第35卷 第20期 计 算 机 工 程 2009年10月
Vol.35 No.20 Computer Engineering October 2009
·软件技术与数据库· 文章编号：1000—3428(2009)20—0088—03 文献标识码：A 中图分类号：TP311

改进的语义 Web 服务匹配算法设计与实现

黄志成，李 华
(重庆大学计算机学院，重庆 400030)

摘 要：引用领域本体对 Web 服务进行语义描述，再进行语义层上的匹配，是 Web 服务匹配研究领域的重要研究方向。针对传统匹配算法在语义 Web 服务发现中的不足，采用语义距离、匹配度系数等对其进行扩展，在此基础上实现一个原型系统。实验表明，该算法能提高 Web 服务的匹配精度。

关键词：语义 Web 服务；本体；服务匹配

Design and Implementation of Improved Matching Algorithm for Semantic Web Service

HUANG Zhi-cheng, LI Hua
(College of Computer Science, Chongqing University, Chongqing 400030)

【Abstract】 Describing the Web service using domain ontology and matching Web service on the semantic level is a hot research spot. After analyzing the disadvantage of traditional approach for semantic matching, an improved matching algorithm which adopts semantic distance and match coefficient is proposed. A prototype system based on this method is implemented. Experimental results show that this algorithm can improve the precision of match degree of service.

【Key words】 semantic Web service; ontology; service matching

近年来，随着 Web 服务[1]应用的普及，Web 服务发现成为该领域内的一项关键技术。利用本体对特定领域的知识进行描述，再利用本体表达的知识对 Web 服务进行标注，就可把传统的基于词法的 Web 服务匹配提升到语义层次上来。在语义层次上定位服务请求者要求的服务称为语义 Web 服务[2]发现，而在语义 Web 服务发现中的服务匹配称为语义 Web 服务匹配。

1 语义 Web 服务的本体语言 OWL-S

实现语义 Web 服务匹配的关键步骤是对 Web 服务进行语义描述。OWL-S[3-4]就是一种描述 Web 服务的本体语言且基于语义 Web 标准。OWL-S 的本体结构如图 1 所示。

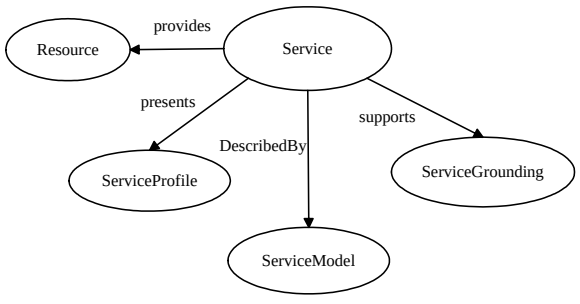

图 1 OWL-S 的本体结构

ServiceProfile 的主要功能是提供 Web 服务的相关信息供用户或服务查找代理确定服务是否符合查找要求，用于请求或广告服务。ServiceModel 描述服务执行的具体过程。包括原子过程、简单过程及组合过程。ServiceGrounding 提供细节信息让用户或智能软件代理知道如何存取服务。一般 grounding 要给出通信协议、消息格式及其他和特定服务相关的细节，如联系服务时的端口号等。总而言之，ServiceProfile 提供 Agent 发现服务所需要的信息，ServiceModel 和 Service Grounding 与服务联系在一起提供给 Agent 足够的信息以便于其使用服务。

2 传统匹配算法

传统匹配算法也叫做弹性匹配算法[5]，支持 Web 服务的非完全匹配。

定义 1 对于 2 个概念 C_i 和 C_j，在 Ontology 本体库中，如果 C_i 跟 C_j 是同一个概念或 C_i 被定义成 C_j 的等价类，则称概念 C_i 和概念 C_j 语义相等，记为 $C_i == C_j$；如果 C_i 被定义成 C_j 的子类，则称概念 C_j 语义包含概念 C_i，记为 $C_j \supseteq C_i$；如果 C_i 与 C_j 没有关联，则称概念 C_j 与概念 C_i 没有语义关系，记为 $C_j \neq C_i$。

该算法有 4 种匹配：完全匹配(Exact)，插拔匹配(Plugin)，包含匹配(Subsume)，匹配失败(Fail)。分别定义如下：

定义 2 对于服务请求概念 C_i 和服务广告概念 C_j，如果 $C_i == C_j$ 或 $C_j \supseteq C_i$ 且 C_i 是 C_j 的直接子类，则称 C_i 和 C_j 完全匹配。

定义 3 对于服务请求概念 C_i 和服务广告概念 C_j，如果 $C_j \supseteq C_i$ 且 C_i 不是 C_j 的直接子类，则称 C_i 和 C_j 插拔匹配。

基金项目：国家科技支撑计划基金资助项目(2006BAH02A24-6)；重庆市自然科学基金资助项目(CSTC, 2007BB2192)
作者简介：黄志成(1982－)，男，硕士研究生，主研方向：语义 Web 服务；李 华，副教授
收稿日期：2009-03-27 **E-mail：**willcheen@gmail.com

定义 4 对于服务请求概念 C_i 和服务广告概念 C_j，如果 $C_i \supseteq C_j$，则称 C_i 和 C_j 包含匹配。

定义 5 对于服务请求概念 C_i 和服务广告概念 C_j，如果 $C_j \neq C_i$，则称 C_i 和 C_j 匹配失败。

由以上定义可以看出，本文在完全匹配之外又增加了 3 种匹配类别，其中匹配失败类别可以被看作默认类别，只要不属于前 3 种类别都归于匹配失败。

3 传统匹配算法的不足及改进

传统匹配算法虽然实现了概念间的模糊匹配，但是其还存在一些缺陷：

(1)只考虑了相同的概念或具有直接子类关系的概念是完全匹配，而没有考虑概念之间的同义关系。从图 2 中可以看出 RealTimeMeeting 与 Meeting_Tools 是直接子类关系，是完全匹配，但是 RealTimeMeeting 跟 SynchronousMeeting 通过 equivalentClass 属性连接，是同义关系，也应是完全匹配。

(2)概念之间的匹配程度没有细化。如图 2 中的 WhiteBoard 和 Drawing_Tools 都是 ChatingRoom 的子类，按照传统匹配算法，它们均是包含匹配。但是从图中看出，相对于 Drawing_Tools，WhiteBoard 与 ChatingRoom 的匹配度更高。

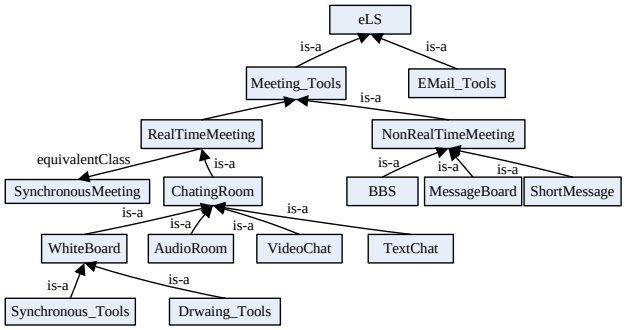

图 2 语义 Web 服务发现本体(部分)

因此，有必要扩展传统匹配算法，使其具有更精确的匹配度。对于第 1 点不足，只需在考虑概念间的层次关系的基础上引入属性名为 equivalentClass 的二元关系。这样当用户请求某一服务时，可以将与这个服务等价的所有服务都找出来。对于第 2 点不足，应当引入一种语义相似度的度量方法，这里用语义距离来表示。

定义 6 语义距离是指在同一个本体中的 2 个不同概念间存在的继承关系或二元关系链中最短的关系链长度的一种度量。

2 个概念在本体中的连接路径越短，它们就越相似。引入语义距离后还应当引入一个函数，这个函数能够细化包含匹配和插拔匹配的匹配程度，使得越靠近请求概念的本体类计算出来的匹配度越高。因此，本文引入通用余弦相似度度量(Generalized Cosine-Similarity Measure, GCSM)函数。

定义 7 最低共同祖先(Lowest Common Ancestor, LCA)，指 2 个本体概念节点共同祖先中深度最深的祖先节点。

如在图 2 中，VideoChat 与 TextChat 的 LCA 是 Chating Room，虽然 RealTimeMeeting 也是它的祖先，但是 Chating Room 的深度比 RealTimeMeeting 深。

定义 8 GCSM 函数定义如下：

$$DS(C_i,C_j)=\frac{depth(C_i)+depth(C_j)}{depth(LCA(C_i,C_j))}$$

其中，C_i，C_j 表示本体中的 2 个概念；$DS(C_i,C_j)$ 表示 2 个概念的语义距离；$depth(C_i)$ 表示 C_i 的深度，这里定义本体的根节点的深度为 0，每增加一层深度加 1。

为求得概念 C_i 的深度，本文采用图的深度优先遍历的递归算法实现。有了求领域本体中某个概念深度的算法，就可以得到概念 C_i，C_j 的 GCSM 距离。如图 2 中 WhiteBoard 和 ChatingRoom 的 GCSM 距离为

$$DS(\text{WhiteBoard},\text{ChatingRoom})=\frac{depth(\text{WhiteBoard})+depth(\text{ChatingRoom})}{depth(LCA(\text{WhiteBoard},\text{ChatingRoom}))}=\frac{4+3}{3}=\frac{7}{3}$$

同理可以得到 Drawing_Tools 和 ChatingRoom 的 GCSM 距离为 $8/3$。

得到了领域本体中 2 个概念的 GCSM 语义距离后，就要构造一个函数来根据语义距离求概念间的相似度，这就是相似度函数。

定义 9 概念 C_i, C_j 的相似度函数如下：

$$SM(C_i,C_j)=\frac{\alpha}{DS(C_i,C_j)+1}$$

其中，α 为传统匹配算法的匹配度系数，对于不同的匹配水平，α 的值不同。由于传统的匹配算法只给出了 4 种匹配类型，并没有给出 α 值，因此 α 的取值直接影响匹配算法的性能，不能随意指定 α 的值。假如指定 α 值如表 1 所示。

表 1 任意指定匹配度系数列表

匹配水平	匹配度系数
Exact	1.0
Plugin	0.8
Subsume	0.6
Fail	0.0

此时假如用户请求输出为 ChatingRoom，那么 WhiteBoard 跟它的语义匹配度为

$$SM(C_i,C_j)=0.3\times\alpha=0.3\times0.6=0.18$$

同理 Meeting_Tools 跟它的匹配度为 0.16。按照匹配度排序的原则，这时返回给用户的是输出为 WhiteBoard 的服务，但通过观察图 2，Meeting_Tools 与 ChatingRoom 是插拔匹配，实际上更符合用户的要求。这就出现计算出来的结果与实际不符的矛盾，因此，应当慎重选择匹配度系数。

推导当匹配水平为 Plugin 和 Subsume 时各自的匹配度系数过程如下：设匹配水平为 Plugin 的匹配度系数为 α，匹配水平为 Subsume 的匹配度系数为 β，那么 α，β 应当满足这样的关系：如果有服务请求输出为 A，服务 B 与 A 是插拔匹配，服务 C 与 A 是包含匹配，则有：

$$\frac{\alpha}{DS(A,B)+1}>\frac{\beta}{DS(A,C)+1} \tag{1}$$

本文只考虑领域本体中的层次关系，概念 A 的所有父类虽然与 A 都是插拔匹配，但层次越高，与 A 的语义距离越大，越不符合用户的需求，因此只要能保证 A 往上的第 3 个父类的语义相似度大于 A 往下的第 1 个子类的语义相似度即可。设 A 的深度为 d_A，B 的深度为 d_B，C 的深度为 d_C，则有：

$$DS(A,B)=(d_A+d_B)/d_B,\quad DS(A,C)=(d_A+d_C)/d_A \tag{2}$$

由于只考虑 A 的往上 3 层与 A 的往下 1 层，则 d_A，d_B，d_C 有如下关系：

$$d_C = d_A + 1，\ d_A = d_B + 3 \tag{3}$$

由式(1)~式(3)可得：

$$\alpha > ((3+(3/d_B))/(3+(1/d_B+3)))\beta \tag{4}$$

由于对任意的 d_B，都要使得式(4)成立，则应该取 β 前面系数公式的最大值。求得当 $d_B=1$ 时，函数有最大值 1.85，则：

$$\alpha > 1.85\beta \tag{5}$$

由此便可以确定 α 和 β 的值，这里取 β =0.35，则 α >0.64，所以 α =0.65。此时的匹配度系数如表 2 所示。

表 2 传统算法匹配度系数

匹配水平	匹配度
Exact	1.00
Plugin	0.65
Subsume	0.35
Fail	0.00

计算 WhiteBoard 和 ChatingRoom 的语义相似度为 $0.3\times0.35=0.105$；Meeting_Tools 与 ChatingRoom 的语义相似度为 $0.2\times0.65=0.13$。这时返回给用户的将是输出为 Meeting_Tools 的服务，符合实际情况。

根据上文的分析，总结改进后的匹配算法描述如下：

输入 概念 C_i，C_j

输出 语义相似度 $SM(C_i,C_j)$

(1)如果 C_i，C_j 匹配水平为 Exact，直接返回 $SM(C_i,C_j)$ =1；

(2)如果 C_i, C_j 匹配水平为 Plugin，先计算 $DS(C_i,C_j)$，再返回 $SM(C_i,C_j)=0.65\times DS(C_i,C_j)$；

(3)如果 C_i, C_j 匹配水平为 Subsume，先计算 $DS(C_i,C_j)$，再返回 $SM(C_i,C_j)=0.35\times DS(C_i,C_j)$；

(4)如果 C_i, C_j 匹配水平为 Fail，直接返回 $SM(C_i,C_j)$ =0。

4 算法实现与分析

本文实验采用 Protégé3.3 作为领域本体建模工具，RacerPro 作为推理机。利用计算机随机产生和人工标注已有 Web 服务相结合的方法随机生成了 970 个 Web 服务的语义描述，人工标注了 30 个 Web 服务(共 1 000 个已发布的 Web 服务语义描述)。匹配时先计算输出语义相似度，再计算输入语义相似度，最后得到总的相似度，并最终按相似度排序。当输入参数 text, ChatingRoom 后，服务搜索结果如表 3 所示。

表 3 服务搜索结果

服务名称	匹配度
ChatingRoomService	1.000
RealTimeMeetingService	1.000
Meeting_ToolsService	0.130
WhiteBoardService	0.105
AudioRoomService	0.105
Drawing_ToolsService	0.095

可以看出，返回的结果跟实际相符合，说明了本文算法的正确性与有效性。

5 结束语

本文在传统匹配算法的基础上使用 GCSM 语义距离、匹配度系数来量化语义 Web 服务的相似度，并且对算法进行了验证。实验结果表明，改进后的算法较原有算法具有更好的查准率和查全率。但本文只考虑了本体的层次关系，下一步将对本体的多元关系进行研究。

参考文献

[1] 岳 昆, 王晓玲, 周傲英. Web 服务核心支撑技术: 研究综述[J]. 软件学报, 2004, 15(3): 114-128.

[2] Berners-Lee T. Hendler J, Lassila O. The Semantic Web[J]. Scientific American, 2001, 284(5): 34-43.

[3] OWL-S Coalition. OWL-S 1.0 Release[EB/OL]. (2004-08-17). http://www.daml.org/services/owl-s/1.0/.

[4] Martin D, Burstein M, Hobbs J, et al. OWL-S: Semantic Markup for Web Service[Z]. (2004-11-22). http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/.

[5] 张 钋. 基于语义的网络服务匹配机制的研究与实现[D]. 北京: 清华大学, 2005.

编辑 顾姣健

~~~~~~

(上接第 87 页)

参考文献

[1] Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Item in Large Database[C]//Proc. of the ACM SIGMOD Conference on Management of Data. Washington, USA: ACM Press, 1993.

[2] Han Jiawei, Pei Jian, Yin Yiwen. Mining Frequent Patterns Without Candidate Generation[C]//Proc. of the ACM SIGMOD International Conference on Management of Data. [S. l.]: ACM Press, 2000.

[3] Park Jong Soo, Chen Ming-syan, Philips S. An Effective Hash Based Algorithm Forming Association Rules[C]//Proc. of the ACM SIGMOD International Conference on Management of Data. [S. l.]: ACM Press, 1995.

[4] 张 铃, 张 钹. 问题求解理论及应用——商空间粒度计算理论及应用[M]. 北京: 清华大学出版社, 2007.

[5] 柏文博. 基于商空间的关联规则的挖掘[D]. 鞍山: 辽宁科技大学, 2007.

[6] 尹世群, 余建桥, 葛继科, 等. 基于粗糙集的分类关联规则挖掘算法研究[J]. 计算机科学, 2007, 34(12): 171-174.

编辑 张 帆
~~~~~~