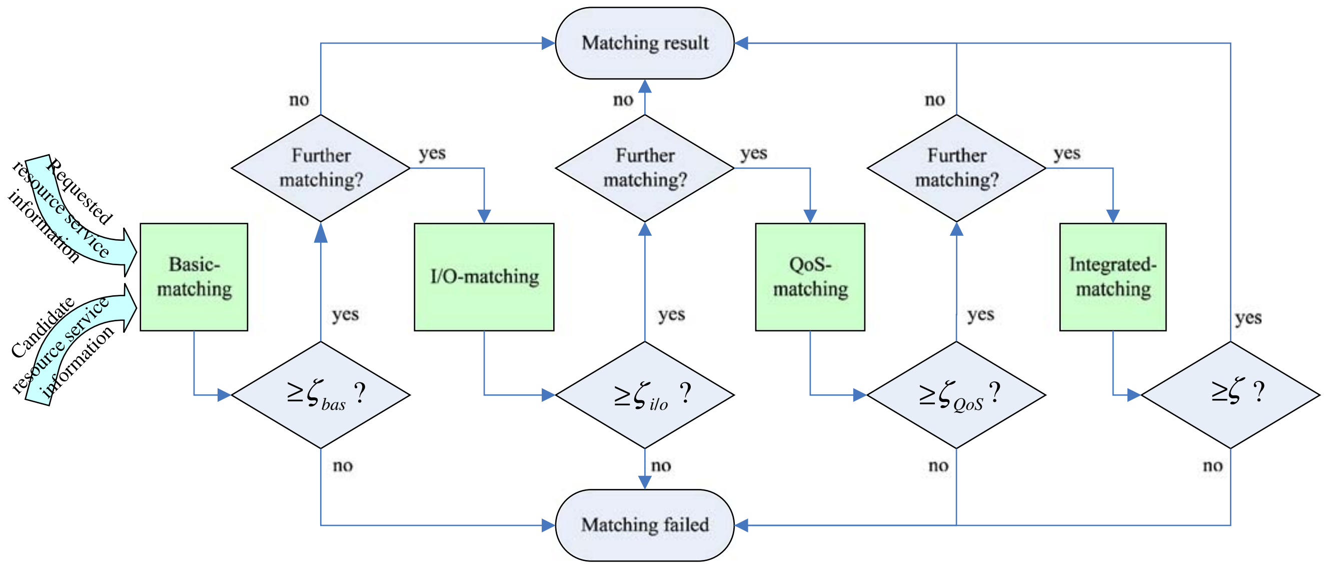


Fig. 7 Four-steps matching process flowchart of MGrid resource service (**a**) Results under condition (I) (**b**) Results under condition (II)

expression 16, the pseudo-codes of I/O matching algorithms between A and B are as follows:

Inputs: W_A,D_A,C_A; W_B,D_B,C_B; $\zeta_{i/o}$
Outputs: resource service I/O matching results $Match_{i/0}(A,B)$
If W_B is not empty
For each $B.w_k$ in W_B and $A.w_k$ in W_A
Calculating $Match_W(A.w_k,B.w_k)$ according to expression 1
End for
End if
If D_B is not empty
For each $B.d_i$ in D_B and corresponding $A.d_i$ in D_A
If $B.d_i$ is a number interval parameter
Calculating $Match_D(A.d_i,B.d_i)$ according to expression 6
End if
If $B.d_i$ is a triangular fuzzy number parameter
Calculating $Match_D(A.d_i,B.d_i)$ according to expressions 9 and 10
End if
If $B.d_i$ is a trapezoidal fuzzy number parameter
Calculating $Match_D(A.d_i,B.d_i)$ according to expressions 9 and 11
End if
End for
End if
If C_B is not empty
For each $B.c_j$ in C_B and corresponding $A.c_i$ in C_A
Calculating $Match_C(A.c_j,B.c_j)$ according to expression 12
End for
End if
Calculating $Match_{i/0}(A,B)$ according to expression 16
If $Match_{i/0}(A,B)<\zeta_{i/o}$
B is unqualified for A's I/O requirements, return failed matching message
End if
B is qualified for A's I/O requirements, return successful message and $Match_{i/0}(A,B)$

5.3 QoS matching

As defined before, QoS matching is primarily responsible for matching the QoS information between resource services, such as time, cost, reliability, trust, maintainability, and satisfaction. Because QoS parameters are main number parameters, the QoS matching primarily depends on NMAs. Therefore, QoS matching can be formulated as follows:

$$\mathrm{Match_{QoS}}(A,B)=\sum_{i=1}^{m}\omega_{\mathrm{QoS}_i}\mathrm{Match_d}(A.\mathrm{QoS}_i,B.\mathrm{QoS}_i). \tag{17}$$

For (i=1,2,3,…,m), $0\le\omega_{\mathrm{QoS}_i}\le1$ and $\sum_{i=1}^{m}\omega_{\mathrm{QoS}_i}=1$, where ω_{QoS_i} is the weight of each corresponding QoS parameter and m is the total number of QoS parameters.

Set $A.QoS$ is the general describing information set of requested resource service A and $B.QoS$ is that of advertising resource service B. Let ζ_{QoS} be the QoS matching threshold value declared by A. According to expression 17, the pseudo-codes of algorithms of QoS matching between A and B are as follows:

Inputs: $A.QoS$, $B.QoS$, ζ_{QoS}
Outputs: resource service I/O matching results $Match_{QoS}(A,B)$
For each QoS parameters $A.QoS_i$ in $A.QoS$ and $B.QoS_i$ in $B.QoS$
If $A.QoS_i$ and $B.QoS_i$ are number interval parameters
Calculating $Match_D(A.d_i,B.d_i)$ according to expression 6
End if
If $A.QoS_i$ and $B.QoS_i$ are trapezoidal fuzzy number parameters
Calculating $Match_D(A.d_i,B.d_i)$ according to expressions 9 and 10
End if

8 If $A.QoS_i$ and $B.QoS_i$ are number interval parameters
9 Calculating $\mathrm{Match}_{\mathrm{D}}(A.d_i,B.d_i)$ according to expressions 9 and 11
10 End if
11 End for
12 Calculating $\mathrm{Match}_{\mathrm{QoS}}(A,B)$ according to expression 17
13 If $\mathrm{Match}_{\mathrm{QoS}}(A,B)<\zeta_{\mathrm{QoS}}$
14 B is unqualified to A's QoS requirements, return matching failed message
15 End if
16 B is qualified for A's QoS requirements, return successful message and $\mathrm{Match}_{\mathrm{QoS}}(A,B)$

5.4 Integrated matching

In practice, there is far more than one advertising resource service for a requested resource service or task. The system has to search and match the qualified resource from a mass of potential resource service set according to user requirements. Therefore, the searching and matching process is not simply comparing between two resource services but is a very complicated process.

Set the requested resource service (or task) describing information is A. The declared matching threshold values are ζ_{bas} (i.e., the basic matching threshold), $\zeta_{\mathrm{i/o}}$ (i.e., the I/O matching threshold), ζ_{QoS} (i.e., the QoS matching threshold), and ζ (i.e., the integrated matching threshold). The requested general information, input information, output information, and QoS information of A are *A.General*, *A.Inputs*, *A.Outputs*, and *A.QoS*, respectively. $\mathrm{W_A},\mathrm{D_A},\mathrm{C_A}$ are the corresponding subsets of word concept parameters, number parameters, entity class parameters of I/O parameters set. The intersection of any two of $\mathrm{W_A},\mathrm{D_A},\mathrm{C_A}$ is empty.

It is assumed that there are $m(m=1,2,3\ldots)$ potential advertising resource services for selection and the advertising resource service set is $B=\{B_1,B_2,B_3,\ldots,B_m\}$. Let B_j*.General*, B_j*.Inputs*, B_j*.Outputs*, B_j*.QoS* be the corresponding description of an arbitrary resource service, B_j, in B. $\mathrm{W_B},\mathrm{D_B},\mathrm{C_B}$ are the corresponding subsets of word concept parameters, number parameters, and entity class parameters of B_j's I/O parameters set.

According to the above matching methods and algorithms, the comprehensive searching and matching algorithms are as follows:

Inputs: *A.General*, *A.Inputs*, *A.Outputs*, *A.QoS*, $\mathrm{W_A},\mathrm{D_A},\mathrm{C_A}$, ζ_{bas}, $\zeta_{\mathrm{i/o}}$, ζ_{QoS}, ζ,$\mathrm{W_B},\mathrm{D_B},\mathrm{C_B}$, B_j*.General*, B_j*.Inputs*, B_j*.Outputs*, B_j*.QoS*, $\mathrm{W_B},\mathrm{D_B},\mathrm{C_B}$
Outputs: Qualified resource service set
1 For each B_j in B
Basic matching:
2 Calculating $\mathrm{Match_w}$ (*A.ServiceName*, B_j*. ServiceName*) according to expression 1
3 Calculating $\mathrm{Match_T}$ (*A.ServiceDescription*, B_j*. ServiceDescription*) according to expression 5
4 Calculating $\mathrm{Match}_{\mathrm{bas}}(A,B_j)$ according to expression 15
5 If $(\mathrm{Match}_{\mathrm{bas}}(A,B_j)<\zeta_{\mathrm{bas}})$,
6 Delete B_j from B, return failed matching message and back to step 1
7 End if
8 Record basic matching result $\mathrm{Match}_{\mathrm{bas}}(A,B_j)$, and execute the following I/O matching
I/O matching:
9 Classify the I/O parameters set of B_j into three subsets $\mathrm{W_B},\mathrm{D_B},\mathrm{C_B}$, which denote the corresponding subsets of word concept parameters, number parameters, entity class parameters, respectively.
10 If $\mathrm{W_B}$ is not empty
11 For each $B_j.w_i$ in $\mathrm{W_B}$ and corresponding $A.w_i$ in $\mathrm{W_A}$
12 Calculate $\mathrm{Match_W}(A.w_i,B_j.w_i)$ according to expression 1
13 End for
14 End if
15 If $\mathrm{D_B}$ is not empty
16 For each $B_j.d_i$ in $\mathrm{D_B}$ and corresponding $A.d_i$ in $\mathrm{D_A}$
17 If $B_j.d_i$ is a number interval parameter
18 Calculate $\mathrm{Match_D}(A.d_i,B_j.d_i)$ according to expression 6
19 End if
20 If $B_j.d_i$ is a triangular fuzzy number parameter
21 Calculate $\mathrm{Match_D}(A.d_i,B_j.d_i)$ according to expressions 9 and 10
22 End if
23 If $B_j.d_i$ is a trapezoidal fuzzy number parameter
24 Calculate $\mathrm{Match_D}(A.d_i,B_j.d_i)$ according to expressions 9 and 11
25 End if
26 End for

27 End if
28 If C_B is not empty
29 For each $B_j.c_i$ in C_B and corresponding $A.c_i$ in C_A
30 Calculate $\mathrm{Match}_C(A.c_i,B_j.c_i)$ according to expression 12
31 End for
32 End if
33 Calculate $\mathrm{Match}_{i/0}(A,B_j)$ according to expression 16
34 If $\mathrm{Match}_{i/0}(A,B_j)<\zeta_{i/o}$
35 Delete B_j from B, return failed matching message and back to step 1
36 End if
37 Record the matching result of $\mathrm{Match}_{i/0}(A,B_j)$ and execute the following QoS matching
QoS matching:
38 For each QoS parameters $A.QoS_i$ in $A.QoS$ and corresponding $B_j.QoS_i$ in $B_j.QoS$
39 If $A.QoS_i$ and $B_j.QoS_i$ are *number interval* parameters
40 Calculate $\mathrm{Match}_D(A.QoS_i,B_j.QoS_i)$ according to expression 6
41 End if
42 If $A.QoS_i$ and $B_j.QoS_i$ are *trapezoidal fuzzy number* parameters
43 Calculate $\mathrm{Match}_D(A.QoS_i,B_j.QoS_i)$ according to expressions 9 and 10
44 End if
45 If $A.QoS_i$ and $B_j.QoS_i$ are *number interval* parameters
46 Calculating $\mathrm{Match}_D(A.QoS_i,B_j.QoS_i)$ according to expressions 9 and 11
47 End if
48 End for
49 Calculating $\mathrm{Match}_{QoS}(A,B_j)$ according to expression 17
50 If $\mathrm{Match}_{QoS}(A,B_j)<\zeta_{QoS}$
51 Deleting B_j from B, return matching failed message and back to step 1
52 End if
53 Memorizing the matching the result $\mathrm{Match}_{QoS}(A,B_j)$ and executing the following integrated matching
Integrated matching:
54 Calculating $\mathrm{Match}(A,B_j)$ according to expression 14
55 If $\mathrm{Match}(A,B_j)<\zeta$
56 Deleting B_j from B, return matching failed message and back to step 1
57 End if
58 Returning the result $\mathrm{Match}(A,Bj)$ and back to step 1
59 End for

6 Case study

Supposed the requested describing information (A) for a parameterized radial magnetic design service on our experimental prototype platform, MBRSSP-MGrid [3], is shown on the left of Table 1. The declared threshold values are ζ_{bas}=0.85, $\zeta_{i/o}$=0.75, ζ_{QoS}=0.80, and ζ= 0.80. An available advertising resource service's corresponding describing information is shown on the right of Table 1.

Step 1 Basic matching

Step 1.1 Calculate the similarity of *A.ServiceName* and *B.ServiceName.*

According to expression 1 and expression 2, Match_w(*A. ServiceName*,*B. ServiceName*)=1.0000.

Step 1.2 Calculate the similarity of *A. ServiceDescription* and *B. ServiceDescription.*

According to Table 1, it is known that *A.Service-Description* = {Providing radial magnetic bearing parameterized design service according to the parameters and QoS requirements submitted by user} and *B.Service-Description* = {Providing client with parameterized design service for radial magnetic bearing according to the submitted parameters and QoS requirements}.

Set *A. ServiceDescription* = $l_i = At_i$,*B. ServiceDescription* = $l_j = Bt_j$. Then, the union set (i.e., $W^{\cup}$) of At_i and Bt_j is as follows:

$W^{\cup}$ = {Providing radial magnetic bearing parameterized design service according to the parameters and QoS requirements submitted by user client with for}.

Table 1 Brief describing information of two compared resource services based on OWL-S: the left is the requested information by a user and the right is the corresponding describing information of an advertising resource service

Requested resource service (*A*)	Available advertising resource service (*B*)

A{
General:{*ServiceName*{Radial Magnetic bearing parameterized design Service}, *ServiceDescription*{Providing radial magnetic bearing parameterized design service according to the parameters and QoS requirements submitted by user}}
Inputs:{{0.40,0.30,0.30}, {order}, {ParameterList {magnetic material, conducting wire, air gap, rotor diameter, static bearing capacity, bias, magnetic flux intensity}}, {3D_softwares}, {0.20,030,0.10,0.10}}
Outputs:{3D picture, 2D picture}
QoS:{*C*{140, 185}, *Trust*{0.30,0.10,0.10}, *Ma*{0.25,0.25,0.15,0.15}{},{},{},{}}
Precondition:{pay 50% money}
Operation:{{computing and design}}
Effect:{}
}
	B{
	General:{*ServiceName* {Radial Magnetic bearing parameterized design Service'}, *ServiceDescription* {Providing client with parameterized design service for radial magnetic bearing according to the submitted parameters and QoS requirements}}
	Inputs:{{0.40,0.10,0.10},{order}, {ParameterList {conducting wire, air gap, rotor diameter, static load capacity, bias, magnetic flux intensity}}, {3D_softwares},{0.55,0.55,0.15,0.15}}
	Outputs:{3D picture, 2D picture}
	QoS:{*C*{165, 215}, *Trust*{0.20,0.10,0.10}, *Ma* {0.20,0.30,0.10,0.10}{},{},{},{}}
	Precondition:{{pay 50% money}}
	Operation:{{computing and design}}
	Effect:{}
	}

The describing information are processed for the sake of illustration.

The required variables (i.e., l_i,l_j,OT_i, OT_j, $\left|OT_j\right|$, $\left|OT_j\right|$, $\sum_{k=1}^{m} s_{i,k}$, $\sum_{k=1}^{m} s_{j,k}$) in expressions 5 are as follows:

$$l_i = 17,\ l_j = 18,\ O_{T_i} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$$
$$O_{T_i} = \{1, 18, 19, 5, 6, 7, 20, 2, 3, 4, 8, 9, 10, 15, 11, 12, 13, 14\}$$
$$\left|OT_j\right| = 16,\ \left|OT_j\right| = 13,\ \sum_{k=1}^{m} s_{i,k} = 17,\ \sum_{k=1}^{m} s_{j,k} = 18.$$

Set α=0.7, β=0.2, δ=0.1, respectively. According to expressions 5, the final similarity between At_i (i.e., *A. ServiceDescription*) and Bt_i (i.e., *B.ServiceDescription*) is calculated as follows:

$$\mathrm{Match}_T\left(At_i, Bt_j\right) = \alpha\left(1 - \frac{|17-18|}{17+18}\right) + \beta\left(1 - \frac{|13-16|}{13+16}\right) + \delta\left(1 - \frac{|17-18|}{17+18}\right) = 0.9564.$$

Step 1.3 Calculate the basic matching degree of *A* and *B*

Set ω_1=0.1, ω_1=0.9, according to expressions 15, $\mathrm{Match}_{bas}(A,B)$=0.1×$\mathrm{Match}_w$(*A.ServiceName*, *B. ServiceName*)+0.9×$\mathrm{Match}_T(At_i,Bt_j)$=0.9698

Step 2 I/O matching

Step 2.1 Classify the parameters of I/O into D,C,W

Let D_A,C_A,W_A be the subsets of number parameters, entity class parameters, word concept parameters of the I/O parameters of *A*, and D_B,C_B, W_B are that of *B*. Then,

D_A={{0.40,0.30,0.30},{0.20,030,0.10,0.10}}

W_A={{order}, {3D_softwares},{3D picture},{2D picture}}

C_A={{*ParameterList*}} = {{magnetic material,conducting wire, air gap, rotor diameter, static bearing capacity, bias, magnetic flux intensity}}

D_B={{0.40, 0.10, 0.10}, {0.55, 0.55, 0.15, 0.15}}

W_B={{order}, {3D_softwares},{3D picture},{2D picture}}

C_B={{*ParameterList*}} = {{magnetic material,conducting wire, air gap, rotor diameter, static bearing capacity, bias, magnetic flux intensity}}

Step 2.2 Calculate the similarity of the word concept parameters of I/O

$$\text{Let } W_A = \{\{\text{order}\}, \{\text{3D\_softwares}\}, \{\text{3D picture}\}, \{\text{2D picture}\}\} = \{A.w_1, A.w_2, A.w_3, A.w_4\}$$
$$W_B = \{\{\text{order}\}, \{\text{3D\_softwares}\}, \{\text{3D\_picture}\}, \{\text{2D picture}\}\} = (B.w_1,\ B.w_2,\ B.w_3,\ B.w_4).$$

According to expression 1 and expression 2, $\mathrm{Match_W}(A.w_1,B.w_1) = \mathrm{Match_W}(A.w_2,B.w_2) = \mathrm{Match_W}(A.w_3,B.w_3) = \mathrm{Match_W}(A.w_4,B.w_4) = 1$

Step 2.3 Calculate the similarity of the number parameters of I/O

$$Let\ D_A = \{\{0.40, 0.30, 0.30\}, \{0.20, 030, 0.10, 0.10\}\} = \{A.d_1, B.d_2\},\ and$$
$$D_B = \{\{0.40, 0.10, 0.10\},\ \{0.55, 0.55, 0.15, 0.15\}\} = \{B.d_1, B.d_2\}$$

Then,

- According to expressions 9 and 10, $\mathrm{Match_D}(A.d_1,B.d_1)$= 0.8752.
- According to expressions 9 and 11, $\mathrm{Match_D}(A.d_2,B.d_2)$= 0.4444.

Step 2.4 Calculate the similarity of the entity class parameters of the I/O

Set A. ParameterList = Ac_1 and *B. ParameterList* = Bc_1. According to expressions 12 and 13,

$Ac \cap Bc$	{conducting wire, air gap, rotor diameter, bias, magnetic flux intensity}
Ac/Bc	{magnetic material, static bearing capacity}
Bc/Ac	{static load capacity}
$\alpha(\mathrm{Ac_1,Bc_1})$ =	α(*A. ParameterList*, *B. ParameterList*)=0.5

Therefore, according to expressions 12 and 13, the similarity between Ac_1 Bc_1 is

$$Match_C(Ac_1, Bc_1) = \frac{5}{5 + 0.5 \times 2 + 0.5 \times 1} = 0.7692.$$

Therefore, the similarity between *A. ParameterList* and *B. ParameterList* is 0.7692.

Step 2.5 Calculate the I/O matching degree of *A* and *B*

Let the weight of each I/O parameter be the same, according to expression 16, $\mathrm{Match_{I/O}}(A,B)$=0.8698.

Step 3 QoS-matching

Step 3.1 Calculate the similarity of each QoS parameter From Table 1, it is known that:

$$\begin{aligned} A.QoS &= \{A.QoS_1, A.QoS_2, A.QoS_3\} \\ &= \{C\{140,\ 185\},\ Trust\{0.30, 0.10, 0.10\}, Ma\{0.25, 0.25, 0.15, 0.15\}\} \\ B.QoS &= \{B.QoS_1, B.QoS_2, B.QoS_3\} \\ &= \{C\{165, 215\},\ Trust\{0.20, 0.10, 0.10\},\ Ma\{0.20, 0.30, 0.10, 0.10\}\} \end{aligned}.$$

Apparently, $A.QoS_1$ and $B.QoS_1$ are numerical interval parameter, $A.QoS_2$ and $B.QoS_2$ are triangular fuzzy number, and $A.QoS_3$ and $B.QoS_3$ are trapezoidal fuzzy number. Therefore,

- according to expression 6, $\mathrm{Match_W}(A.QoS_1,B.QoS_1)$=|[140,185]|/|[165,215]|=0.9000
- according to expressions 9 and 10, $\mathrm{Match_D}(A.QoS_2, B.QoS_2)$=0.7866
- according to expressions 9 and 11, $\mathrm{Match_D}(A.QoS_3, B.QoS_3)$=0.8607

Step 3.2 Calculate the QoS matching degree of *A* and *B*

Set $\omega_{\mathrm{QoS}_1} = 0.2$, $\omega_{\mathrm{QoS}_2} = 0.5$ and $\omega_{\mathrm{QoS}_3} = 0.3$, according to expressions 17, $\mathrm{Match_{QoS}}(A,B)$=0.2×0.9000+0.5×0.7866+0.3×0.8607=0.8315

Step 4: Integrated matching

According to above proposed matching method and algorithms in Section 5, the system first calculates the basic matching values of *A* and *B* then calculates the I/O matching value and QoS matching value and last calculates the entire matching value, i.e., integrated matching. The corresponding matching values are shown in Table 2. From the result shown in Table 2, it can be concluded that the advertising service *B* is qualified and can be selected as a candidate resource service.

7 Performance results and discussion

To validate the proposed method, a set of experiments are conducted on the experimental MGrid prototype platform, "magnetic bearing resources sharing and service system under manufacturing grid environment (MBRSSP-MGrid)" [3], which is developed by us. In MBRSSP-MGrid, all users can publish their idle resources (including equipment resources, software resources, human resources, application resources, technique resources, service resources, etc.) through the resource service publication center of MBRSSP-MGrid. Users can also search the resources or services (e.g., remote parameterized design service of magnetic bearing) they required via the resource and service optimal allocation center of MBRSSP-MGrid. The experiments and evaluation

Table 2 Corresponding matching result between *A* and *B*

	Matching value	Weights	Threshold	Compared with threshold	Whether qualified?
Basic matching	0.9608	0.2	0.85	>	Yes
I/O matching	0.8698	0.45	0.75	>	Yes
QoS matching	0.8315	0.35	0.80	>	Yes
Integrated matching	0.8746	/	0.80	>	Yes

was focused on the accuracy and efficiency of our proposed method.

7.1 Accuracy

The described RSMS method is compared with UDDI keyword matching (hereinafter referred to as UDDI). In our experiments, a set of 50 resource services are selected for a submitted task, of which 20 services are relevant to the task. *Precision* and *recall*, which are the standard measures that have been used in information retrieval for measuring the accuracy of a search method or search engine, are selected as the criteria to test the accuracy of RSMS. Let N_{Rel} be the set of relevant resource services, N_{Ret} be the set of returned resource services, and $N_{\mathrm{Rel}}^{\mathrm{Ret}}$ be the set of returned relevant resource services. In this work, recall, R_{ecall}, and precision, P_{recision}, be defined as follows [40]:

$$R_{ecall} = \frac{N^{\mathrm{Ret}}}{N_{\mathrm{Rel}}}, P_{\mathrm{recision}} = \frac{N_{\mathrm{Rel}}^{\mathrm{Ret}}}{N^{\mathrm{Ret}}}.$$

During the implementations, two conditions are considered: (I) the 50 candidate resource services have formal describing information, without fuzzy describing information, entity class describing information, etc. and at least one parameters of each candidate resource service is assigned an *exact* match with the submitted query of task and (II) without the above constraints. Each condition has ten tests, and each test is executed ten times; the result of each test is the average of ten times executions, as shown in Fig. 8.

It can be concluded from Fig. 8 that RMSM has better performance in the tests both under condition (I) and condition (II). In condition (I), the recall of UDDI can reach 100% because of the two constraints. But its precision is lower than RSMS because UDDI only considers keyword matching, and the matching of other kinds of describing information for resource service is ignored. For example, if a candidate resource has five describing parameters, only one is keyword, the other four are fuzzy numbers, entity class, sentence, etc. UDDI only matches one parameter, and RSMS matches all five parameters. Apparently, the precision of RSMS is higher than UDDI.

Figure 8b illustrates that UDDI does not reach a recall of 90% and 100% because of its limitations in resource service matching. For example, if there is not keyword or exact match between a resource service and task, then the matching degree of this resource service is zero when using UDDI. As a result, the resource service cannot be returned as a candidate for selection.

7.2 Efficiency

In order to measure the efficiency of the proposed method, the time between the submission of a batch of tasks and the return of matched results are measured, which is called *resource service matching response time* (RSMS-time) in this work. A batch of 10, 20, and 30 tasks are submitted, and the required resource services for each task vary from two to five. During the matching process, each resource service in RSIC can only be assigned to one task. A match for a task is successful only if all required resource services of the task are found and matched. The whole matching process stops only when all submitted tasks are matched to related resource services.

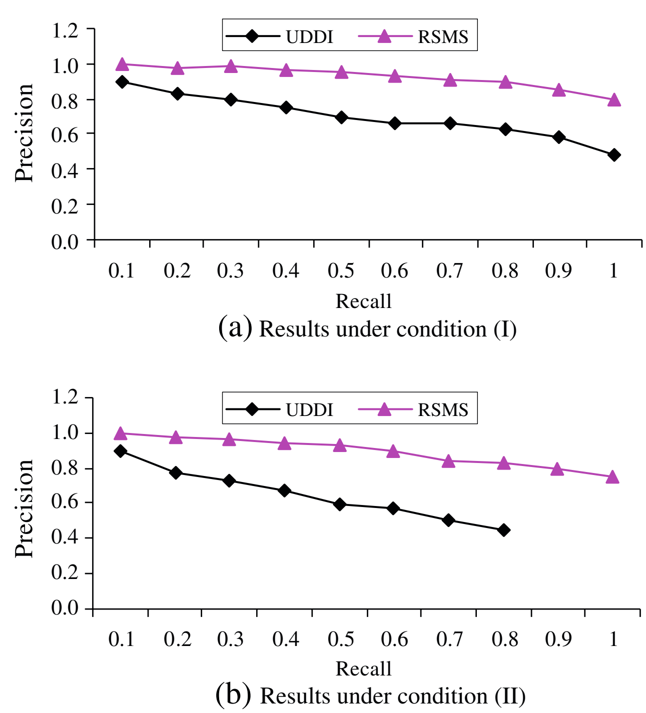


Fig. 8 Comparisons of performance between RSMS and UDDI

The experiments are repeated 20 times and the RSMS-time is the average over the 20 executions. The results are shown in Fig. 9. The results in Fig. 9 indicate that the proposed method can handle resource services matching for tens of tasks in several seconds. In order to test the practical efficiency of the proposed matching algorithms, the RSMS-time is broken down into three parts: (a) the time spent on communication, denoted as *CommTime*; (b) the time used for matching, denoted as *MatchTime*; and (c) the time spent on querying resource services information from RSIC or corresponding repository, denoted as *QueryTime*.

From Fig. 9, we can see that most time is spent on communications. It is almost the same for different number of tasks because all task submissions make the same service call with different parameters. The second largest part of the time is spent on querying resource service repository. It increases linearly with number of tasks. This is because in our implementation, the system makes one query for each task's request. The smallest part of time, tens milliseconds, is spent on matching and finding qualified resource services for tasks. This result indicates that the proposed matching method is time-efficient.

8 Conclusion and future work

RSMS is the key in implementing a real-time MGrid. Current works to service match and search primarily concentrate on abstract service discovery mechanisms and methods. The research to practical SMAs between basic describing information of services is insufficient. Without the supports of SMAs for basic describing information of services, the upper service discovery cannot be realized. In this paper, a resource services match and search mechanism is proposed. It is suitable for MGrid environment where resource services are described not only using common describing information such as keyword but also are described using sentence information, fuzzy number information, entity class information, etc. The primary works and contribution of this paper are as follows:

1. The describing information of resource services are classified into four categories: (a) word concept information, (b) sentence information, (c) number information, including number interval and fuzzy number, and (d) entity class (or data structure) information. The SMAs for the four kinds of basic describing information are described, including (a) WMAs, (b) SeMAs, (c) NMAs, and (d) ECMAs.
2. Under the supports of the proposed SMAs, the process of resource services match and search are divided into four steps, they are (a) basic matching, matching the basic information of resource services, such as service name and service description; (b) I/O matching, matching the inputs and outputs information of resource services; (c) QoS matching, matching the QoS information of resource services; and (d) integrated matching, combining the above three matching results and generating an integrated matching result. The algorithms of each step are presented.
3. A case study is presented to illustrate the application of the SMAs and resource service match and search method described in this paper. The performance measurements from our prototype implementation indicate that proposed match and search method for MGrid resource service are efficient in accuracy and efficiency.

The paper only considered the primary four kinds of basic describing information for resource services; some other kinds of describing information are not considered. Describing information of tasks and resource services involved in matching are assumed symmetrical, e.g., have the same number of parameters and the same type of information. In the future, we will further research the classification of describing information for resource services in MGird and study the corresponding similarity matching algorithms and use them in MGrid resource services match and search. Furthermore, investigation for conflicts and failures detection and recovery are recommended for farther research.

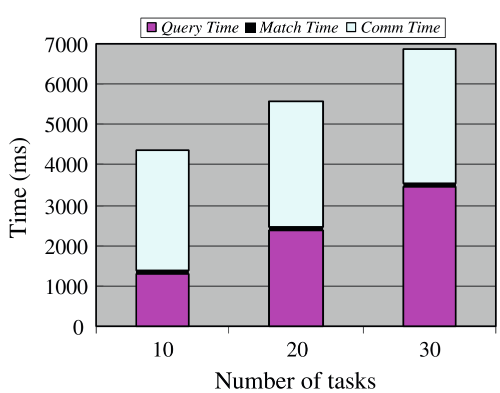


Fig. 9 RSMS-time

Acknowledgments The first author wishes to acknowledge the financial support of the Excellent Doctoral Dissertation Fund of WHUT (Wuhan University of Technology). This paper is supported by Hubei Digital Manufacturing Key Laboratory Opening Fund project: research on resource service search and optimal-selection theories and experiment in manufacturing grid system (Project No. SZ0621), and the National Nature Science Fund Project of China: Research on new theories and new technologies for network-based digital manufacturing environment (Project No.50620130441).

Thanks for all the authors of the references who gives us inspirations and helps. The authors are grateful to the editors and anonymous reviewers for their valuables comments that improved the quality of this paper.

References

1. Qiu RG (2004) Manufacturing grid: a next generation manufacturing. 2004 IEEE International Conference on System, Man and Cybemetics (SMC2004), October 10–13. The Hague, The Netherlands, pp 4667–4672
2. Li Z, Jin X, Cao Y, Zhang X, Li Y (2007) Conception and implementation of a collaborative manufacturing grid. Int J Adv Manuf Technol 34:1224–1235. doi:10.1007/s00170-006-0677-1
3. Tao F, Hu YF, Zhou ZD (2008a) Study on manufacturing grid & its resources optimal-selection system. Int J Adv Manuf Technol 37(9–10):1022–1041. doi:10.1007/s00170-007-1033-9
4. Tao F, Hu YF, Zhao DM, Zhou ZD, Zhang HJ, Lei ZZ (2008b) Study on manufacturing grid resource service QoS modeling and evaluation. Int J Adv Manuf Technol. doi. 10.1007/s00170-008-1534-1
5. Tao F, Hu YF, Zhou ZD (2007) Application and modeling of resource service trust-QoS evaluation in manufacturing grid system. Int J Prod Research. doi:10.1080/00207540701551927
6. Li L, Horrock I (2004) A software framework for matchmaking based on semantic web technology. Int J Electron Commerce 8 (4):39–60
7. Paolucci M, Kawamura T, Payne T, Sycara K (2002) Semantic matching of web services capabilities. Proceedings of the First International Semantic Web Conference (ISWC 2002), June 9–12. Sardinia, Italy, pp 333–347
8. Sycara K, Klusch M, Widoff S, Lu J (2002) Larks: dynamic matchmaking among heterogeneous software agents in cyberspace. J Autonomous Agents Multi-Agent Syst 5(2):173–203. doi:10.1023/A:1014897210525
9. Shen ZN, Su JW (2005) Web service discovery based on behavior signatures. Proceedings of the 2005 IEEE International Conference on Service Computing (SCC'05), July 11–15. Orlando, Florida, USA, pp 279–286
10. Perryea CA, Chuang S (2006) Community-based service discovery. The 2006 IEEE International Conference on Web Service (ICWS'06), September 18–22. Chicago, USA, pp 903–906
11. Doulkeridis C, Zafeiris V, Norvag K, Vazirgiannis M, Giakoumakis EA (2007) Context-based caching and routing for P2P web service discovery. Distrib Parallel Databases 21(1):59–84. doi:10.1007/s10619-006-7000-x
12. Balken R, Haukrogh J, Jensen JL, Jensen MN, Roost LJ, Toft PN et al (2007) Context-sensitive service discovery experimental prototype and evaluation. Wirel Pers Commun 40(3):417–431. doi:10.1007/s11277-006-9200-0
13. Raverdy PG, Issarny V (2005) Context-aware service discovery in heterogeneous networks. Proceedings of the sixth IEEE International Symposium on a Word of Wireless Mobile and Multimedia Networks (WoWMoM'05), June 13–16. Taormina, Italy, pp 478–480
14. Lee C, Helal S (2003) Context attributes: an approach to enable context-awareness for service discovery. Proceedings of the 2003 symposium on application and the Internet (SAINT'03), January 27–31. Orlando, Florida, pp 22–30
15. Kokash N, Birukou A, D'Andrea V (2007) Web service discovery based on past user experience. LNCS 4439:95–107
16. Alberto F Matteo, Cesar C, Sascha O (2007) A role-based support mechanism for service description and discovery. SOCASE 2007, LNCS 4504, pp 132–146
17. Tomas V, Maciej Z, Matthew M (2007) Dynamic service discovery through met-interaction with service provider, ESWC 2007. LNCS 4519:84–98
18. Jia Y, Srikumar V, Rajlumar B (2006) A market-oriented grid directory service for publication and discovery of grid service providers and their services. J Supercomput 36(1):17–31. doi:10.1007/s11227-006-3073-6
19. Zisman A, Spanoudakis G (2006) UML-based service discovery framework. 4th International Conference on Service Oriented Computing (ICSOC 2006), December 4–7. Chicago, USA, LNCS 4292, pp 402–414
20. Stollberg M, Keller U, Lausen H, Heymans S (2007) Two-phase web service discovery based on rich functional description. 4th European Semantic Web Conference (ESWC 2007), June 3–7. Tyrol region of Innsbruck, Austria, pp 99–113
21. Bianchini D, DeAntonellis V, Melchiori M (2005) An ontology-based architecture for service discovery and advice system. Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05), August 22–26. Copenhagen, Denmark, pp 551–556
22. Liu LL, Yu T, Shi ZB, Fang ML (2003) Self-organization manufacturing grid and its task scheduling algorithm. Comput Integr Manuf Syst 9(6):449–454
23. Deng H, Chen L, Wang CT, Deng QN (2006) A grid-based scheduling system of manufacturing resources for a virtual enterprise. Int J Adv Manuf Technol 28:137–141. doi:10.1007/s00170-004-2335-9. doi:10.1007/s00170-004-2388-9
24. Chen Li Deng H, Deng Q N, Wu Z Y (2004) A research of grid manufacturing and its application in custom artificial joint. Proceedings of International Conference of Computer Science 2004(ICCS2004), April. Krakow, Poland, LNCS 3036:507–510
25. Zhang CS, Mo R, Shi SY, Chang ZY (2006a) Research on manufacturing grid resource scheduling based on genetic algorithm. Chin Mech Eng 17(18):1916–1920
26. Lv BS, Shi SY, Mo R, Chang ZY, Yang HC (2006) Market equilibrium based resource optimal allocation for manufacturing grid. Comput Integr Manuf Syst 12(12):2011–2016
27. Tan W, Fan YS (2005) Research on service matching and composition in networked manufacturing environment. Comput Integr Manuf Syst 11(106):1408–1413
28. Zhang L, Yuan WZ, Wang W (2006b) An ontology based approach of automated service chaining for manufacturing grid. Chin Mech Eng 17(14):1484–1488
29. Zhang L, Yuan WZ, Wang W (2006c) Automotive service composition for manufacturing grid based on domain-specific ontology. J Comput Appl 26(1):57–60
30. Lee KM, Chio KH, Her SP, Shin DR (2005a) Matchmaking algorithms to improve dynamic service matching in ubiquitous environments. Proceedings of Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05), July 14–16. Jeju Island, South Korea, pp 239–244
31. Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence, August 20–25. Montreal, Quebec, Canada, pp 448–453
32. Jiang JJ, Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. International Conference on Research in Computational Linguistics (ROCLING X 1997), August 22–24. Taiwan, pp 19–33
33. Lin LF, Gao P, Cai M, Dong JX (2005) A knowledge service-based model of collaborative manufacturing process planning for networked manufacturing. Journal of Computer-Aided Design & Computer Graphics 17(9):2085–2091

34. Li YH, Bandar ZA, McLean D (2003) An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans Knowl Data Eng 15(4):871–882. doi:10.1109/TKDE.2003.1209005
35. Li YH, McLean D, Bandar ZA, O'Shea JD, Crockett K (2006) Sentence similarity based on semantic nets and corpus statistics. IEEE Trans Knowl Data Eng 18(8):1138–1150. doi:10.1109/TKDE.2006.130
36. Lee KH (2005b) Fuzzy number, First course on fuzzy theory and applications (Advance in Soft Computing) [M], vol 27. Springer, Berlin, pp 129–151 (ISBN:978-3-540-22988-9). http://dx.doi.org/10.1007/3-540-32366-X
37. Yang MS, Hung WL, Chang-Chien SJ (2005) On a similarity measure between LR-type fuzzy numbers and its application to database acquisition. Int J Intell Syst 20(10):1001–1016. doi:10.1002/int.20102
38. Tversky A (1977) Features of similarity. Psychol Rev 84(4):327–352. doi:10.1037/0033-295X.84.4.327
39. Rodriguez MA, Egenhofer MJ (2003) Determining semantic similarity among entity classes from different ontologies. IEEE Trans Knowl Data Eng 15(2):442–456. doi:10.1109/TKDE.2003.1185844
40. Li M, Yu B, Rana O, Wang Z (2008) Grid service discovery with rough sets. IEEE Trans Knowl Data Eng 20(6):851–862. doi:10.1109/TKDE.2007.190744