

# 一种WEB服务发现与匹配方法<sup>①</sup>

陈国彬

(重庆工商大学 融智学院, 重庆 400033)

**摘要:** 随着电子商务在互联网的飞速发展, Web 服务近年来备受关注。企业可以在互联网上提供自己的服务业务, 并且可以动态地将各个服务提供新的增值服务。然而问题在于所需的服务如何被发现。提出了 Web 服务的发现和匹配机制, 给出一个匹配树算法来衡量发现与服务连接, 以评估服务的重要性。

**关键词:** Web 服务; 服务发现; 树匹配; 评估服务

## WEB Service Discovery and Matching Method

CHEN Guo-Bin

(Rongzhi College, Chongqing Technology and Business University, Chongqing 400033, China)

**Abstract:** With the rapid development of electronic commerce on the Internet, Web services have attracted much attention in recent years. Enterprises can provide their own Internet services business, and each service can dynamically offer new value-added services. However, the question is how the services needed to be found. In this paper, we propose a Web service discovery and matching mechanisms and a matching tree algorithm to measure the service connection, to evaluate the importance of service.

**Key words:** Web services; service discovery; tree matching; assessment services

### 1 引言

Web 服务基于互联网业务, 用户可以访问并通过例如 XML, WSDL<sup>[1]</sup>, UDDI 和 SOAP<sup>[2]</sup>标准进行调用。随着电子商务在互联网的飞速发展, Web 服务已经吸引了近年来备受关注。企业可以在互联网上提供自己的服务业务, 使他们通过网络<sup>[3]</sup>访问。然而, web 服务可以组合成更复杂的个性化服务。随着网络服务和面向服务计算的应用<sup>[4]</sup>, web 服务日益增长, Web 服务发现和匹配的有效机制显得至关重要。为了解决上述问题, 在本文中提出的 Web 服务发现与匹配, 具体工作如下: 1、首先我们定义服务的相关性和服务的重要性; 2、服务连接, 以评估服务的重要性。

### 2 web服务的度量模型

设 Q 是一个所需的 Web 服务所需的请求集,  $S = \{S_1, S_2, \dots, S_K\}$  是通过 UDDI 发布的所有可用的服务集,

$D = \{D_1, D_2, \dots, D_K\}$  是对于每个服务 WSDL 文件  $D_i$  对应于  $S_i$  的服务。我们描述了每个服务的 N 记为  $\{(t_1, w_{i1}), (t_2, w_{i2}), \dots, (t_N, w_{iN})\}$ , 其中每一项都分配了一个权值。权的公式是  $t_f / i_{df}$ , 即词出现的次数 (TF) 和文档出现的次数 (IDF) 上。通常情况下, 每个文件  $t_j$  权  $w_{ij} = t_{ij} \times idf_j = t_{ij} \times \log(k/n_j)$ , 其中 K 为有效的 web 服务的个数。

定义 1 (服务相关性) 一个服务  $S_i$  相关性描述为, 对于 Web 服务描述, 定义如公式 1:

$$RS_i = \frac{\sum_{k=1}^N w_{ik} \times w'_{ik}}{\sqrt{\sum_{k=1}^N (w_{ik})^2} \cdot \sqrt{\sum_{k=1}^N (w'_{ik})^2}} \quad (1)$$

其中  $w_{ik}$  是 K 项  $\vec{s}_i$  的权和  $w'_{ik}$  是 K 项  $\vec{q}$ ,  $RS_i \in (0, 1)$ 。RSI 值越高, 服务 SI 对应的 Q 的相关性越高, 说明用户请求 Q 和有效的 Web 服务紧密相关。

① 收稿时间:2011-06-15;收到修改稿时间:2011-07-20

定义 2 Web 服务是一个三元组  $WS=(TpSet,MsgSet,OpSet)$ ，其中  $TpSet$  是数据类型集合， $MsgSet$  是一组消息（参数）中的数据符合  $TpSet$  定义类型， $OpSet = \{opi (inputi, outputi) | i= 1,2,\dots,n\}$  是一个操作集合，其中  $inputi$  和  $outputi$  是参数集（消息）为 Web 服务操作交换数据。图 1 给了三个 Web 服务在本文的例子使用的操作。根据定义 2，Web 服务可以简单地描述为一组操作。

WS1: Web Service: CreateOrder	
Operation: <i>OrderBuilder</i>	
Input: UserID	Data Type: <i>int</i>
Requirement	Data Type: <i>ItemList</i>
Output: ProductsList	Data Type: <i>BuyingOrder</i>
WS2: Web Service: ProcessPayment	
Operation: <i>CheckoutOrder</i>	
Input: UserProducts	Data Type: <i>UserOrder</i>
Output: PaymentConfirmation	Data Type: <i>bool</i>
WS3: Web Service: TransportOrder	
Operation: <i>ShippingOrder</i>	
Input: Cargo	Data Type: <i>Order</i>
Output: PickupTime	Data Type: <i>TimeLimit</i>

定义 3 (web 服务操作重要性) 一个 Web 服务操作 OP 的重要性,计算公式 2 迭代直到达到收敛:

$$I_{op}^r = (1-p) + p * \sum_{j \in F_{op_j}} Con_{op_j \rightarrow op} * I_{op_j}^{r-1} * 1/N_{op_j} \quad (2)$$

其中  $Con_{op_j \rightarrow op}$  为 OP 连接操作  $op_j$  服务的相似度,  $R$  表示迭代的次数,  $F_{op_j}$  是能够被  $op_j$  连接的集合,  $N_{op_j} = |F_{op_j}|$  连接数量,  $0 \leq p \leq 1$  时是一个调整参数, Web 服务操作的重要性是被其它因素影响。对于服务操作,最初的重要性  $I_0$  为设置为  $1/N$ , 其中  $N$  是可用的 Web 服务操作的总数。

定义 4 每个 Web 服务操作具有多输入多输出功能  $F$  序列:  $S_1, S_2, \dots, S_{n-1} \rightarrow T_1, T_2, \dots, T_m$  其中  $S_i$  和  $T_j$  都符合 W3C、XML 模式数据类型,  $F$  为相关和  $S_i = T_j$  一个相关属性。一个相关项属性可以是一个复杂的数据类型。现在每个数据类型的网络服务的操作型定义可携带语义,使用 XML 架构规范。它更换为它是基于 XML 的 DTD 的语法。XML 文档树的结构, 形成一个在根和叶节点。节点  $n$  标签分类:

1 标签节点: 每个标签节点  $n$  的一个元素类型  $T$  相关联的, 也是全节点的标记名称。

2 约束节点:

序列节点: 一个节点序列表示其子元素类型的有序集合。用  $[,]$  表示一个序列节点。

复合节点: 一个复合节点代表一个复合类型, 即

实例只能是与 XML 模型规范根据子节点的类型。我们使用  $[,]$  表示一个复合节点。

多节点: 每个节点可以多选择性  $[M,N]$ , 在  $M$  与  $N$  之间, 是全这相当于最大值与最小值之间在 XML 模式约束, 使用  $[M,N]$  来表示的多节点。数据类型订单模式树状显示 如图 2。

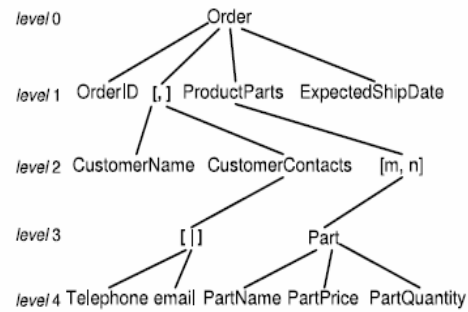


图 2 订单的树状图

web 服务从服务相关性、服务重要性和 web 树型结构等方面来进行研究, 服务相关性研究服务间相似程度, 服务重要性讨论在相关性基础上对服务操作进行研究, 树型结构对服务进行分类自底而上从子服务到复合服务。

### 3 服务操作连接

定义 5 (Web 服务操作连接): 两个 Web 服务操作  $OP_1$  的:  $S_1, S_2, \dots, S_n \rightarrow T_1, T_2, \dots, T_m$  和  $OP_2$ :  $U_1, U_2, \dots, U_l \rightarrow V_1, V_2, \dots, V_k$ , 设  $X = \{T_1, T_2, \dots, T_m\}$  和  $Y = \{U_1, U_2, \dots, U_l\}$ 。则  $OP_1$  的  $OP_2$  的连接操作可用  $X$  和  $Y$  相似度, 定义  $Con_{op_2 \rightarrow op_1} = SIM(X, Y)$ 。服务操作  $OP_1$  为  $OP_2$  的连接, 如果连接度  $Con_{op_2 \rightarrow op_1}$  是一个阈值 ( $0 << 1$ )。如果  $OP_1$  的具有完全相同的  $OP_2$  的输出数据的输入方式, 我们将有  $Con_{op_2 \rightarrow op_1} = SIM(X, Y) = 1$ , 说明  $OP_1$  的  $OP_2$  的连接最佳。在这种情况下, 我们说  $OP_1$  为  $OP_2$  的完全连接。相反, 如果  $OP_1$  输出与  $OP_2$  输入是完全不同, 有  $Con_{op_2 \rightarrow op_1} = SIM(X, Y) = 0$ , 表明  $OP_1$  与  $OP_2$  的完全不能连接。

#### 3.1 树距

树距是描述两棵树之间的差异, 也是用来说明不同服务之间的连接。首先介绍树距操作, 树距操作一般包括: 1、节点删除, 2、节点插入, 3、节点重新标记。两棵有序树且有根的顶点距离是从一棵树转化另一棵树的操作序列包括节点的删除与重新标记, 以及

插入新节点的最小代价。这样一组操作两棵树之间的最低成本的映射。映射的概念被正式定义如下:

定义 6 设  $T_x$  树,  $T_x[i]$  是  $T_x$  树前序遍历的第  $i$  个节点。树  $T_1$  和  $T_2$  的树映射是一个有序对  $pair(i,j)$  的  $M$  集, 满足所有下列条件  $(i_1,j_1), (i_2,j_2) \in M$ :

- ① 如果  $j_1 = j_2$ , 则  $i_1 = i_2$ ;
- ② 如果  $T_2[j_1]$  是  $T_2[j_2]$  的左结点, 则  $T_1[i_1]$  是  $T_1[i_2]$  的左结点;
- ③ 如果  $T_2[j_1]$  是  $T_2[j_2]$  的根结点, 则  $T_1[i_1]$  是  $T_1[i_2]$  的根结点;

图 3 给出了一个树映射的例子。这种映射显示从左结点到右结点路径。 $T_1$  节点到节点  $T_2$  虚线表明,  $T_1$  节点应该改变, 如果相应节点是不同的, 否则保持不变。不是虚线连接  $T_1$  节点都将被删除, 并没有连接  $T_2$  的节点插入。

$$\text{cost}(\rho) = \begin{cases} \text{weight}(n)/W(T_1, T_2), & \text{if } \rho = \text{insert}(n) \\ \text{weight}(n)/W(T_1, T_2), & \text{if } \rho = \text{delete}(n) \\ \alpha \times \text{wd}(\text{node}_1, \text{node}_2) + \beta \times \text{sd}(\text{node}_1, \text{node}_2), & \text{if } \rho \text{ relabels } \text{node}_1 \text{ to } \text{node}_2 \end{cases} \quad (3)$$

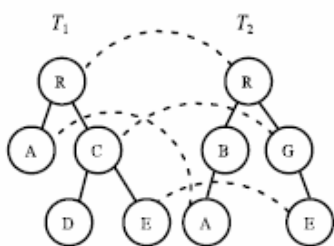


图 3 树映射

这些操作都分配了一个成本。树之间的距离为最小集合的操作转换成另一种树的定义。我们的模式匹配算法是基于树距算法如下:

```

input: Web 服务操作集合: OP={op1, op2, ..., opN}
output: OP 操作的重要值存放在 I[1:N]
foreach 服务操作 opi ∈ OP do
    Iicur = 1 / N ;
conver [1:N]=false;
end
repeat
foreach 服务操作 opi ∈ OP do
    计 Iicur = 1 / N 算在公式 2 ;
    if |Iinew - Iicur / Iicur ≤ C then
        conver[i]=true;
    else

```

```

conver[i]=false;
end
end
until conver [1:N]=true;
Return I;

```

### 3.2 成本模型

在两个 XML 模式相似性树找到一个成本最小的映射, 在涉及的每个的映射操作成本计算。成本模型的集成节点和节点之间的语义连接权重。设  $T_1$  和  $T_2$  两棵树, 设  $n, \text{Node}_1, \text{Node}_2$  都是树节点, 在形式上, 成本模型定义如公式 3:

其中表示树操作。 $\text{weight}(n)$  是节点  $n$  的权值, 这是在定义 7 进行描述,  $\text{wd}(\text{node}_1, \text{node}_2)$  和  $\text{sd}(\text{node}_1, \text{node}_2)$  分别表示  $\text{Node}_1$  和  $\text{Node}_2$  权值和语义的差异, 相应地用来表示  $\text{wd}$  和  $\text{sd}$ , 满足  $\text{wd} + \text{sd} = 1$ 。  $W(T_1, T_2)$  的定义  $W(T_1, T_2) = \text{Weight}(T_1) + \text{Weight}(T_2)$ , 其中权是树体各节点权的总和 ( $\mu = 1, 2$ )。  $\text{wd}(T_1, T_2)$  定义如公式 4:

在  $\text{node}_1 \in T_1$  和  $\text{node}_2 \in T_2$ 。在公式 3 中,  $\text{weight}(n)/W(T_1, T_2)$  为插入或删除节点  $n$  的费用, 对于重新标记操作, 既  $\text{node}_1$  和  $\text{node}_2$  权和语义有所不同, 因此重新标记成本为权和语义差异的组合。  $W(T_1, T_2)$  称为费用总和, 即树  $T_1$  和  $T_2$  所有节点权重的总和。

### 3.3 XML 树转换

定义 7 设  $\text{level}(n)$  表示的树  $T$  中的节点  $n$  层次。节点  $n$  的权定义公式 5:

$$\text{weight}(n) = 2^{\text{depth}(T) - \text{level}(n)} \quad (n \in T) \quad (5)$$

所有节点的权在  $[2, 2^{\text{depth}(T)}]$  范围内。每个权重反映了一个节点模式树  $T$  的重要性。我们提出了三个转换规则, 这些规则是用来变换约束节点, 序列节点, 复合节点和多节点来标记节点。在同一时刻, 节点的权重重新分配按以下规则:

1) 分割: 用于序列节点。一个序列  $\text{node}_k = [k_1, k_2, \dots, k_n]$  被分成一个有序的节点的  $k_1, k_2, \dots, k_n$ , 其中  $l_i = (i=1, 2, \dots, s)$  是一序列节点  $k$  子节点, 每个节点序列取代它的子节点。每个子节点  $k_i$  继承了其父节点  $k$  作为一个新权。图 4(a) 给出了一个分割规则的例子。

2) 合并: 用于复合节点。合并后的过程中, 每个节点联盟所取代所有的子节点, 复合节点的所有子节点为  $[k_1 | k_2 | \dots | k_s]$  合并成一个新节点  $k^*$ , 而子节点被删除。节点的权为  $k^*$  的权为,  $\text{weight}(k^*) = \text{weight}(k) * s$ , 图 4 (b) 给出了一个合并规则的例子。

(下转第 116 页)

表 3 网络半径和最佳分环数之间的关系

网络半径 $R/m$	20	30	50	80	100	120	150	170	200	250
最优分环数 $n$	3	4	6	9	11	13	15	17	20	24

### 4 结论

本文提出了一种在能量均衡条件下的传感器节点最小能耗的策略，该策略通过对圆形网络模型进行等距划分，比较不同的分环数来得出能耗最小的环数划分，同时给出了不同分环下每环的簇头选举数目。利用 MATLAB 计算结果表明，该策略能够有效地减少节点的能量消耗，延长网络的寿命。

### 参考文献

1 郭书城,卢昱,许定根.基于分簇无线传感器网络的路由算法研究.通信学报,2010,31(8A):63-69.

2 陈祖爵,麻颢光,陈媛.基于能量均衡的动态间隔分层路由协议.计算机应用研究,2011,28(1):271-274.  
 3 吴振华,尹志军.基于优化簇半径的 WSNs 非均匀分簇路由.计算机工程与设计,2010,31(15):3374-3377.  
 4 刘浩,唐培和.无线传感器网络中能耗平衡的传输策略.计算机工程与应用,2010,46(33):112-114.  
 5 何永刚,徐汀荣,彭俊.无线传感器网络分簇方法的优化.计算机工程与应用,2010,47(1):92-95.  
 6 陈林星.无线传感器网络技术与应用.北京:电子工业出版社,2009.157-159.  
 7 靳淑娟.无线传感器网络中的能量均衡路由协议研究.大连:大连理工大学,2009.

(上接第 124 页)

3) 删除: 用于多节点。删除一个多节点  $k=[M,N]$  ( $m,n \in N$ ) 和扩大了其子节点  $k_i$  的权。删除后，每个节点取代它的子节点。用  $weight(k^*)=weight(k) * (m+n)/2$ 。图 4(c)给出了一个删除规则的例子。

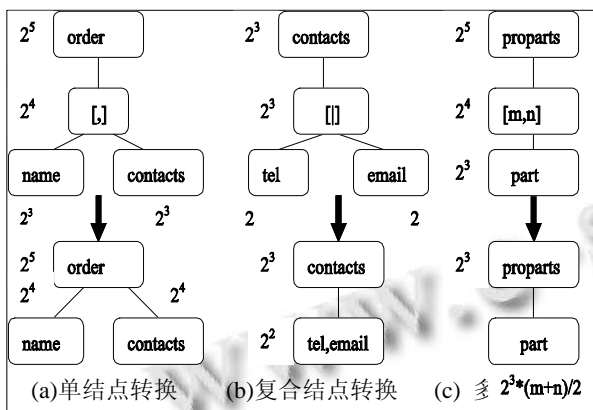


图 4 树转换

树距描述了 web 服务在不服务器之间的连接关系，不同服务之间映射关系，服务连接从树距的关系进行连接服务。XML 树转换是对树距进行转换，把复

杂的服务进行转换成简单地树距模型，从而降低了服务的成本。

### 4 结语

本文提出了 Web 服务的发现和匹配机制，给出一个匹配树算法来衡量服务连接，把 Web 服务通过对树匹配模型来分析服务连接，再把服务成本计算进行来评估服务的成本问题。

### 参考文献

1 Christensen E. Web services description language(WSDL)1.1.2001. http://www.w3.org/.  
 2 Feng YC, Yi BL, Tan ZP. Design and Implementation of WebGIS Based on SOAP Wuhan University Journal of Natural Sciences, 2008,8(4).  
 3 Bhiri S, Perrin O, Godart C. Ensuring required failure atomicity of composite web services. Proc. of WWW Conference, 2005,138-147.  
 4 王先平,齐逸.面向服务计算中 Web 服务动态绑定模型研究.计算机工程与设计,2010,31.