

一种基于功能语义的 Web 服务描述及预检索方法

赵文栋 陶晓臻 彭来献 田 畅

(解放军理工大学通信工程学院 南京 210007)

摘 要 实现基于功能语义的 Web 服务过滤是简化服务检索计算量、提高服务发现准确率的有效方法之一。针对现有常用 Web 服务描述语言不能很好地支持服务基于功能的语义描述的问题,通过定义服务功能描述模型,构建领域功能本体,提出了一种基于功能语义的 Web 服务描述方法,并详细给出了基于 OWL-S 的实现,在此基础上,给出了一种服务预检索方法。该方法可有效地支持用户实现基于功能的语义服务检索,滤除无关服务,降低精确匹配计算量,提高服务查找效率。

关键词 功能语义,服务描述,领域本体,预检索

中图分类号 TP393 文献标识码 A

Function Semantic-based Web Service Description and Pre-filter Method

ZHAO Wen-dong TAO Xiao-zhen PENG Lai-xian TIAN Chang

(Institute of Communication Engineering, PLA University of Science and Technology, Nanjing 210007, China)

Abstract Functional semantic based Web service filter is a useful method to decrease the computation of service discovery. According to the problem that current semantic-based services description languages can not support service functional description, this paper firstly defined a functional based Web service semantic description model and created a domain oriented functional ontology. Based on this, a service description method was raised and a realize method based on OWL-S was expressed clearly. Finally a pre-filter method was raised. The pre-filter method can support the service discovery based on functional semantic, filter out unrelated services, decrease the computation of accurate service match and increase the matching efficiency.

Keywords Functional semantic, Service description, Domain ontology, Pre-filter

1 引言

随着 Web 服务的普及、服务数量的急剧增加,如何在海量服务群中发现满足需要的服务,即 Web 服务检索,成为影响 Web 服务进一步发展的一个瓶颈问题。Web 服务的检索技术主要分为基于关键词的服务检索技术与基于语义的服务检索技术两类。基于语义的服务检索技术利用本体技术,增强了对 Web 服务的功能、行为的语义描述,通过逻辑推理、相似度计算等方法实现 Web 服务的检索,解决了基于关键字的服务匹配算法准确性不高的问题。

虽然基于语义的服务描述及检索技术提高了服务检索的准确性,但是匹配算法普遍存在设计复杂、计算量大的问题。如果服务器每收到一个服务请求,都要与本地存贮的所有服务完成相似度计算,则会引入巨大的计算量,特别是在服务数量剧增的情况下,采用逐个比对的方式会使服务器的检索效率明显降低。同时,从 Web 服务的发现流程考虑,基于功能语义进行服务发现为用户最为直接的需求。即在服务匹配时,用户会首先考虑所查找的服务的功能是否能满足自身需求,而后才会做进一步的精确匹配。而现有主流基于语义的服务描述语言,如 OWL-S、WSDL-S 等重点考虑语言适用范

围的广泛性,无法很好支持基于功能语义的服务检索。

服务描述是服务检索的基础,针对上述问题,本文从服务描述方法入手,通过定义服务功能描述模型,构建领域功能本体,提出了一种基于功能语义的服务描述方法,并给出了基于 OWL-S 的实现。在此基础上,提出了一种基于功能的服务预检索方法。该方法可支持用户基于功能语义的服务查找,有效地滤除无关服务,提高服务查找效率。

本文第 2、3 节介绍了相关工作及背景知识;第 4 节给出了服务功能语义描述模型及基于 OWL-S 的扩展实现;第 5 节介绍了一种基于功能语义本体的服务映射方法及性能仿真分析;最后总结全文。

2 相关工作

现有的服务描述及检索方法主要分为两大类^[1]:一是基于语法的描述及检索方法^[2],本类方法主要采用 WSDL 作为服务描述语言,通过关键字匹配进行服务搜索,如 UDDI,采用了基于预定义分类的服务发现机制。由于描述语言的限制,使其不能很好地支持对服务功能与行为的语义描述。算法虽然实现简单,但是由于基于关键词的描述对用户的描述内容提出了很高的要求,致使查全率和查准率较低,且无法

到稿日期:2013-01-20 返修日期:2013-07-23 本文受国家 973 项目(2009CB3020402),国家自然科学基金项目(61103224),江苏省自然科学基金项目(BK2011118)资助。

赵文栋(1972-),男,博士生,副教授,主要研究方向为计算机网络,E-mail:nj_mouse@163.com。

支持用户基于功能语义进行服务发现。二是基于语义的服务描述及检索方法^[3],其主要思路是通过本体描述各概念间的关系,然后基于本体,通过逻辑推理进行服务匹配。这种方法增强了对 Web 服务的语义描述,放宽了对用户服务(请求)描述的限制,具有查准率高的优点。存在的主要问题有:一是现有基于语义的服务描述语言,如 SAWSDL、OWL-S、SWSO,对 Web 服务描述过于笼统、简单,限制了服务检索效率的进一步提高。二是基于语义的服务检索匹配算法涉及参数多,普遍存在计算量大的问题,因此现有多篇文献针对特定的应用领域及服务检索需求,采用本体与多级匹配相结合的方式,提出了不同的服务描述及检索方法。文献[4]针对军事信息服务领域的指挥特点,提出了一种基于行为约束服务描述及服务过滤方法。文献[5,6]则通过扩充本体的方式,增加对服务 QoS 的描述,以达到提高查准率的目的。但其服务匹配计算量没有降低。文献[7]在 OWL-S 的基础之上扩充了对服务 QoS 的描述,并以此对服务进行基于 QoS 分类,以降低服务匹配计算量。与本文类似,文献[8-10]重点讨论了基于功能的服务检索技术。文献[10]通过分析服务本体描述中各概念间的关联关系,来获取服务的功能信息,并给出了一种基于 OWL-S 的实现方法。文献[8]设计了一种服务功能的格式化描述方式,通过分析、提取服务描述的功能信息,完成服务基于功能的格式化描述。文献[9]则在 WSDL 的基础之上,采用本体技术实现了对服务功能描述能力的扩充。本文与文献[8-10]不同的是,针对特定领域功能需求,重点考虑了服务功能及其属性限制条件的描述及复杂服务与简单服务间的关联关系,设计了服务功能描述模型和基于功能的服务描述及预检索方法。本描述方法是通过扩充 OWL-S 本体实现的,可有效地继承现有基于语义的服务精确匹配的研究成果,便于用户进行二次精确匹配。下面,为了便于描述,首先介绍相关背景知识。

3 背景知识

(1) 本体

本体是指“对共享的概念体系进行明确的形式化规范说明”,作为语义技术的主要实现手段,一直是研究者们关注的重点。其最初是一个哲学概念,出现在公元前四世纪亚里士多德的机械唯物主义中,表征事物存在的本质。本体论与认识论相对应,认识论研究人类知识的本质和来源。也就是说,本体论研究客观存在,认识论研究主观认知。后来,人们将本体引入自然科学领域,其含义仍然是客观存在,但更加强调形式化描述。用形式化语言描述本体,一方面能使表述更加严谨,更重要的是方便计算机理解,从而实现认识过程的自动化。但从表达语义的角度来讲,广义的本体可认为是概念及其关系的集合。或者可将本体作为一种技术,利用本体技术表达语义,而不必受限于本体的一致性和完备性。

(2) OWL-S 及其扩充方法

OWL-S(Ontology-based Web Language-Service)是基于 OWL 的语义服务描述语言,它从语义的角度出发,使用 OWL 语言定义了服务本体,并将具体的服务作为服务本体的一个实例。OWL-S 语言实际上可以看作是 OWL 语言应用的特例。

OWL-S 建立了语义服务的顶层描述语义本体,其基本结

构如图 1 所示。将 Service 作为语义服务的根类,它有 3 个属性 presents(表达)、describedBy(被描述)、supports(支持),这 3 个属性的值域类分别是 ServiceProfile(服务概况)、ServiceModel(服务模型)、ServiceGrounding(服务基础)。每个具体的服务都是 Service 类的一个实例,声明为 Service 类的一个个体。不同的服务对应的 3 个属性在各自值域中取不同的值。简单来讲,可将 ServiceProfile(服务概况)、ServiceModel(服务模型)、ServiceGrounding(服务基础)看作是 OWL-S 描述语义服务的 3 个模块,分别描述服务的 3 个方面。

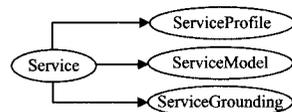


图 1 OWL-S 顶层服务本体

OWL-S 的灵活之处在于它本身也是一个本体文件,遵循 OWL 的所有语法规则,也可实现本体中所有关系的定义。实际上,OWL-S 的开发者将服务本体的原始定义写成了 Service. owl、Profile. owl、Process. owl、Grounding. owl 等几个本体文件,当程序解析 OWL-S 服务描述时,需要这些本体文件作为支持。专用的服务描述解析器,多数是在本体解析器的基础上,内嵌了上述本体文件,因此可在没有上述文件的情况下解析服务。由此可知,OWL-S 对服务的描述在一定程度上是开放的,并不会受到已有标记的限制。例如,在 Profile. owl 共描述了“Profile”的 11 个属性,包括“serviceName”、“text-Description”、“serviceCategory”、“hasInput”、“hasOutput”、“hasPrecondition”、“hasResult”等,但有些用户认为这些还不能完整地描述服务或者不便于服务的匹配组合,则可添加其他的标签,描述服务的其他方面,比如描述服务的功能,可为“Profile”添加一个“serviceFunction”属性,再根据需要设置相应的值域。修改本体文件并不影响解析器的使用,只要采用标准的本体解析器,就能准确解析得到服务的功能描述。

4 服务功能语义描述

为了统一服务提供者和用户对服务功能的描述方式,消除描述方式的差异,需要建立统一的 Web 服务功能语义描述模型。本节首先介绍了领域功能本体的设计模型,然后给出了基于 OWL-S 服务功能扩展实现方法及服务描述实例。

4.1 领域功能本体

针对特定领域设计的功能本体,其核心思想是用于统一描述此领域内的服务需求,即需要哪些服务功能,各服务功能之间的逻辑关系是什么。所以本文在设计领域功能本体时,以描述 Function Description 的概念为主体,Function Limitation 则作为 Function Description 的属性。

定义 1(原子服务) 特定领域内不可再分的服务。其功能语义描述如式(1)所示:

$$F_{atom} = \langle \text{Function Description}, \text{Function Limitation} \rangle \quad (1)$$

式中,Function Description 用于描述原子服务所能提供的功能,通常采用一组关键词描述,同时假设各原子服务功能是不相关的。即从功能的角度看,对任一原子服务,以下条件成立:

$$\forall F_{atom} \notin D \Rightarrow F_{atom} \cap F_{Batom} = \emptyset, \text{ iff } A \neq B$$

Function Limitation 用于描述服务的约束条件,如输入、

输出数据的格式、调用方式、生成结果的精度等。可表示为： $Function\ Limitation = P_1 \wedge P_2 \wedge P_3 \cdots \wedge P_n$ ，其中 P_i 为约束条件。

定义2(复合服务) 由原子功能服务通过一系列的操作组合成的服务。其组合方式可包括顺序、选择、并发、循环等方式，其功能语义描述如式(2)所示：

$$F_{complex} = \left\{ \begin{array}{l} \langle \{F_i | F_i = F_{atom}\}, \text{Complex Function Description} \rangle \\ F_{atom} = \langle \text{Function Description}, \text{Function Limitation} \rangle \end{array} \right. \quad (2)$$

即复合服务 $F_{complex}$ 由两部分组成，一部分是所涉及的各原子服务，二是 Complex Function Description(复杂服务功能描述)，它主要描述各服务组合后所能提供的功能及限制属性。并有以下条件成立：

$$F_{complex} = \Omega(F_1, F_2, F_3, \dots, F_n) \quad (3)$$

式中， $\Omega()$ 代表转换函数，即复合服务是由一组简单服务通过一定的组合方式完成的，组合服务并不是原子服务功能简单的叠加。需说明的是，为保证推理的正确性，对复合服务限制及性能的描述不能与原子服务的功能相抵触。在描述时，一个复杂的复合服务也可以是几个相对简单的复合服务的集合，但是，采用递归下降的分解方法，一定可以将一个复合服务表示成完全由原子服务组成的集合。

定义3 领域功能本体。 设 O 是领域功能本体，则 $O = \langle \{C\}, \{R\} \rangle$ (4)

式中， C 为概念集合，包括所包含的各分类功能概念及对各功能概念属性的描述。

R 为关系集合，由于文中本体的设计主要专注于领域内各功能及功能间关系的描述，此处在本体定义时，我们主要定义了两类关系： $include, equal$ 。其中 $include$ 关系表达复合服务功能与原子服务功能间的包含关系， $equal$ 表示两个服务功能是等价关系。

以战场态势信息服务为例，按照定义3，简要给出了服务功能本体关系图。其中“□”表示复合服务概念，“○”表示原子服务概念。

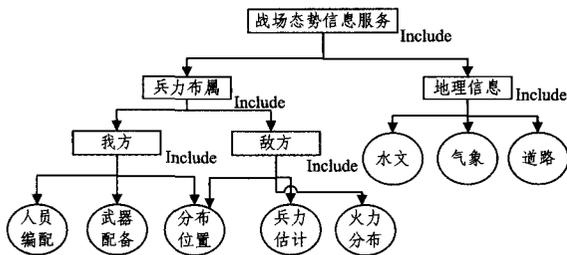


图2 战场态势信息服务功能本体关系图

从清晰的角度考虑，图中，只显示了概念间的包含关系。

4.2 基于OWL-S的服务功能描述扩展

根据本文对OWL-S的介绍，标准的OWL-S语言并不能显式地描述服务的功能，其对服务的功能描述主要隐含在Service Profile中的服务功能特性，即IOPE部分，计算机可以借助本体知识并通过对IOPE相关参数的分析，间接地对当前服务功能描述信息进行提取，但是这样做会存在以下问题：一是会引入很大的冗余计算量，降低服务匹配的效率；二是如果提取算法设计不当，会影响提取结果的准确度。

由于OWL-S的最大特点是，其描述语法结构是通过本

体文件给出的，只要我们对其本体描述文件稍做修改即可实现对OWL-S语言的扩充。下面首先给出扩展后的OWL-S的结构示意图，如图3所示。

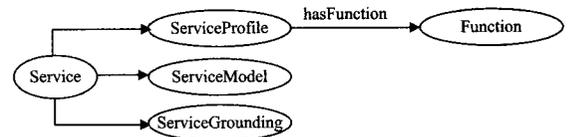


图3 OWL-S扩展顶级服务本体

本扩展过程需要修改两个本体文件：

(1)修改 profile.owl 文件，添加“hasFunction”描述。

```

在 profile.owl 中添加：
<owl:ObjectProperty rdf:ID="hasFunction">
  <rdfs:domain rdf:resource="#Profile"/>
  <rdfs:range rdf:resource="#&-process;#Function"/>
</owl:ObjectProperty>

```

(2)在 process.owl 文件中，增加“Function”类的定义

```

在 process.owl 中添加：
<owl:Class rdf:ID="Function">
  <rdfs:subClassOf rdf:resource="#Parameter"/>
</owl:Class>

```

以上两步完成了对服务描述语言OWL-S的扩充，描述具体的服务时可按照与描述输入输出相同的方法描述服务的功能。例如某服务的功能为“获取目标位置”，提供的目标位置能精确到0.1度。首先，在功能本体中应包含“获取目标位置”这一功能的定义，并且定义了相应的“精度”属性，用户描述必须遵循本体，不能使用本体中未定义的概念。功能本体中的相关描述如下：

```

<owl:DatatypeProperty rdf:ID="精度">
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#获取目标位置"/>
</owl:DatatypeProperty>

```

在这个服务的描述文件 Service1.owl 文件中，应增加功能参数“获取目标位置”，再进一步对此参数进行限制，限制条件应写在服务的 Precondition 中。示例代码如下(其中的规则描述采用WSRL语言)：

```

<profile:hasFunction rdf:resource="#_FUNCTION1"/>
.....
<process:Function rdf:ID="_FUNCTION1">
  <process:parameterType rdf:datatype="
    http://www.w3.org/2001/XMLSchema#anyURI">
    http://127.0.0.1/ontology/funOnto.owl
    #获取目标位置</process:parameterType>
  <rdfs:label></rdfs:label>
</process:Function>
.....
<process:hasPrecondition>
  <expr:SWRL-Condition >
  <expr:expressionLanguage rdf:resource="
    http://www.daml.org/services/owl-s/1.1/
    generic/Expression.owl#SWRL"/>
  <expr:expressionBody rdf:parseType="Literal">
  <swrl:AtomList>

```

```

<rdf:first>
  <swrl:DatavaluedPropertyAtom>
    <swrl:propertyPredicate
      rdf:resource=" &funOnto; #精度">
    <swrl:argument1
      rdf:resource=" #_FUNCTION1"/>
    <swrl:argument2
      rdf:datatype=" http://www.w3.org/2001/XMLSchema#
        float " >0.01</swrl:argument2>
    </swrl:DatavaluedPropertyAtom >
  </rdf:first>
<rdf:rest rdf:resource="
  "http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</swrl:AtomList>
</expr:expressionBody>
</expr:SWRL-Condition>
</process:hasPrecondition>

```

至此,只要采用标准的本体解析器,就能准确解析并获取服务的功能描述。

5 基于功能本体的服务映射及预检索方法

为了快速、简便地实现基于功能语义进行服务发现,本文提出了一种功能语义映射机制,即利用领域功能本体定义的知识,让服务提供者和使用者的功能描述具有共同的语义理解,避免语义冲突。功能语义映射主要是利用领域功能本体定义的概念及概念间的关系,完全地或部分地替换服务(请求)功能语义描述所使用的相关术语和概念,从而生成一种有利于计算机可自动识别的服务功能描述。

服务(请求)基于功能语义的基本映射过程主要包括功能语义提取与映射两个阶段:

(1)功能语义提取。依据给定的服务功能本体描述文件,采用语义本体解析器对服务的描述文档进行解析,从中抽取出服务的功能语义描述信息。

(2)功能语义映射。即将复合服务分解成原子服务的过程。

如某用户在网内发布服务 A“提供我方兵力布属信息服务”,经服务映射分解后,则有:

$A = \langle \langle \text{人员编配}, X \rangle, \langle \text{武器装备}, Y \rangle, \langle \text{分布位置}, Z \rangle, M \rangle$,其中字母代表对各服务的限制,此处不做详细描述。从映射的过程也可以看出,如果我们不考虑对各原子服务的限制描述,其映射过程也可看成是各概念基于语义的转换过程。

根据定义 1,从语义的角度,可把所生成的映射原子服务集合看成两部分:第一部分是各原子服务功能的基本含义的描述,即它可以完成的功能是什么。第二部分是对这些功能的限制。同样,在匹配时,为了简化匹配计算量,也可将基于功能的匹配操作分作两步,首先完成功能的匹配,即粗粒度地滤除功能上肯定无关的服务,然后,根据功能的限制条件做进一步的筛选。功能服务本体给出了本领域所有相关原子服务的描述,且依据定义 1,各原子服务概念间是相互独立的。不失一般性,我们可以把本领域的原子服务概念(不包括限制条件)采用向量的方式表示。

设服务功能本体原子服务概念全集 $U = \{F_1, F_2, F_3, \dots, F_n\}$,则有

$$F_{complex} = \bigcup F_{atom} = \begin{cases} \langle u_1, u_2, u_3, \dots, u_n \rangle \\ u_i = \begin{cases} 1, & \text{iff } F_i \in F_{complex} \\ 0, & \text{other} \end{cases} \end{cases} \quad (5)$$

同时为了消除表示的二义性,我们规定各原子服务概念是可偏序排序的,其排序原则是在本体树中从左至右出现的先后顺序。如图 2,本体中共包括 8 个原子服务,如某用户在网内发布服务 A“提供我方兵力布属信息服务”且只考虑其服务的功能,则可表示为 $\langle 1, 1, 1, 0, 0, 0, 0, 0 \rangle$ 。至此,借助本体,把服务功能描述转化成了一个简单的向量形式。

定义 4(基于功能向量的服务匹配度) 服务与请求基于功能向量的匹配度定义为 $M(S, Q) = \frac{|S \cap Q|}{|Q|}$ 。其中 $|S \cap Q|$ 表示服务 S 与请求 Q 之间功能向量对应位相同的个数。 $|Q|$ 则表示请求所包括的所有原子服务的个数。

定义 5(基于功能向量的服务相似度) 服务 S 与请求 Q 之间的相似度定义为 $L(S, Q) = \frac{|S \cap Q|}{|S \cup Q|}$ 。其中 $|S \cap Q|$ 表示服务 S 与请求 Q 之间功能向量对应位相同的个数。 $|S \cup Q|$ 则表示二者所包括的所有原子服务的个数。

基于功能向量的服务预检索算法的伪代码描述如图 4 所示。

```

Input: OWL-S based Web service request
Output: Set of functional match services
Begin:
  1. Analysis the request and extract the functional conceptions
  2. Map the conceptions into vector according to functional ontology
  3. for i=1 to M
      scan each service in repository.
      if  $M(S_i, Q) = 1$ , then put  $S_i$  into Set
  4. Sort the Set according to  $L(S_i, Q)$ .
  5. for i=1 to |Set|
      if the service function para meet the request
        put the service int to Select Set
end

```

图 4 基于功能向量的服务预检索算法

当服务器收到用户发来的服务请求时,首先通过标准的本体解析器提取请求中的功能请求描述及限制条件,并依据功能本体将服务请求映射成功能向量。然后,扫描本地存贮的服务,如果 $M(S, Q) = 1$,则说明当前服务可以满足用户的请求,并放入候选集合。当扫描完毕后,对集合中的服务按 $L(S, Q)$ 排序,以保证排在前面的服务其功能与请求最接近。接下来采用精确匹配算法实现限制条件的匹配。二次扫描完毕后,即完成本地所存贮服务基于功能语义的预检索。

根据上述对算法运行过程的描述,只要领域功能本体设计合理,且不考虑在功能限制条件精确匹配阶段引入的匹配误差,则本算法滤除无关服务的效果应是接近 100% 的。因此,本文对此未作仿真验证。

5.1 服务概念排序方法及仿真分析

提取服务描述(请求)中的信息,并转化为向量的表示方法,虽然在服务查找的初期,可明显减少精确匹配引入的计算量,但是,如果本体树中各原子服务排列得不合理,则无法很好地体现原子服务功能概念间的相关度。如图 2 所示,希望其原子功能服务概念的排列顺序是:

〈人员编配、武器配置、分布位置、兵力估计、火力分布、水文、气象、道路〉

而不是其它顺序,如:

〈分布位置、人员编配、水文、武器配置、气象、道路、兵力估计、火力分布〉

即最优的排列顺序应是:如果两个概念的相关度越大,则二者在向量中排列的顺序越近。如果服务提供者所提供的重多服务具有很强的聚类特点,则按照最优排序生成的向量中“1”的分布情况,即可体现类似服务的聚类特征,这也便于进一步优化服务查找算法。

对于任意给定的本体关系树,如何更好地实现各原子概念间按相关度的排序?一种方法是采用现有聚类算法,如经典的 k-means 算法或基于神经网络的服务聚类算法。但是这些算法普遍存在计算量大、限制条件多(如需要预先指定分类的数目或需要大量的训练序列)的问题。下面给出了一种采用递归下降的扫描排序算法,其只依据给定的本体关系树即可生成相对合理的排列顺序。首先给出算法的排序规则,然后给出算法的描述。

(1) 排序规则

本体树中的概念分类经常是模糊的,一个概念可能有多个父概念,本体树中经常出现交叉的树枝。综合考虑本体树中各概念间的关联关系,并参考服务匹配算法中常用的距离、深度等判别原则,制定排序规则如下:

a)若两个概念位于同一层,且为兄弟概念,即具有相同的直接父概念,则两个概念的位置靠近,当概念具有多个直接父概念时,相同父概念的个数越多则位置越近。

b)若两个概念位于同层不是兄弟概念,即无相同直接父概念,则两个概念的父概念差异越小,即父概念的不同子概念越少,位置越靠近。

c)若概念 A 的父概念为 F, B₁ 为 A 的兄弟,而 B₂ 为 F 的兄弟,则 B₁ 比 B₂ 更靠近 A。

d)若概念 A 与 B₁ 最近共同祖先为 F₁,与 B₂ 最近共同祖先为 F₂,F₁ 比 F₂ 更接近底层,则 B₁ 比 B₂ 更靠近。

依据以下判别条件,现给出算法的过程描述如下。

(2) 算法描述

对于给定的服务功能本体树,

a)自顶向下调整本体树,置顶层概念的序号 $i=1$ 。

b)对于第 w 层的概念 C,若存在父概念 C_f,其层次 $w_f < w-1$,则将 C_f 的序号 i 转换为 $(i - \frac{1}{2}) \cdot \frac{n_{w-1}}{n_f}$,其中 n_{w-1} 为第 $w-1$ 层的概念个数, n_f 为第 w_f 层的概念个数。C 的序号为转换后的所有父概念序号的平均值。

c)第 w 层的概念序号全部得到后,按从小到大的顺序排列。若 C₁, C₂, ..., C_k 的序号相同,则选出子概念 C_j,使 C_j 的直接子概念最多,排在最前面,并将其所有子概念放入已选择子概念集合 S,再选出概念 C_i,使 C_i 中的直接子概念与 S 差别最小,将其排在 C_j 后面,并将 S 更新为 SUC_i,其中 C_i 表示 C_i 的所有子概念。按此规则直至得到 k 个概念的排列顺序。

d)更新第 w 层的概念序号,使其为从 1 开始依次递增的整数。

e)整个本体树排列完成后,取最底层的叶子节点按概念

序号排成一列。

f)自底向上逐层扫描,添加不在最底层的叶子节点。对于不在最底层的叶子节点 P,向上找到一个最近的父节点 Q,使 Q 具有已在序列中的叶子节点,将 P 添加在 Q 所在序列中最后一个叶子节点的后面。

(3) 性能分析

本节采用 matlab 作为仿真工具对本体树调整算法的性能进行了仿真,仿真主要包括两个方面:一是本体树中各原子概念调整前后,其原子服务间关系相似性关系体现的变化;二是本体树中各原子服务的排列顺序经过调整后,其功能向量的表示方式所显示的服务间关系的变化。在仿真时,各概念间的关系及相对位置采用矩阵的方式保存,文中为了表述方便,采用图形化的方式表示仿真参数及结果。图 5 为本体树各原子服务概念调整前示意图,图 6 为调整后示意图。图中复合概念编号采用“x.y”的方式表示,其中“x”表示层数,“y”表示在层中的序号。

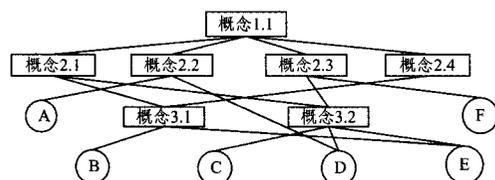


图 5 本体关系树调整前示意图

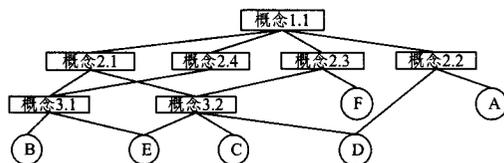


图 6 本体关系树调整后示意图

调整前,各原子服务的排列顺序为〈A, B, C, D, E, F〉,调整后,各原子服务的排列顺序为〈B, E, C, D, F, A〉,由图可以直接看出,原子概念 E 同时属于复合概念(3.1)及(3.2),原子概念 F, A 属于第 3 层的概念。其调整后的排列顺序要比调整前合理得多,且采用常用的概念距离及深度相似度计算方法也可得出相同的结果。

同时在仿真中也发现,如果复合概念与原子概念间出现复杂的多对多关系,即从语义的角度看,各原子(复合)概念间的相互独立性差时,采用本算法并不能产生很好的排列顺序。如果出现这种情况,则与本文定义 1 中给出的前提条件(各原子服务概念间是相互独立的)差别比较大,说明功能本体设计得不合理。此时,根据实际情况采用逐层分解的方式,把同层的概念用集合的方式表示,然后,将这些相交的集合分解成合取范式来修正本体的设计,就可较好地解决这一问题。

依据图 5 及图 6 给出的原子服务间排列顺序,表 1 给出了各复合服务的向量表示。

表 1 向量表示对比

概念编号	调整前向量表示	调整后向量表示
2.1	〈0,1,1,1,1,0〉	〈1,1,1,1,0,0〉
2.4	〈0,1,0,0,1,0〉	〈1,1,0,0,0,0〉
2.3	〈0,0,1,1,1,1〉	〈0,1,1,1,1,0〉
2.2	〈1,0,0,1,0,0〉	〈0,0,0,1,0,1〉
3.1	〈0,1,0,0,1,0〉	〈1,1,0,0,0,0〉
3.2	〈0,0,1,1,1,0〉	〈0,1,1,1,0,0〉

从原子服务顺序调整前后的表示情况看,处于同层的复合服务,其向量表示与调整前相比,“1”的排列情况显示出了较好的聚合特性。在特定的服务领域,如果用户所发布的服务具有较好的聚类特性,则从每个服务向量中“1”的分布情况,即可粗略地估算出服务间功能的相似程度,并初步实现服务基于功能的聚类。本文由于表述问题,采用的本体比较简单,在本体中所包含的原子服务更多的情况下,这种聚合特性将表现得更加明显。

结束语 随着 Web 服务数量的急剧增加和应用的日益普及,如何快速、准确和高效地发现满足用户需求的服务已经成为制约 Web 服务发展的瓶颈之一。从用户查找服务的实际操作过程出发,采用显式地描述服务功能语义、有效地滤除与请求功能无关的 Web 服务是简化服务检索计算量、提高服务发现准确率的有效方法之一。本文针对现有常用 Web 服务描述语言(如 OWL-S),不能显式地描述服务功能语义的问题,通过定义服务功能描述模型,构建领域功能本体,提出了一种基于功能语义的 Web 服务描述方法,并对 OWL-S 语言进行扩充,在此基础上,为简化服务滤除操作,设计了一种服务映射方法及原子服务排序算法,使生成的服务向量能更好地反映原子服务间的相关特性,最后给出了一种服务预检索方法。分析表明,该方法可有效地支持用户实现基于功能语义服务检索,滤除无关服务,降低精确匹配计算量,提高服务查找效率。同时由于本描述方法是通过扩充 OWL-S 语言实现的,可以更好地继承现有基于 OWL-S 语义服务描述语言获取的研究成果。

参 考 文 献

[1] D'Mello D A, Ananthanarayana V S. A Review of Dynamic Web Service Description and Discovery Techniques[C]//2010 First

International Conference on Integrated Intelligent Computing, 2010;246-251

- [2] D'Mello D A, Ananthanarayana V S. A Tree Structure for Efficient Web Service Discovery[C]//Second International Conference on Emerging Trends in Engineering and Technology, 2009;826-831
- [3] Paliwal A V, Shafiq B, Vaidya J, et al. Semantics-Based Automated Service Discovery [J]. Services Computing, IEEE Transactions on, 2012, 5(2);260-275
- [4] 于晓浩, 罗雪山, 胡丹, 等. 基于行为约束的军事信息服务描述及过滤方法[J]. 国防科技大学学报, 2011, 33(4);157-162
- [5] 万长林, 史忠植, 胡宏, 等. 基于本体的语义 Web 服务 QoS 描述和发现[J]. 计算机研究与发展, 2011, 48(6);1059-1066
- [6] Jean S, Losavio F, Matteot A, et al. An Extension of OWL-S with Quality Standards[C]//Fourth International Conference on Research Challenges in Information Science, IEEE, 2010;483-494
- [7] Yousefipour A, Ne A G, Mohsenzadeh M, et al. An Ontology-based Approach for Ranking Suggested Semantic Web Services [C]//Sixth International Conference on Advanced Information Management and Service, 2010;17-23
- [8] D'Mello D A, Ananthanarayana V S. Effective Web Service Discovery Based on Functional Semantics[C]//International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009;1-3
- [9] Ye Lei, Zhang Bin. Web Service Discovery Based on Functional Semantics[C]//Second International Conference on Semantics, Knowledge, and Grid, 2006;57-60
- [10] Amorim R, Claro D B, Lopes D, et al. Improving Web service discovery by a functional and structural approach[C]//International Conference on Web Services, 2011;411-418

(上接第 210 页)

图 4 描述了最小支持度为 0.4 时,3 种算法的运行时间对比结果。分析时间结果能够得出,同关联规则算法、模糊关联规则算法对比,随着数据库规模的不断增加,本文算法的运行时间越来越小于关联规则算法,具有较高的运行效率,更加适合于大型数据库的挖掘。

结束语 本文提出了基于特征模糊贴近的数据库约束挖掘算法,其通过数据模糊集间的贴程度描述数据间的一致度,采用神经网络数据融合技术对数据进行一致性处理后获取初始挖掘数据,对原始挖掘数据的动态特征进行分析获取新的评估模型,以实现在大规模数据库中,准确查询相应的目标数据。仿真实验结果表明,本文提出的算法挖掘稀疏数据集和密集数据集的效率都优于传统的关联规则算法,极大提高了数据库的挖掘效率,具有重要的应用价值。

参 考 文 献

[1] 何娟,涂中英,牛玉刚.一种遗传蚁群算法的机器人路径规划方法[J]. 计算机仿真, 2010, 27(3);170-175

- [2] Abiteboul S, Bnueman P, Suci D. Data on the Web[Z]. Morgan Kaufmann Publishers, 2010;89-93
- [3] 丁艳辉,王洪国,高明,等.一种基于矩阵的关联规则挖掘新算法[J]. 计算机科学, 2006, 33(4);188-189
- [4] 郑泉,王建设.基于 FP 树挖掘大数据的方法及算法 PCM[J]. 计算机工程与应用, 2009, 40(7);182-184
- [5] Kitts B, Freed D, Vrieze M. Cross-sell; A Fast Promotion Tunable Customer-Item Recommendation Method Based on Conditionally Independent Probabilities[C]//Proceedings of KDD. Based on ACM Press, 2010;437-446
- [6] 胡刚,刘建鑫,郭炜.基于 PIC18F4550 的环境监测数据采集系统设计[J]. 科技通报, 2012, 10(28);17-18
- [7] Gauch S, Chaffee J. Ontology-based personalized search and browsing[J]. Web Intelligence and Agent Systems, 2011, 1(3);77-79
- [8] 赵国锋,李兵,徐川,等.移动社交网的生命周期评估模型研究[J]. 计算机学报, 2013(4);727-737
- [9] 赵妍妍,秦兵,刘挺.文本情感分析[J]. 软件学报, 2012, 21(8);1834-1848