

一种基于关联匹配度计算的语义 Web 服务匹配方法*

A Matching Method for Semantic Web Services Based on Association Matching Degree Calculation

唐海燕^{1,2}, 陈宁江¹

TANG Hai-yan^{1,2}, CHEN Ning-jiang¹

(1. 广西大学计算机与电子信息学院, 广西南宁 530004; 2. 广西交通运输学校, 广西南宁 530007)

(1. College of Computer, Electronic and Information, Guangxi University, Nanning, Guangxi, 530004, China; 2. Guangxi Transportation School, Nanning, Guangxi, 530007, China)

摘要:在现有的语义 Web 服务匹配方法基础上加入服务语义关联匹配度计算,从而提出一种改进的语义服务匹配方法。该方法通过语义关联将一对一匹配淘汰的服务进行组合来实现一对多的服务匹配,在得到的候选服务和候选服务集中进一步计算 Web 服务的匹配度,经过匹配度的排序,得到最佳匹配度的服务。该方法避免了仅从单一接口参数上进行语义服务匹配,能在一定程度上提高匹配的效率和实现查准率、查全率和查询效率之间的有效平衡。

关键词: Web 服务 服务匹配 关联匹配度 语义

中图分类号: TP311 **文献标识码:** A **文章编号:** 1002-7378(2012)01-0023-05

Abstract: An improved semantic Web services matching method is proposed, which introduces the calculation of association matching degree into the existing common service matching process. It uses the semantic association to realize the one to multiple matchings through composing the one to one matchings eliminated services. Further, the calculation of association matching degree of services is performed on the candidate services set. Then the best matched service is obtained by ordering the matching degree. It can avoid the shortcomings of service matching on parameters from single interface and be helpful for improving the effect of matching as well as the tradeoff among the precision rate, recall and query efficiency at certain extent.

Key words: Web services, service matching, association matching degree, semantic

在服务计算环境下,随着可用的服务数量的增长,如何快速准确地找到满足用户需求的服务已成为 Web 服务研究领域的关键问题。传统的 Web 服务基础架构在面向复杂需求的服务选择上存在局限

性。例如,早期 UDDI 上的服务发现是基于关键词的查询来对 Web 服务进行匹配的,由于缺乏语义信息的支持,无法区分关键词“同名不同义”、“同义不同名”等情况。为了改进这种技术,人们使用语义技术来支持服务的多维搜索标准,利用丰富的语义描述能力和强大的逻辑推理能力来准确地描述 Web 服务的语义,从而产生了语义 Web 服务^[1]。而语义 Web 服务匹配^[2]是语义 Web 服务研究中的基本问题,它的作用是从已经发布的服务中发现与用户需求最匹配的服务,同时也是服务需求方与服务提供方进行服务语义匹配的过程。语义 Web 服务匹配涉及的主要问题有 Web 服务的语义描述、基于语义

收稿日期:2011-12-15

修回日期:2011-12-29

作者简介:唐海燕(1976-),女,硕士,讲师,主要从事服务计算研究。
陈宁江(1975-),男,博士,教授,主要从事软件工程、网络分布计算研究。

* 国家科技支撑计划课题(编号:2009BAH53B03),国家自然科学基金项目(编号:61063012),广西教育厅科研项目(编号:桂教科研[2010]10号)资助。

的 Web 服务匹配算法等。

目前语义 Web 服务匹配算法主要有两类:(1) 基于词汇相似度计算的匹配算法^[2,3]。该算法的原理是利用信息检索 IR 技术,计算不同语义 Web 服务的各项功能和性能描述之间的语义距离,进而判断它们的语义匹配度。由于仅从单一接口参数上进行语义服务匹配,该方法往往只能给出单个服务接口与服务请求之间的匹配程度。(2) 基于逻辑推理的匹配算法^[4,5]。该方法的基础是描述逻辑(DL),算法通过 DL 推理机的逻辑推理能力来判断不同语义 Web 服务之间描述的匹配程度。它把 Web 服务看做一个动作,其执行能导致状态空间和数据空间的改变。此种方法不仅进行一对一的服务匹配,还进行一对多的服务匹配,即组合一组服务来满足一个服务请求。在这两种方法中,后者的查准率和查全率比前者高,但是查询效率很低;前者虽然胜在查询效率,但是查准率和查全率较低。当前的许多语义服务匹配方法都仅停留在一对一的服务匹配上^[6,7],放弃了许多潜在的匹配服务。而复杂的一对多的服务匹配不仅需要定义组合规则而且需要创建过程模型^[8,9]。

为了克服这些局限,本文通过使用服务描述中的语义信息,提出服务之间语义关联的概念。通过语义关联将一对一匹配淘汰的服务进行组合来实现一对多的服务匹配,在得到的候选服务和候选服务集中进一步计算 Web 服务的匹配度,经过匹配度的排序,得到最佳匹配度的服务。

1 语义关联的相关概念

服务的语义反映服务的功能,而语义关联反映不同服务功能之间的关系。为了简化,在文中假设服务需求方与服务提供方在一个领域中使用相同的本体。在系统中,对本体的定义采用一种面向对象的方法,其中类型定义为类(用 C 表示),包含名称、同义词、属性、父类。类的实例定义为对象,用 o 表示,包含类名、类的属性值。

定义 1 服务提供方提供的服务称作服务提供接口,表示为 $IP_i(P_I, P_O)$, 其中, $P_I = \langle C_1, C_2, \dots, C_n \rangle$ 表示输入参数类列表, $P_O = \langle C_1, C_2, \dots, C_m \rangle$ 表示输出参数类列表。

定义 2 服务请求表示为 $R(O_I, E_O)$, 此处 $O_I = \langle o_1, o_2, \dots, o_n \rangle$ 表示服务请求提供的输入对象集合, $E_O = \langle C_1, C_2, \dots, C_m \rangle$ 表示期望从服务提供接口得到的输出参数列表。

定义 3 如果在本体中,类 C_i 和类 C_j 的名称等价,或者是 C_i 继承 C_j , 那么称 C_i 与 C_j 之间具有服务语义关联,记为 $C_i \langle \Rightarrow \rangle C_j$ 。

对于类集合 SC , 如果 $\exists C_i \in SC, C \langle \Rightarrow \rangle C_i$, 则称类 C 语义属于类集 SC , 记为 $C \stackrel{S}{\Rightarrow} SC$ 。

在服务匹配问题中,一个候选服务提供接口是指能接受服务请求的输入对象、并且能提供与服务请求需要的输出语义关联的所有输出的服务提供接口。假设 $\Omega = \{IP_1, IP_2, \dots, IP_n\}$ 是一组服务提供接口的集合, $SC_{P_{I_i}}$ 是由接口 IP_i 的所有输入参数类组成的类集, $SC_{P_{O_i}}$ 是由接口 IP_i 的所有输出参数类组成的类集; R 是服务请求, SC_{O_I} 是由输入对象类组成的类集, SC_{E_O} 是由期望的输出参数类组成的类集; o_R 是用于作为关联关键词的输入对象,称之为关联对象。当满足以下 3 条规则时,则 Ω 是 R 的一个关联服务接口集。

规则 1 R 能提供每一个接口需要的所有输入,即:对于 Ω 中所有输入参数类集的集合 $SC_{P_I} = \{SC_{P_{I_1}}, SC_{P_{I_2}}, \dots, SC_{P_{I_i}}, \dots, SC_{P_{I_n}} \mid 1 \leq i \leq n, i \in Z\}$, 有 $\forall C \in SC_{P_I}, C \stackrel{S}{\Rightarrow} SC_{O_I}$ 。

规则 2 R 期望的所有输出都能被提供,即:对于 Ω 中所有输出参数类集的集合 $SC_{P_O} = \{SC_{P_{O_1}}, SC_{P_{O_2}}, \dots, SC_{P_{O_i}}, \dots, SC_{P_{O_n}} \mid 1 \leq i \leq n, i \in Z\}$, 有 $\forall C \in SC_{E_O}, C \stackrel{S}{\Rightarrow} SC_{P_O}$ 。

规则 3 如果 Ω 中所有的接口均含有 o_R 作为输入参数(也就是说它们是可关联的),并且每个接口提供至少一个唯一的输出,即:对于 $o_R \in C_R$, 有 $\forall SC_{P_{I_i}}, \exists C_R^S, C_R \langle \Rightarrow \rangle C_R^S$, 同时, $\forall SC_{P_{O_i}}, \exists C, C \stackrel{S}{\Rightarrow} SC_{E_O} \wedge C \notin (\cup_{j=1}^{i-1} SC_{P_{O_j}} \cup \cup_{j=i+1}^n SC_{P_{O_j}})$, 其中 \cup 表示集合的并运算。

当关联一组服务提供接口来实现一个服务请求时,以上 3 个规则都需要被满足。

2 基于语义关联匹配度计算的服务匹配方法

方法的基本思想是通过在服务代理中使用本体,除了单一接口的匹配外,扩展服务匹配算法,加入服务提供接口相关的语义信息,然后定义它们之间的关系,通过描述服务的语义关联,来实现一对多的服务匹配,并对两种匹配的筛选结果计算关联匹配度,从而进一步优化匹配结果。匹配过程包括如

下 4 个阶段:

(I) 根据传统的精确匹配方法,找出与服务请求有完全相同的输入输出参数类的服务提供接口;

(II) 找出与服务请求参数类语义关联的服务提供接口。

(III) 搜索一组服务提供接口,这组接口能接受服务请求的输入对象并且能被关联起来提供服务请求期望的输出。如果共有一些输入参数和不同的输出参数,并且其共有的输入参数是服务请求指定的关联对象,那么多个服务提供接口就可以关联成一个接口。但是并不是所有能关联的接口都可做为关联接口集满足服务请求,因此,当执行关联匹配以组合一个能提供服务请求需要的所有输出的服务接口时,首先要搜索可以关联的服务接口,也就是具有由服务请求指定的关键输入参数的一组服务接口,然后在能关联的服务接口中找出能提供服务请求需要的所有输出的关联服务接口集。

(IV) 计算服务关联匹配度。借助 OWL-s 中的概念 IOPE^[10] 来进行匹配度计算。IO 为 Input/Output,描述服务执行前后数据空间的变化,PE 是 Precondition/Effect,描述服务执行前后状态空间的变化。我们通过输入、输出参数来进行匹配度计算。匹配度由高到低分为 4 个等级:Exact、PlugIn、Subsumes、Fail。Exact 表示期望的参数与提供的参数完全相同或前者是后者的直接子类;PlugIn 表示期望的参数被提供的参数包含但不是提供的参数的直接子类;Subsumes 表示期望的参数包含提供的参数;3 种情况以外的即为 Fail。

2.1 Web 服务单个参数的匹配度计算

根据上述的匹配度定义,对服务请求中的每一个输出,算出其与服务提供接口所有的输出参数两两相匹配后最佳的匹配结果,然后把服务请求中所有输出参数中匹配度最低的作为服务请求与服务提供接口输出参数的匹配度。用伪代码描述如下:

```
outputMatch(Expect_out, Publish_out) {
    globalDegreeMatch = Exact;
    forall Eo in Expect_out do {
        find Po in Publish_out such that
            degreeMatch = max(DegreeMatch(Eo, Po))
            if (degreeMatch = fail) return fail;
            if (degreeMatch < globalDegreeMatch)
                globalDegreeMatch = degreeMatch;
    }
    return globalDegreeMatch;
}
```

}

同理,对服务提供接口中的每一个输入,算出其与服务请求所有的输入参数两两相匹配后最佳的匹配结果,然后把服务提供接口中所有输入参数中匹配度最低的作为服务提供接口与服务请求输入参数的匹配度。

2.2 整个 Web 服务匹配度排序

输出参数匹配度越高其整个 Web 服务匹配度越高,如果输出参数匹配度相等,那么就由输入参数的匹配度高低来决定整个 Web 服务匹配度的高低。

//用函数 MaxDegreeMatch() 求出请求 R 的最佳匹配接口。

```
MaxDegreeMatch(R) {
    Foreach Interface in canList do {
        //canList 表示服务请求 R 的候选服务提供接口清单
        call outputMatch(Expect_out, Publish_out);
        call inputMatch(Publish_in, Object_in);
    }
    MaxDegreeMatch = IP1
    Foreach Interface in canList do {
        If Interface.output > MaxDegreeMatch.output
            then MaxDegreeMatch = Interface;
        If Interface.output = MaxDegreeMatch.output
            & Interface.input > MaxDegreeMatch.input
            then MaxDegreeMatch = Interface;
    }
    Return MaxDegreeMatch;
}
```

在以上过程中,除了通过单一接口的精确匹配和语义匹配得到 Exact、PlugIn 匹配的服务,还考虑通过语义关联体现服务功能之间的关系,把多个功能单一的接口关联起来满足一个服务请求,并对 3 个阶段的匹配结果计算匹配度然后排序。我们的方法可以有效地利用 Subsume、Fail 匹配的服务,在分工越来越细的社会里,各企业各部门提供的服务功能愈加单一,但是,随着 Web 服务应用的迅速普及,用户会对 Web 服务的功能提出更高的要求,这种情况下当期望的服务提供接口不存在或在运行时变得不可用时,服务代理可以把多个单一的服务关联起来形成一个“虚拟服务”来满足服务请求。

3 实例分析

当用户提出一个服务请求,服务代理接受用户

的请求并为其服务。服务代理告诉用户有哪些服务可以被调用。

匹配过程:

服务请求: $R_1(O_I\langle\text{二手房}\rangle, E_O\langle\text{二手房贷款, 二手房评估}\rangle)$

已知可用的服务集合:

$IP_1(P_I\langle\text{房产}\rangle, P_O\langle\text{房产贷款, 房产评估}\rangle)$,

$IP_2(P_I\langle\text{房产}\rangle, P_O\langle\text{房产贷款}\rangle)$,

$IP_3(P_I\langle\text{房产}\rangle, P_O\langle\text{房产装修}\rangle)$,

$IP_4(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房贷款}\rangle)$,

$IP_5(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房评估}\rangle)$,

$IP_6(P_I\langle\text{旅行}\rangle, P_O\langle\text{旅行路线}\rangle)$,

$IP_7(P_I\langle\text{旅行}\rangle, P_O\langle\text{酒店预订}\rangle)$ 。

候选服务提供接口:

$IP_1(P_I\langle\text{房产}\rangle, P_O\langle\text{房产贷款, 房产评估}\rangle)$ 。

可关联的接口:

$\Omega_1 = \{IP_2, IP_3\}$, 其中 $IP_2(P_I\langle\text{房产}\rangle, P_O\langle\text{房产贷款}\rangle)$ 、 $IP_3(P_I\langle\text{房产}\rangle, P_O\langle\text{房产装修}\rangle)$,

$\Omega_2 = \{IP_4, IP_5\}$, 其中 $IP_4(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房贷款}\rangle)$ 、 $IP_5(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房评估}\rangle)$ 。

关联接口集:

$\Omega_2 = \{IP_4, IP_5\}$, 其中 $IP_4(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房贷款}\rangle)$ 、 $IP_5(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房评估}\rangle)$ 。

输出匹配度最高接口:

$\Omega_2 = \{IP_4, IP_5\}$ 。

对于服务请求 R_1 , 在第一阶段: 由于没有与服务请求有完全相同的输入输出参数类的服务提供接口, 故得不到匹配的服务提供接口, 而在第二阶段, 接口 IP_1 能被服务请求调用, 因为输入对象“二手房”与“房产”语义关联。同时, IP_1 能提供 R_1 需要的所有输出, 因为二者的输出参数也是语义关联的。所以, IP_1 是 R_1 的一个候选服务接口。

第三阶段关联匹配, 找出关联接口集。根据关联接口集的定义, $IP_2(P_I\langle\text{房产}\rangle, P_O\langle\text{房产贷款}\rangle)$ 和 $IP_3(P_I\langle\text{房产}\rangle, P_O\langle\text{房产装修}\rangle)$ 可以被关联, 因为他们都有输入参数“房产”, 与 R_1 的输入对象语义关联; 同理, $IP_4(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房贷款}\rangle)$ 和 $IP_5(P_I\langle\text{二手房}\rangle, P_O\langle\text{二手房评估}\rangle)$ 可以被关联。而 $IP_6(P_I\langle\text{旅行}\rangle, P_O\langle\text{旅行路线}\rangle)$ 和 $IP_7(P_I\langle\text{旅行}\rangle, P_O\langle\text{酒店预订}\rangle)$ 虽然共有输入参数“旅行”, 但它并不是服务请求指定的关联对象, 所以是不可关联的。接着, 从可关联的接口中找出关联接口集, IP_2 和 IP_3 的集合 Ω_1 提供的输出是“房产贷款”和“房产装修”, 这已不满足规则 2, 所以 Ω_1 不能

做为关联接口集。 IP_4 和 IP_5 的集合 Ω_2 的输入能由服务请求提供, 这满足了规则 1; 并且它们能提供服务请求需要的所有输出, 这满足了规则 2。这两个接口的输入参数都为“二手房”, 服务请求 R_1 的关联对象; 同时, IP_4 和 IP_5 分别提供唯一的输出“二手房贷款”和“二手房评估”, 这满足了规则 3。因此, Ω_2 可做为服务请求的一个关联服务接口集。

到目前为止, 对于服务请求 R_1 已得到候选服务提供接口 IP_1 和关联服务接口集 $\Omega_2 = \{IP_4, IP_5\}$, 接下来服务匹配方法将进行第 4 阶段匹配度计算, 从候选服务提供接口和关联服务接口集中选择匹配度最高的一个服务提供接口来满足服务请求。

根据上面的计算规则, IP_1 与 R_1 的输出参数匹配度为

$\text{globalDegreeMatch} = \min(\max(\text{Exact}, \text{Fail}), \max(\text{Fail}, \text{Exact})) = \min(\text{Exact}, \text{Exact}) = \text{Exact}$ 。

显然, Ω_2 与 R_1 的输出参数匹配度也为 Exact 。所以, 接下来继续计算输入参数匹配度, IP_1 与 R_1 的输入参数匹配度为 Subsumes , 而 Ω_2 与 R_1 的输入参数匹配度为 Exact , 所以, Ω_2 的匹配度高于 IP_1 , 至此, 得到 R_1 的最佳匹配接口为 Ω_2 , 等同于精确匹配。

值得注意的是, 大多数服务匹配方法只对单一接口参数直接进行第 4 个步骤计算服务匹配度来查找, 那么会放弃掉 IP_4 、 IP_5 这样的潜在匹配服务, 而选择了 IP_1 , 因为计算的结果是 IP_2 、 IP_3 、 IP_4 、 IP_5 、 IP_6 、 IP_7 与 R_1 的输出参数匹配度均为 Fail , 这样导致选择了次优的服务。如果只进行到第 3 个步骤, 那么就无法进一步确定候选服务提供接口与关联服务接口集的匹配度高低, 而随机选择一个服务。同时, 经过前 3 个步骤的筛选, 第 4 个步骤只对候选服务提供接口和关联服务接口集进行匹配度的计算, 使得计算时间缩短, 提高了查询效率, 也提高了查全率和查准率。

4 结束语

目前大多数语义 Web 服务匹配方法仅从单一接口参数上进行语义服务匹配, 放弃了许多潜在的匹配服务, 导致较低的查全率和查准率。而能进行一对多服务匹配的方法却存在查询效率低、对同一等级的匹配结果无法进一步区分匹配程度的弊端。本文在现有的语义 Web 服务匹配方法基础上加入服务语义关联匹配度计算, 避免了仅从单一接口参数上进行语义服务匹配, 不仅能进行一对一的服务

匹配,还能进行一对多的服务匹配,把语义关联的多个服务组合起来满足一个服务请求,与传统的一对多匹配不同的是我们的方法无需创建过程模型,通过匹配度计算对匹配结果进行优化,在查准率、查全率和查询效率之间得到有效平衡。在后继的工作中,我们将通过更丰富的实验对该方法进行进一步验证,并对匹配度计算方法做深入探讨,考虑更丰富的服务语义关联优化。

参考文献:

- [1] Mcilraith S A, Son T C, Zeng H L. Semantic web services[J]. IEEE Intelligent Systems Special Issue on the Semantic Web, Mar-Apr, 2001, 16(2):46-53.
- [2] Fernandez A, Polleres A, Ossowski S. Towards fine-grained service matchmaking by using concept similarity[C]//Proc of the 1st Int Joint Workshop SMR2 on Service Matchmaking and Resource Retrieval, Berlin: Springer, 2007:31-45.
- [3] Gunay A, Yolum P. Structural and semantic similarity metrics for web service matchmaking[C]//Proc of EC-Web, Berlin: Springer, 2007:129-138.
- [4] Sirin E. Combining descriptin logic reasoning with AI matching for composition of Web service[D]. College

- Park; University of Maryland, 2006.
- [5] Benatallah B, Hacid M S, Leger A, et al. On automating Web service discovery [J]. The International Journal on Very Lange Data Bases, 2005, 14(1):84-96.
- [6] 李瑞,蒋翠清. 基于接口匹配的语义 Web 服务自动组合研究[J]. 情报理论与实践, 2009, 32(8):109-112.
- [7] 骆琴,张永胜,聂学武,等. 一种基于语义 Web 服务的匹配算法研究 [J]. 计算机技术与发展, 2011, 21(2):136-139.
- [8] Zeng L, Benatallah B, Ngu Ahh, et al. QoS-aware middleware for web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5):311 - 327.
- [9] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition [C]. WWW Track: Web Engineering / Session: Service Oriented Development, 2009:881-890.
- [10] Caragea D, Tanveer S M. Semantic API matching for automatic service composition may[C]. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, 2004:436-437.

(责任编辑:尹 闯)

(上第 15 页)

改进算法应用了分群思想,当标签的个数远远大于时隙数时,通过分群来大幅减小冲突的概率。所以改进的算法较好地解决了原 Type C 标准算法中分配的时隙数太多导致的响应概率呈指数降低的问题。

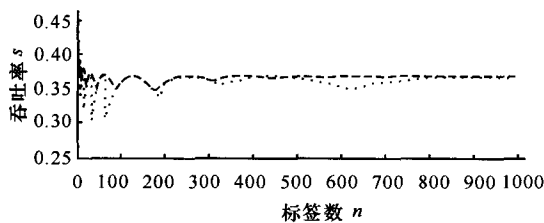


图 5 两个算法的吞吐率比较
·····:原算法;----:改进后的算法。

参考文献:

- [1] ISO/IEC. Information technology-radio frequency identification for item mangement-part 6: parameters for air interface communications at 860 MHz to 960MHz, amendment 1: extension with type C and update of

types A and B[S]. ISO/IEC 180002-6:2004/A md 1: (E), 2006.

- [2] EPCglobal Inc. EPCTM radio-frequency identity protocols class 1 generation-2 UHF RFID protocol for communications at 860-960MHz version 109[S]. EPCglobal IncTM, 2005.
- [3] 蒋泰,高广尚. ISO18000-6 Type C 标准中防冲突技术的改进[J]. 计算机工程与应用, 2010(20):42-04.
- [4] 陈家鼎,郑忠国. 概率与统计[M]. 北京:北京大学出版社, 2007.
- [5] 禹士朋, UHF 射频识别系统的防冲突算法研究[D]. 郑州:郑州大学, 2010.
- [6] XiaoFan, InChanSong, Kyung Hi Chang. Gen2-based hybrid tag anti-collision Q algorithm using chebyshev's inequality for passive RFID systems[EB/OL]. IEEE press, 2010. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4699508.

(责任编辑:尹 闯)