一种基于循环匹配的 Web 服务发现策略①

余 伟 陆 杰 (重庆大学 计算机学院 重庆 400044)

摘 要: 随着互联网的高速发展和广泛应用，互联网用户对服务的要求越来越高，同时开发人员对代码的重用和应用服务松耦合的要求也越来越高。Web 服务则提供了一种解决上述问题的方法，而传统的 Web 服务匹配主要是基于关键字的，缺乏对语义的支持，因而在服务匹配方面的效率较低。采用语义概念来表达 Web 服务的描述性内容,并针对概念之间的二义性提出了一种循环匹配的方法来计算概念之间的相似度，并最终来确定请求服务和广告服务的相似度。最后，实验证明本文提出的策略有较好的可行性和有效性。

关键词: Web 服务匹配；语义概念；循环匹配；本体；二义性

A Web Service Discovery Strategy Based on Cycle Matching

YU Wei，LU Jie (Computer College Chongqing University, Chongqing 400044, China)

Abstract: With the rapid development and extensive application of the Internet, Internet users have become increasingly demanding for services. At the same time, developers have also become increasingly demanding on code reuse and loose coupling of application services. Web services provide a solution to these problems. The traditional matching of Web services mainly based on keyword, lacks support for semantic. Thus, it has a less efficiency when used on Web service discovery. In this paper, semantic concept is first used to express the descriptive content of Web services. For the ambiguity between the concepts, a cycle matching method is then presented to calculate the similarity between the concepts, and to ultimately determine the similarity between requested service and advertising service. Finally, the proposed policy is proved to be feasible and effective via experiments.

Keywords: Web service matching; semantic concept; cycle matching; ontology; ambiguity

1 引言

Web 服务匹配是 Web 服务发现过程中一个重要步骤，匹配策略的优良程度直接影响 Web 服务发现的质量。目前，Web 服务发现中的匹配方法主要可以分为两种类型：

① 语法级服务匹配：语法级服务匹配策略中的服务是采用语法级的语言进行描述的，没有考虑服务语义方面的问题。该匹配策略是基于简单的分类和关键字进行匹配的，并通过计算服务描述文本中关键字的权重，来度量请求服务和广告服务之间的相似度。这种匹配策略既不能区分语法相同语义不同的情况，也无法区分语义相同语法不同的情况。

② 语义级服务匹配：语义级服务匹配主要基于本体描述的服务进行的，其增强了对 Web 服务的功能、行为的描述，在匹配时，可根据相应的本体描述进行合理的逻辑演绎和推理。通过语义描述的 Web 服务能够增强 Web 服务发现过程中的智能化水平，提高服务发现的查全率和查准率[1]。基于上述认识，本文采用语义概念来表达 Web 服务的描述性内容，并提出一种循环匹配的方法来解决语义概念之间的二义性，并最终来确定广告服务和请求服务的相似度。

1 语义Web服务描述语言(OWL-S)

OWL-S(Ontology Web Language for Services)

① 收稿时间:2010-01-30;收到修改稿时间:2010-03-11

是 web 服务的本体语言，主要是为了解决 Web 服务描述和发现以及组合的语义表示[2]。其包括三个组件见图 1：Service Profile：描述服务是做什么的；Service Model：描述服务是怎么做的；Service Grounding：描述怎样访问服务。

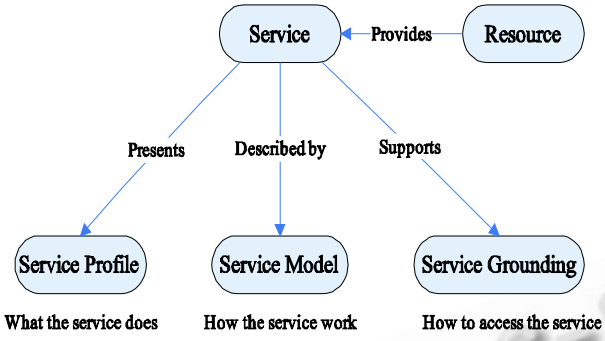

图 1　OWL-S 组织结构图

基于语义的 Web 服务匹配，主要是用到 Service Profile 的属性。Service Profile 则主要是从三个方面对服务进行了描述[3]：

服务的基本信息：包括服务名称(Service Name)、文本描述(Text Description)和连接信息(Contact Information)。在服务的描述中需要引入语义本体库[4]，而本体库中的概念存在着二义性的问题，因此本文也主要是对本体库中概念存在的二义性问题，提出一种有效的解决方法。

服务的功能描述：主要有输入(Inputs)、输出(Outputs)、前置条件(Precondition)、执行结果(Effects)，上述四个要素简称为 IOPE；

服务的特征描述：服务类别(Service Category)，运用本体论中的分类指定该服务所属的类；质量等级(Quality Rating)，用来描述服务质量。

2　本体概念及语义距离

2.1 本体概念间的二义性

本体库可看作一个存在上层和下层关系的层次结构。在这样的层次结构中除了继承的关系外，还存在着二义关系。如图 2 是一个存在二义关系的示例。即 Computer“控制”Mazda 。如果按照传统的继承关系来计算的话，Computer 和 Mazda 之间的相似度很低。但是在实际中这两个概念之间是存在着密切的关联，因此，概念间的二义性对概念间的相似度有很重要的影响。

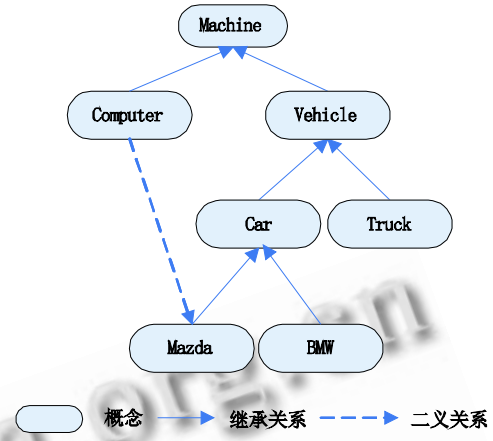

图 2　本体概念关系示例

2.2 本体概念的信息量

根据上述本体概念的图示可以得知，Vehicle 是 Machine 的子概念，Car 是 Vehicle 的子概念。概念出现的概率：Machine > Vehicle > Car，由信息论理论，出现的概率越大，所含的信息量越少，反之，概率越小，所含信息量越大。因此，概念所含的信息量可定义为如下公式(1)：

$$IC(c) = -\log_2^{p(c)} \tag{1}$$

其中，P(c)为概念出现的概率。可用如下公式(2)表示

$$p(c) = \frac{\sum_{w \in words(c)} count(w)}{N} \tag{2}$$

其中，N 为所有概念的总数量，$count(w)$ 是概念 w 出现的次数，$words(c)$ 是被概念 c 包含的子概念的集合。

由于是统计概率，则可以将上述概率的计算方法定义为：$p(c) = p(f)/n$ ，$p(R) = 1$，其中概念 f 为概念 c 的父概念，n 为概念 f 直接子概念的数量，概念 R 为根概念。如上述图中，由于出现的概念必定属于 Machine，故 P(Machine)=1，即可知 IC(Machine)=0。

2.3 本体概念之间的语义距离

在计算两个概念之间的相似度，传统的方式是基于共享概念进行的。这种计算方式不能区别具有相同最具体共同概念的概念之间的区别。因此本文将采用语义距离来计算概念之间的相似度。对父概念和子概念之间的距离通过语义距离来度量。语义距离的大小

与父概念和子概念所含有的信息量以及该父概念的直接子概念的数目有关。

对于存在直接的父子关系的概念而言，其语义距离的计算方法如公式(3)所示：

$$dis(c, parent(c)) = \frac{IC(c) - IC(parent(c))}{n \times IC(parent(c))} \quad (3)$$

其中 Parent(c)为概念的直接父概念，IC(c)为概念所含有的信息量，IC(parent(c))为概念 c 的直接父概念所含的信息量，n 为概念 c 的直接父概念所含有的直接子概念的数量。

由公式(3)可知，针对任意的两个概念其语义距离可定义为如下公式(4)

$$dis(c1,c2) = \sum_{c \in \{Path(c1,c2) - ComPar(c1,c2)\}} dis(c, parent(c)) \quad (4)$$

其中 $Path(c1,c2)$ 为连接概念 $c1$，$c2$ 的最短路径，$ComPar(c1,c2)$ 为概念 $c1$，$c2$ 的最具体的共同概念，$parent(c)$ 为概念 c 的父概念。

3 基于循环匹配的Web服务发现策略

Web 服务的发现可以分为如下几个步骤：服务类别匹配、服务描述性匹配、服务功能性匹配[5]。本文主要是针对描述性匹配中，概念间存在的二义性问题提出了一种循环匹配的解决方法。

基于上述概念间二义性关系和语义距离的描述，下面将针对本体中的概念关系，提出概念相似度计算算法。该算法的描述如下：

1)输入概念 C1、C2，并令相似度 MaxSim=0，整型变量 $i=1$，$j=1$；

2)在对应的本体概念树中，从根节点 R 采取深度优先方法遍历 C1、C2，并得到其遍历路径集合 G(Path(C1))，G(Path(C2))，并可得 G(Path(C1))中的路径条数为 n，G(Path(C2))中的路径条数为 m；

3)设定 Gi(Path(C1))表示 G(Path(C1))中的第 i 条路径，Gj(Path(C2))表示 G(Path(C2))中的第 j 条路径，计算路径 Gi(Path(C1))和 Gj(Path(C2))上的每两个相邻概念的语义距离，得到两个语义距离数组 dis[n1]，len[n2]，其中 n1 表示从根节点 R 到概念 C1 的概念个数，n2 表示从根节点 R 到概念 C2 的概念个数；

4)对遍历路径 Path(C1)、Path(C2)求交集，得到交集路径 ComPath(Path(C1,C2))，并得交集路径中概念的个数为 n0 和深度最大的概念节点 Cmax；

5)从深度最大的概念节点 Cmax 到概念 C1，C2 的语义距离分别为 $\sum_{i=(n0-1)}^{n1-1} dis[i]$，$\sum_{j=(n0-1)}^{n2-1} len[j]$；

6)则从概念 C1 到概念 C2 的语义距离可用如下的表达式进行计算

$$dis(c1,c2) = \sum_{i=(n0-1)}^{n1-1} dis[i] + \sum_{i=(n0-1)}^{n2-1} dis[i]$$；

7)由于相似度的值范围为[0,1]，且随着语义距离的增加，概念间的相似度减小，则两个概念间相似度的值可定义为 $Sim(c1,c2) = \frac{1}{dis(c1,c2)+1}$；

8)Sim(C1,C2)>MaxSim,MaxSim=Sim(C1,C2)；

9)如果 $i=n$，则继续第(10)步，否则令 $i=i+1$，继续第(3)步；

10)如果 $j=m$，则继续第(11)步，否则令 $j=j+1$，且令 $i=1$，继续第(3)步；

11)返回概念 C1、C2 的相似度 MaxSim。

如本文中给出的本体概念关系示例中，比较 Mazda 和 Computer 的相似度时，从 Machine 到 Mazda 的路径集合为{(Machine->Vehicle->Car->Mazda)}，从 Machine 到 Computer 的路径集合为{(Machine ->Computer)，(Machine -> Vehicle -> Car -> Mazda->Computer)}。则根据上述的算法，则需要先计算路径(Machine->Vehicle->Car->Mazda) 和路径 (Machine->Computer)的语义距离数组和路径交集，则通过上述算法计算出到 Mazda 到 Computer 的语义距离为：

$$dis_1 = dis(Mazda,Car) + dis(Car,Vehicle) + dis(Vehicle,Machine) + dis(Computer,Machine)$$

则可以得到的一个相似度 Sim1；然后通过变量 i，j 的循环得到另外的两条路径{(Machine->Vehicle->Car->Mazda)}和(Machine -> Vehicle -> Car -> Mazda->Computer)，同理，其语义距离为 $dis_2 = dis(Computer,Mazda)$，则得到相似度 Sim_2，由于 $Sim_2 > Sim_1$ 且 $i=1$，$j=2$，故退出循环，最终得到 MaxSim= Sim_2。

将上述算法用于计算请求服务和广告服务的描述

性匹配 Simdes，同时结合服务类别匹配 Simcat 和服务功能性匹配 Simfun 则广告服务和请求服务的相似度可表示为

$$Sim(adv,req)=w_1\times Sim_{des}+w_2\times Sim_{cat}+w_3\times Sim_{fun}$$

其中，$\sum w_i=1$，本文中取 $w_i=\frac{1}{3}$。

4　实验结果和性能分析

为了验证上述策略的可行性和有效性，做了相应的测试实验。该实验采用 OWLS-TC V2 作为服务测试集合，对 35 个服务请求按照关键字匹配策略和本文的匹配算法进行了相应的测试，35 次测试的结果取平均值得到实验结果如图 3 所示。实验结果表明，本文提出的匹配策略比基于关键字的匹配策略在查全率和查准率方面都明显要优越的多。

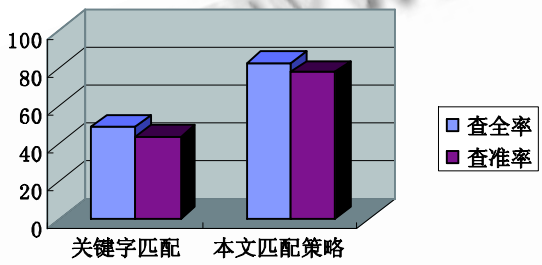

图 3　实验测试结果

参考文献

1 Massimo P, Takahiro K, Terry PR, Katia S. Importing the semantic Web in UDDI. Proc. of Web Services, E-business and Semantic Web Workshop, Toronto, Canada, 2002,225－236.

2 Martin D, et al. OWL-S: Semantic Markup for Web Services.Technical report,Damlconsortium,[2009-3-9] http://www.daml.org/services/owl-s/1.0/owl-s.pdf

3 杜小勇,李曼,王珊.本体学习研究综述.软件学报, 2006,17(9):1837－1847.

4 胡建强,邹鹏,王怀民,等.Web 服务描述语言 QWSDL 和服务匹配模型研究.计算机学报, 2005,28(4):505－513.

5 吴健,吴朝辉,李莹,等.基于本体论和词汇语义相似度的 Web 服务发现.计算机科学, 2005,28(4):595－602.