

文章编号: 1009-3443( 2008) 06-0667-04

# 一种基于语义的组合服务模板推荐算法

艾未华, 周 宁, 黄云仙

(解放军理工大学 气象学院,江苏 南京 211101)

**摘 要:** 为了实现组合服务的查找功能,提出了一种基于语义的组合服务模板推荐算法。用 Web 服务本体 OWL-S 和领域本体描述 Web 服务,将 2 个服务之间的语义相似度计算转化为加权二部图的最优匹配问题,利用改进的 Kuhn-Munkres 算法计算服务间的语义相似度;在此语义相似度的基础上提出一种基于语义的组合服务模板推荐算法。实验结果表明,提出的组合服务模板推荐算法可以快速地搜索出满足用户请求的组合服务模板。

**关键词:** Web 服务本体; 本体; 二部图; 最佳匹配; 组合服务

**中图分类号:** TP311      **文献标识码:** A

## Composite service template selection algorithm based on semantics

AI Wei-hua, ZHOU Ning, HUANG Yun-xian

(Institute of Meteorology, PLA Univ. of Sci. & Tech., Nanjing 211101, China)

**Abstract:** In order to realize the searching function of composite service, a new composite service template selection method based on semantics was proposed. In this method, the Web services were described by OWL-S and domain ontology, and the measure of semantic similarity between services was reduced to the matter of finding perfect matching in service bipartite graph which is solved by the improved Kuhn-Munkres algorithm. Finally, the composite service template selection algorithm based on the semantic similarity was proposed. The experiments show that the method proposed can accomplish the composite service selection according to the template request efficiency.

**Key words:** OWL-S (ontology web language for services); ontology; bipartite graph; perfect matching; composite service

Web 服务<sup>[1]</sup>是一种重要的分布式计算模型,通过采用 WSDL、UDDI 和 SOAP 等基于 XML 的标准和协议,解决了异构分布式计算以及代码与数据重用等问题,具有高度的互操作性、跨平台性和松耦合性的特点。单个 Web 服务提供的功能有限,如果对已有的 Web 服务进行组合,就能向用户提供更多功能的应用服务。目前 Web 服务的组合大多数是手工完成的,大多需要处理底层的编程细节,设计者需要

考虑操作交互的方法、消息之间的映射和服务调用的方法等,这种人工的方法费时费力且很难适应服务的动态性。语义 Web<sup>[2]</sup>是对现有 Web 的扩展,它与传统 Web 的不同在于,信息的语义在语义 Web 环境下能够很好地加以定义,并使得计算机之间能够更好地协同工作,其目标是让 Web 上的信息能够被机器理解,从而实现 Web 信息的自动处理。语义 Web 服务就是将 Web 服务与语义 Web 相结合,利用语义 Web 中的知识标识手段来描述服务的语义,使计算机可以理解 Web 服务,以支持服务的动态自动组合。

在面向应用的 Web 服务组合中,如果所有的应

收稿日期: 2007-09-18。  
基金项目: 解放军理工大学气象学院博士(后)科研启动基金资助项目。  
作者简介: 艾未华(1979-),男,讲师;研究方向: 语义 Web 和信号处理等; E-mail: awh1979@126.com。

用都从头开始构建,不仅应用的正确性难以保证,而且构造的效率也较低。组合服务模板封装了一定的领域知识,包含一定数量的基本服务以及它们之间的逻辑关系;如果充分利用已有的组合服务模板,就可以提高系统的效率,减少系统运行时间,最大程度的做到 Web 服务的复用。

本文研究组合服务模板发现技术,用 Web 服务本体 OWL-S<sup>[3]</sup>和领域本体描述服务,提出了一种基于语义的组合服务模板推荐算法。该算法在领域本体的支持下,用语义相似度量服务属性和参数概念之间的相符程度,并在此基础上引入服务二部图的概念,将两个服务相似程度的计算转化成二部图的匹配程度的度量,最后,在此语义相似度的基础上提出一种基于语义的组合服务模板推荐算法。

## 1 相关描述

**定义 1** 一个 Web 服务可以表示为  $WS(I_k, O_k)$ ,其中,  $WS_k$  是 Web 服务的名字,  $I_k$  是服务的输入集合,  $O_k$  是服务的输出集合。

**定义 2** 假设  $O_m(o_1, o_2, \dots, o_m)$  表示服务  $WS_m$  的输出参数的概念集合,  $O_n(i_1, i_2, \dots, i_n)$  表示服务  $WS_n$  的输参数的概念集合。定义服务参数对相似度:  $S_{i,j}(o_i, i_j)$ , 其中,  $o_i$  是  $O_m(o_1, o_2, \dots, o_m)$  中的一个概念,  $i_j$  是  $O_n(i_1, i_2, \dots, i_n)$  中的一个概念,  $S_{i,j}(o_i, i_j)$  是概念  $o_i$  和  $i_j$  之间语义相似度。

**定义 3** 计算 2 个服务  $WS_m$  和  $WS_n$  之间的语义相似程度的实质可转变成计算  $O_m(o_1, o_2, \dots, o_m)$  和  $O_n(i_1, i_2, \dots, i_n)$  2 组输出概念之间的相似程度以及 2 个服务输入参数集合之间的相似度。这个服务相似度的计算可用加权二部图匹配的最优匹配方法解决。定义  $G_{n,n} = (O_m, O_n, E)$  为一个二部图, 其中,  $O_m = (o_1, o_2, \dots, o_m)$ ,  $O_n = (i_1, i_2, \dots, i_n)$ ,  $E = (e_{11}, e_{12}, \dots, e_{mn})$ 。  $I_m$  和  $O_n$  是图顶点的集合;  $E$  是连接任意 2 个概念顶点的边的集合, 且任意边  $e_{ij}$  的二部图权值为  $S_{ij}(o_i, i_j)$ , 也就是任意 2 个概念  $o_i$  和  $i_j$  之间的概念相似度。

## 2 服务相似度计算

### 2.1 最大匹配和最优匹配

最大匹配和最优匹配是二部图中的 2 个经典问题<sup>[4]</sup>。所谓匹配就是指给定一个二部图  $G_{n,n} = (O_m, I_n, E)$ , 在  $G$  的一个子图  $M$  中,  $M$  的边集  $E$  中的任意

2 条边都不依附于同一个顶点, 则称  $M$  是一个匹配。选择这样的边数最大的子集称为图的最大匹配问题。如果边上带权, 找出权和最大的匹配叫做求最优匹配。

最大匹配解决的一个典型应用问题如下: 假设有  $n$  个工作人员  $x_1, x_2, \dots, x_n$  安排做  $m$  项工作任务  $y_1, y_2, \dots, y_m$ , 如图 1 所示, 其中, 边  $e_{ij} = (x_i, y_j)$  表示  $x_i$  可以从事  $y_j$ , 如果每个人最多从事其中一项工作, 且每项工作只能由一个人承担。如何才能让尽可能多的人安排上工作就是一个最大匹配问题。但是在最大匹配中只着眼于每一个工作人员都能安排一份工作, 并没有考虑如何更好的发挥工作人员专长的问题。而如图 2 所示, 其中, 边  $e_{ij} = (x_i, y_j)$  的权值为  $k_{ij}$ , 最优匹配解决的典型问题是: 假设每个工作人员  $x_i$  从事  $y_j$  工作时的效率为  $k_{ij}$ , 怎样合理安排才能使总的工作效率最高?

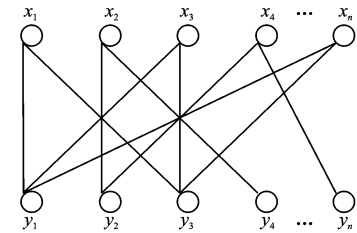


图 1 最大匹配  
Fig. 1 Maximum matching

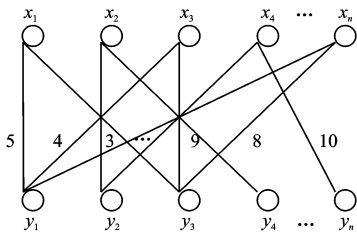


图 2 最佳匹配  
Fig. 2 Perfect matching

### 2.2 服务相似度算法

由以上分析不难看出, 计算服务相似度, 求其相似度最大值的问题类似于二部图的最优匹配问题。最优匹配方法的实质是在一一对应的前提下, 衡量两个服务的输出和输入概念集合所能达到的最大相似度。本文将二部图的思想应用到 Web 服务组合中, 将 2 个服务的关联程度的度量转化成概念集合的相似程度的计算。Kuhn-Munkres<sup>[5]</sup>算法是计算二部图的最优匹配的经典方法, 本文将该算法用于计算服务的两组概念集合的相似程度。由定义 4 知:  $G_{n,n} = (O_m, I_n, E)$  为一个二部图,  $O_m$  和  $I_n$  是图顶点

的集合,  $E$  是边的集合, 其中,  $O_m = (o_1, o_2, \dots, o_m)$ ,  $I_n = (i_1, i_2, \dots, i_n)$ ,  $E = (e_{11}, e_{12}, \dots, e_{mn})$ ,  $S_{ij}(o_i, i_j)$  是任意边的权值, 即  $e_{ij} = S_{ij}(o_i, i_j)$ . 以二部图  $G_{m,n} = (O_m, I_n, E)$  作为算法的输入, 则具体算法描述如下:

(1) 给出初始标号  $l(\alpha) = \max_j S_{ij}(\alpha, i_j)$ ,  $l(i_j) = 0$ , 其中  $i, j = 1, 2, \dots, t, t = \max(n, m)$ ;

(2) 利用 Hungarian 算法<sup>[6]</sup>求解边集  $E_l = \{(o_i, i_j) | l(\alpha) + l(i_j) = S_{ij}(o_i, i_j)\}$ ,  $G = (O_m, I_n, E_l)$  及  $G$  中的完备匹配  $M$ ;

(3) 若  $M$  是  $G$  中的完备匹配, 则  $M$  即是  $G$  的最优匹配, 计算结束, 否则进行下一步;

(4) 在  $O_m$  中找  $M$  的非饱和点  $o_0$ , 令  $A \leftarrow \{o_0\}$ ,  $B \leftarrow Q, A, B$  是 2 个集合;

(5) 若  $P_{G_l}(A) = B$ , 则转第 (9) 步, 否则进行下一步, 其中,  $P_{G_l}(A) \subseteq I_n$  是与  $A$  中结点邻接的结点集合;

(6) 找一结点  $i \in P_{G_l}(A) - B$ ;

(7) 若  $i$  是  $M$  饱和点, 则找出  $i$  的配对点  $z$ , 令  $A \leftarrow A \cup \{z\}$ ,  $B \leftarrow B \cup \{i\}$ , 转第 (5) 步, 否则进行下一步;

(8) 存在一条从  $o_0$  到  $i$  的可增广路径  $R$ , 令  $M \leftarrow M \oplus E(R)$ , 转第 (3) 步;

(9) 按下式计算  $T$  值:

$T = \min_{\substack{o_i \in A \\ i_j \notin N_{G_l}(A)}} \{l(o_i) + l(i_j) - S_{ij}(o_i, i_j)\}$ , 修改标

号:

$$l'(v) = \begin{cases} l(v) - T, & v \in A, \\ l(v) + T, & v \in B, \\ l(v), & \text{其他} \end{cases}$$

根据  $l'$  求  $E_l'$  及  $G'$ ;

(10)  $l \leftarrow l', G \leftarrow G'$ , 转第 (6) 步。

Kuhn-Munkres 算法的时间复杂度是  $O(p^4)$ , 其中,  $p = m + n$ . 求出  $G_{m,n} = (O_m, I_n, E)$  的最优匹配  $M$  后, 把  $M$  中每条边的权值  $S_{ij}(i_j, o_j)$  相加, 可以求得  $G_{m,n} = (O_m, I_n, E)$  的最大权值之和  $W$ . 本文定义 2 个服务 ( $WS_m$  和  $WS_n$ ) 的服务相似度值为, 使用  $\min(n, m)$  将这 2 个服务归一化到 0, 1 之间, 值越大表明 2 个服务的关联程度越大。

### 3 组合服务推荐算法及试验仿真

#### 3.1 组合服务推荐算法

组合服务模板库中的组合服务模板都封装了一定的领域知识, 包含一定数量的基本服务以及它们之间的逻辑关系, 为了避免对于所有的服务组合请

求都重新进行组合, 就应该充分利用组合服务模板库中的组合服务模板。当模板库中的组合服务模板数量较少时, 可以由采取手工方式查找需要的服务模板; 但是当模板库中组合服务模板达到较大的数量时, 用户要快速的找到需要的模板是比较困难的。为此, 给出了一种组合服务模板的搜索算法, 应用该算法可以自动地在服务组合模板库中搜索出满足用户需求的组合服务模板。该算法考虑了服务的功能需求和组合服务的逻辑关系; 首先找出所有与用户请求具有相同组合结构的组合服务模板, 然后利用算法 1 计算请求服务与基本服务之间的语义相似度, 再比较总体功能相似度, 最后返回候选的组合服务模板。算法的输入是: 用户请求的组合服务 SCR 以及要完成的若干基本目标  $g_1, g_2, \dots, g_m$ , 基本目标由基本服务来实现; 模板相似度阈值  $SIM-T$ , 组合服务模板库  $T(C_1, C_2, \dots, C_n)$ ,  $C_j$  是其中的一个模板。输出为符合用户请求的组合服务模板候选集合  $Team$ 。具体算法描述如框图 3 所示:

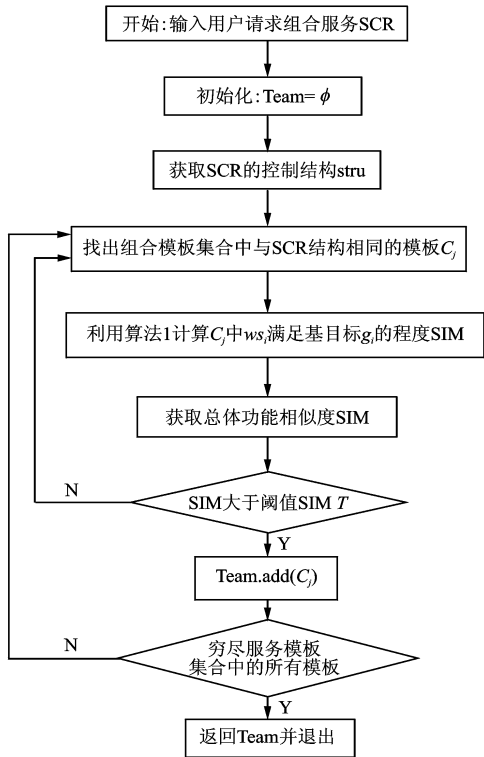


图 3 组合服务模板推荐流程  
Fig. 3 Flow of composite services template searching

#### 3.2 试验仿真

硬件环境: CPU 为 Intel Pentium IV 2.4GHz, 内存为 768 M, 操作系统为 Windows XP, 算法实现工具为 Eclipse. 实验数据: 鉴于目前没有相关的标

准平台和标准测试数据集,借助工具于 OWL-S API<sup>[7]</sup>及 JEN A API<sup>[8]</sup>等工具随机生成的模拟 Web 服务作为测试用例,所有生成的 Web 服务的服务属性和接口输入输出都是用领域本体所描述。

用 JEN A API抽取本体中的概念和相关属性描述服务接口,再利用 OWL-S API生成用于测试的语义 Web 服务集合。算法 1 是算法 2 执行的基础,为了测试算法 1 的执行效率,采用该方法进行单个基本服务间的匹配实验,即进行基本服务匹配与发现图 4 是服务匹配运行时间与候选服务个数之间的关系,从该图可以看出,随着候选基本服务个数的增加,匹配的时间也是曾线性增长趋势。在服务候选数为 200 时,服务匹配的时间为 2.88 s

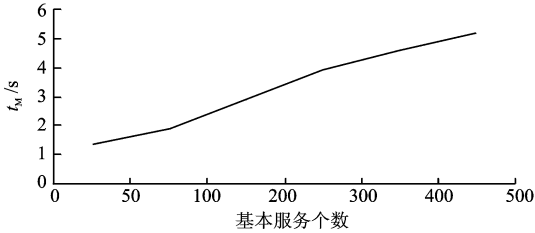


图 4 匹配运行时间与候选服务个数之间的关系

Fig. 4 Increase of searching time increases with the number of services

图 5 是在模板相似度阈值分别为 0.5 0.6 的情况下,改变候选组合服务模板数,完成一次搜索所需要的时间。可以看出,随着候选组合服务模板数的增加以及模板相似度阈值的减小,搜索的时间也相应也在增长。在阈值为 0.6 和候选组合服务模板数为 100 时,完成搜索所需的时间为 3.15 s

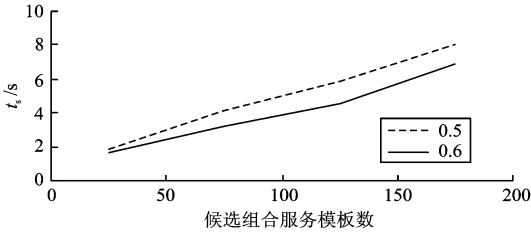


图 5 组合服务模板选择时间与候选模板数的关系

Fig. 5 Increase of searching time increases with the number of templates

4 结 语

本文研究组合服务模板发现技术,提出了一种基于语义的组合服务模板推荐算法。该算法在领域本体的支持下,用语义相似度量服务属性和参数概念之间的语义相似度,并在此基础上引入服务二部图的概念,将两个服务相似程度的计算转化成二部图的匹配程度的度量,最后,在此语义相似度的基础上提出一种基于语义的组合服务模板推荐算法。实验结果表明,论文提出的组合服务模板推荐算法可以快速地搜索出满足用户请求的组合服务模板。

参考文献:

[1] ALONSO G, CASATI F, KUNO H, et al. Web services concepts, architecture and applications [M]. Springer Verlag (ISBN: 3540440089), 2003.

[2] LEE T B, HENDLER J, LASSILA O. The semantic web[J]. Scientific American (0036-8733), 2001, 284 (5): 34-43.

[3] The OWL Services Coalition. OWL-S Semantic markup for Web Services [EB/OL]. <http://www.daml.org>, 2003.

[4] 彭宇新, NGO C W, 董庆杰,等.一种通过视频片段进行视频检索的方法 [J].软件学报, 2003, 14(8): 1409-1417.

PEN G Yu-xin, NGO CW, DONG Qing-jie. An approach for video retrieval by video clip [J]. Journal of Software, 2003, 14: 1409-1417.

[5] DAVES A M. The optimal assignment problem [DB/OL]. <http://www.math.uwo.ca/~mdawes/courses/344/kuhn-munkres.pdf>, 2003-06-21.

[6] KUHN H W. The Hungarian method for the assignment problem [J]. Natal Research Logistics Quart, 1955(2): 83-97.

[7] Mindswap. OWL-S Java API [DB/OL]. <http://www.mindswap.org/2004/owl-s/api/>. 2004-9-23.

[8] HP Labs. Jena-A Semantic Web Framework for Java. <http://jena.sourceforge.net/>. 2005-11-23.

(责任编辑:程 群)