

基于语义的 Web 服务动态组合框架的研究

闫志刚, 胡海涛

(华北电力大学 控制与计算机工程学院, 北京 102206)

摘 要: 提出一种基于语义的 Web 服务动态组合框架. 根据服务接口之间的依赖关系, 动态地确定服务组合, 使用基于语义的概念匹配, 完成服务接口之间的匹配. 用户根据需求, 确定输入输出参数, 程序可以自动地确定服务组合, 来满足用户的需求.

关键词: Web 服务; SAWSDL; 概念匹配; 服务组合

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2010)08-0197-04

Study on Web Service Dynamic Composition Framework Based on Semantics

YAN Zhi-gang, HU Hai-tao

(School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China)

Abstract: A framework was proposed for web service dynamic composition based on semantics. According to the dependence of service's interfaces, service composition can be confirmed dynamically. Concept matching based on semantics is used to accomplish the match between service interfaces. Users confirm the input and output parameters in line with the requirement, then service composition can be confirmed automatically by procedure.

Key words: web service; SAWSDL; concept matching; service composition

1 引言

随着 Web 服务技术的日益成熟,越来越多的稳定易用 Web 服务共享在网络上. 但单个的 Web 服务能够提供的功能有限, 为了更加充分地利用共享的 Web 服务, 有必要将共享的 Web 服务组合起来, 提供更为强大的服务功能, 加快系统开发的速度, 满足用户需求.

Web 服务组合方法从组合方案生成方式来分有两大类: 静态组合和动态组合. 静态组合服务是在设计或装载时进行, 缺少应用的灵活性; 动态服务组合是在运行时进行, 能给用户带来更大的灵活性^[1]. 静态组合中最常用的是用图来描述过程模型. 而动态组合不仅自动地选择、绑定 Web 服务, 同时更重要的是自动地创建过程模型. 这需要请求者

指定一些约束关系, 包括 Web 服务间^[2]的依赖关系、用户的偏爱等.

文中提出一种动态的 Web 服务组合的方式, 用户只需要确定要通过 Web 服务得到什么(输出)及相关的已知条件(输入). 根据这些输入和输出就能动态地确定 Web 服务的组合.

2 相关背景

文献[3-4]给出了关于领域本体及 Web 服务的相关定义, 文中借鉴这些定义, 给出如下形式化描述.

定义 1 领域本体模型可以用五元组表示: $O = \{C, P, R, H, X\}$, 其中 C 表示概念的集合, P 表示属性的集合, R 表示关系的集合, H 表示继承关系集, 包括概念继承和属性继承, X 为本体规则集.

定义 2 一个 Web 服务可以用下面的表达式来描述: $WS = \{ F, I, O, P, E \}$, 其中 F 表示该服务的功能, I 表示该服务的输入集合, O 表示该服务的输出集合, P 表示该服务执行的前提条件, E 表示该服务执行后的效果.

2.1 建立领域本体

本体的描述语言有很多, 文中使用 OWL 来建立领域本体, 具体用到一种图形化的本体编辑工具 Protégé^[5] 来建立领域本体. 这里以足球比赛为例子, 建立一个简单的本体. 在后面的概念匹配中, 要用到此本体模型.

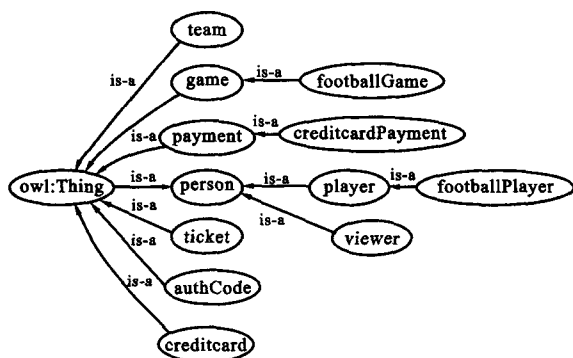


图 1 足球比赛本体模型

2.2 使用 SAWSDL 语义标注

SAWSDL^[6] 试图通过从 WSDL 和 XSD 元素中引用语义模型的方法来填补 Web 服务和语义网的鸿沟, 这些语义模型在 WSDL 外部定义. 引用由标注指定, 它利用了 WSDL、WSDL 2.0 和 XSD 的可扩展框架. 有两种主要的标注机制:

(1) 一个扩展属性, 称为 modelReference, 指定 WSDL 或 XML Schema 组件与某些语义模型中概念之间的关联. 它用来标注 XML Schema 类型定义、元素声明和属性声明, 以及 WSDL interfaces、operations 和 faults.

(2) 两个扩展属性, 称为 liftingSchemaMapping 和 loweringSchemaMapping, 它们被加入到 XML Schema 元素声明和类型定义, 用于指定语义数据和 XML 之间的映射.

2.3 概念匹配

基于语义推理的概念匹配是建立在领域本体概念间的层次关系之上的. Massimo Paolucci 等人于 2002 年提出了一种使用离散值来评价两个概念匹配程度的算法^[7]. 匹配结果采用离散值来度量, 分为 exact、plugIn、subsumes 和 fail 四个级别.

定义 1 中描述了领域本体的五部分, 其中 C 表示概念的集合, R 表示关系的集合. 两个概念之间

的关系可以使用 Jena 和 SPARQL 语言^[8] 来查询得到, 比如, equivalentClassOf, subClassOf 或没有关系. 用 inR 表示请求者的输入参数, inA 表示提供者的输入参数, outR 表示请求者的输出参数, outA 表示提供者的输出参数. 具体描述如下:

(1) exact, 如果 inR 和 inA 使用同一概念, 或者 inR 和 inA 是“equivalentClass”关系, 则返回 exact; 或者是 inR 是 inA 的直接子类, 也认为满足要求, 返回 exact.

(2) plugIn: inA 包含 inR, 但是 inR 不是 inA 的直接子类, 即服务提供者的输入参数概念包含服务请求者的输入参数概念.

(3) subsumes: 和 plugIn 相反, inR 包含 inA, 即服务请求者的输入参数概念包含服务提供者的输入参数概念.

(4) 当两者不属于以上三种情况时, 我们认为匹配失败.

3 动态 Web 服务组合新框架

3.1 框架结构

本框架采用的一种基于服务接口匹配的 Web 服务组合方法, 通过计算服务接口之间的匹配度, 找出潜在的消息的依赖关系, 就能自动发现潜在的 Web 服务, 从而实现 Web 服务的动态组合. 框架的核心部分在服务组合模块, 由服务组合模块接受输入分析模块的输入, 调用匹配模块来进行概念匹配, 将得到的服务组合结果传给服务执行模块, 框架的结构如图 2 所示.

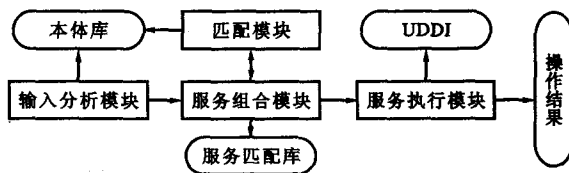


图 2 服务组合框架结构图

框架各部分的功能如下:

(1) 输入分析模块: 将用户的输入和本体库中的概念相关联, 使其能通过领域本体被机器理解.

(2) 匹配模块: 实现概念之间的匹配, 主要是将 2.3 节部分用代码实现.

概念匹配的算法如下, inR 为请求服务的输入参数, inA 为广告服务的输入参数, 其中 β 为一个调整参数, 可以根据需要赋值.

```
MatchDegree(inR, inA) {
    cReltn = conceptMatch(inR, inA);
    if (cReltn == exact)
```

```

k = 0;
if (cReltn == plungIn)
    k = 1;
if (cReltn == subsumes)
    k = 2;
else
    return 0;
return  $\beta / (\beta + k)$ ;

```

返回的匹配度的范围在 $[0, 1]$, 为 0 即为不匹配, 为 1 为完全匹配。

(3) 服务组合模块: 根据服务组合的算法, 对输入分析模块传过来的输入输出参数进行处理, 根据服务接口直接的依赖关系, 在服务匹配库中找到符合要求的服务组合。

(4) 服务执行模块: 接收服务组合模块传过来的服务组合, 从 UDDI 中找到相应的 Web 服务. 使用 J2SE 6.0 中的 wsimport 命令, 根据服务的 WSDL 地址来生成调用 Web 服务的相关类文件, 来生成客户端程序. 这里使用 Axis2 来开发服务执行模块, Axis2 的体系结构高度灵活, 支持很多附加功能, 如可靠消息传递和安全性等^[9]. 根据得到的服务组合, 自动生成客户端程序, 用户输入数据, 运行程序, 得到结果。

3.2 实现服务动态组合

考虑到算法的时间和空间复杂度, 文中处理的 Web 服务具有这种特点: 多输入, 单输出, 这样实现起来就比较简单, 运行时间也大大缩短. 针对 Web 服务的这种特点, 本文采取一种从用户的输出开始的反向匹配的策略^[10]. 即首先服务请求者确定想要查找的 Web 服务的输入和输出参数, 并和领域本体中的概念对应起来, 然后在匹配库中找到和用户的输出参数匹配的服务列表, 将其按匹配度排序, 然后一直反向匹配, 直到所有服务的输入参数和用户的输入参数全部匹配。

具体的服务组合算法用伪代码表示如下:

输入: 请求服务的输入参数集合 inRList, 输出参数 outR, 同分类的服务列表 sList.

输出: 满足条件的服务组合.

```
Composite (inRList, outR, sList){
```

```
    MatchedList = findMatchedList(outR, sList); // 找到输出和 outR 匹配的服务列表
```

```
    sort(MatchedList); // 将列表按匹配度排序
```

```
    for service in MatchedList{
```

```
        unMatchedParams = MatchInputParam (inRList,
        service.inRList); // 将请求服务的输入参数与
        service 的输入匹配, 得到未匹配的参数集合
```

```
    if unMatchedParams is empty{
```

```
        serviceLink.append(service);
        return serviceLink; // 如果输出参数全部匹配, 则
        返回 serviceLink;
    }
```

```
else if part matched // 如果输出参数部分匹配, 则递归
调用 Composite()方法.
```

```
for ALL param in unMatchedParams{
    serviceLink.append(Composite(unMatchedParams,
    param, sList));
}
```

此算法递归调用了 Composite()方法, 直到成功返回 serviceLink, 则为得到的服务组合. 如最后 serviceLink 不为空, 则认为匹配成功. 算法中对输出参数的匹配时, 对得到的服务列表按匹配度来排序, 按匹配度从高到底开始重新匹配, 一旦所有输入参数得以匹配, 即返回 serviceLink. 因此, 这样得到的服务组合一定是匹配度最高的, 并且所用时间也很少。

4 应用实例分析

假如要找到一个购买南非世界杯比赛的门票的服务, 现在 UDDI 中有 GetGameNum, GetAvailability, CreditCardPayment, PurchaseTicket 这几个服务, 一个用户想要根据比赛的双方、信用卡的卡号和密码以及票数来购买门票, 则输入参数概念为"teamName"、"cardID"、"password"、"ticketNum", 输出参数为"purchaseResult". 此时服务匹配库中有服务如表 1 所示。

表 1 服务信息表

服务名称	输入	输出
GetGameNum	teamName, teamName	gameID
GetAvailability	gameID	ticketAvailableNum
CreditCardPayment	cardID, password	authCode
PurchaseTicket	ticketNum, gameNum, authCode	purchaseResult

根据服务组合算法, 整个服务组合的过程如下:

首先, 根据请求服务的输出参数概念 purchaseResult 和服务匹配库中的服务匹配, 发现只有服务 PurchaseTicket 的输出满足要求. 然后再以 PurchaseTicket 的输入和用户输入的参数做匹配, 得到未匹配的参数列表为 {gameNum, authCode}. 再以 gameNum 为输出和匹配库中服务匹配, 发现服务 GetGameNum 与其匹配, 而此服务的输入和用户输入参数完全匹配; 再以 authCode 做为输出和匹配库中的服务匹配, 发现服务 CreditCardPayment 与其匹配, 而此服务的输入参数 {cardID, password}, 和用户输入参数完全匹配, 此时整个匹配过程结束. 得

到的服务组合如下:

purchase Ticket Service(get Game Num Service (team1, team2), payment (cardNum, password), ticketNum).

根据得到的服务组合,生成客户端程序,用户输入相应的输入参数的值(team1:南非, team2:墨西哥, cardNum: 12345, password: 123, ticketNum: 3),再运行客户端程序,就能得到结果,完成购票业务.

5 结束语

文中提出了一种基于本体的 Web 服务组合框架,使用 SAWSDL 来给 WSDL 文件添加语义标注,以 OWL^[11]作为本件建模语言,采用 Protege 建立领域本体,概念匹配中使用 Jena 和 SPARQL 语言来推理.

本框架的优点:

(1) 匹配过程中使用了本体技术,使得查准率和查全率都有所提高.

(2) 根据输入和输出参数来动态确定服务组合,这样比直接用 UDDI 查询的灵活性大大提高,并且提供自动服务组合的能力.

(3) 对普通的 WSDL 文件进行扩展,可以跟原有系统良好兼容,不用做很大的改动.

缺点是当服务匹配库中的服务数量很多时,效率就会比较低;而且由于匹配中没用涉及到服务的前置条件(Preconditons)和效果(Effects),查询到的 Web 服务,不一定完全符合用户的需求;由于在服务组合时 Qos 模型建立起来比较困难,此框架并没有考虑服务的 Qos 约束;另外此模型对概念匹配的推理只有使用了包含关系.今后可以进一步的加入更复杂的推理机制,从而更大程度上发挥 OWL 的推理能力,并进一步完善此框架.

参考文献:

- [1] 李景霞,侯紫峰. Web 服务组合综述[J]. 计算机研究应用,2005,5(12):4-7.
- [2] 汪清明. 基于领域本体的 Web 服务动态组合模型[J]. 计算机应用,2009,7(7):1957-1959.
- [3] 李曼,王大治,杜小勇,等. 基于领域本体的 Web 服务动态组合[J]. 计算机学报, 2005, 28(4): 644-650.
- [4] 付燕宁,刘磊,金成植. 基于服务链的 Web 服务组合方法[J]. 通讯学报, 2007, 28(7): 92-97.
- [5] Stanford University. Protege user documentation[EB/OL]. [2009-10-11]. <http://protege.stanford.edu/doc/users.html>.
- [6] Joel Farrell, Holger Lausen. Semantic annotations for WSDL and XML schema[EB/OL]. [2007-08-28]. www.w3.org/TR/Swawdl.
- [7] Paolucci M, Kawamura T, Payne T R, et al. Semantic matching of web services capabilities[C]// Proceedings of the First International Semantic Web Conference on The Semantic Web (ISWC). Berlin: Heidelberg, 2002: 333-347.
- [8] Eric Prud'hommeaux, Andy Seaborne, Hewlett-Packard Laboratories, Bristol. SPARQL query language for RDF. [EB/OL]. [2008-01-15]. <http://www.w3.org/TR/rdf-sparql-query/>.
- [9] Deepal Jayasinghe. 使用 Axis2 进行 SOA 开发[EB/OL]. [2006-12-28]. <http://www.ibm.com>.
- [10] 韩永国,孙世新. 动态服务组合构造与最优组合服务算法研究[J]. 计算机科学, 2005, 32(12): 104-105.
- [11] Michael K Smith, Chris Welty, Deborah L. McGuinness. OWL web ontology language guide[EB/OL]. [2004-02-10]. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.

作者简介:

闫志刚 男, (1985-), 硕士研究生. 研究方向为面向服务的计算.

胡海涛 男, (1973-), 博士, 硕士生导师. 研究方向为软件集成、服务网格、面向服务计算.

(上接第 196 页)

- [5] Gnawali O, Yarvis M, Heidemann J, et al. Interaction of retransmission, blacklisting, and routing metrics for reliability in sensor network routing[C]// Sensor and Ad Hoc Communications and Networks, First Annual IEEE Communications Society Conference (SECON). Santa Clara, 2004: 34-43.
- [6] Jing Weipeng, Liu Yaqiu, DECDC: An energy-aware route protocol for wireless sensor networks[C]// 2nd international Symposium on Systems and Control in Aerospace

and Astronautics. China: Shenzhen, 2008.

作者简介:

杨显辉 男, (1980-), 硕士研究生. 研究方向为无线传感器网络.

任洪娥 女, (1962-), 教授, 博士生导师. 研究方向为模式识别与智能控制、现代信息技术与信息安全等.

景维鹏 男, (1979-), 讲师. 研究方向为移动计算、容错计算等.