

目标 Web 服务描述本体和服务发现模型

刘传昌, 陈俊亮

(北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

摘 要: 目标 Web 服务描述语言和服务匹配算法是解决 Web 服务发现问题的方法。在分析现有 Web 服务描述语言的基础上, 该文设计了一个目标 Web 服务描述本体, 该本体从功能、性能和语义特征 3 个方面描述 Web 服务, 以提高服务发现的效率和质量为原则, 提出了一个基于 Web 服务特征元素的服务发现模型, 并结合旅游领域, 实现了一个原型系统。

关键词: Web 服务; 服务本体; 服务发现

Goal Web Services Description Ontology and Service Discovery Model

LIU Chuan-chang, CHEN Jun-liang

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

【Abstract】 Goal Web services description languages and service discovery algorithms are methods of dealing with the Web services discovery problem. Based on the analysis of the current Web services description languages, the paper designs a goal Web service description ontology which can depict the functional features, the performance features and the semantic features of Web services. With the principle of enhancing efficiency and quality of the service discovery, it presents a Web services feature elements-based service discovery model, and implements a prototype system in the tourist domain.

【Key words】 Web services; service ontology; service discovery

基于 Web 的商务应用为客户提供了不同类型的 Web 服务, 而这些服务可能具有不同的形式, 并且复杂程度也不同。所谓 Web 服务发现, 就是服务请求者以某种方式在这些不同类型的 Web 服务中, 找到所需的服务, 并完成某些任务。

现有的 Web 服务描述语言不能完整地描述功能特征、性能特征和语义特征:

(1) WSDL 只描述了 Web 服务的功能特征, 缺乏对性能特征和语义特征的描述;

(2) OWL-S 只描述了 Web 服务的功能特征和语义特征, 缺乏对性能特征的描述。

Web 服务描述语言的不足导致现有的 Web 服务发现方法存在 2 个方面的问题:

(1) 采用语法级 Web 服务描述语言, 由于依赖关键字匹配和语义信息的不足, 因此查准率低, 影响了服务复用和服务组合的相容性, 如 IBM, SUN 等公司各类 UDDI 系统^[1] (基于 WSDL 描述语言);

(2) 采用语义级 Web 服务描述语言, 由于缺乏服务质量描述和灵活、有效的服务匹配算法, 因此难以保证服务组合的性能和质量, 如卡内基梅隆大学的 augment UDDI Registry 系统^[2] (基于 DAML 本体的非轻量级 DAML-S 语言), 乔治亚大学的 SpeedR 项目^[3] (基于 OWL 本体的 OWL-S 语言)。

1 目标 Web 服务描述本体

目标 Web 服务需求分为功能需求和非功能需求, 前者描述目标 Web 服务的功能特征, 即接口信息, 如输入和输出参数、前置条件和后置效果等, 后者描述 Web 服务的性能特征, 即服务质量(QoS)等性能指标, 如响应时间、使用费用和可靠性等。

通过把 Web 服务抽象成由输入参数、输出参数、前置条件、效果和 QoS 组成的实体, Web 服务表示为

$WS(I, O, P, E, Q)$

其中, I, O, P, E, Q 分别表示输入参数集、输出参数集、前置条件集、效果集和服务质量参数集。基于上述抽象设计的目标 Web 服务描述本体如图 1 所示。

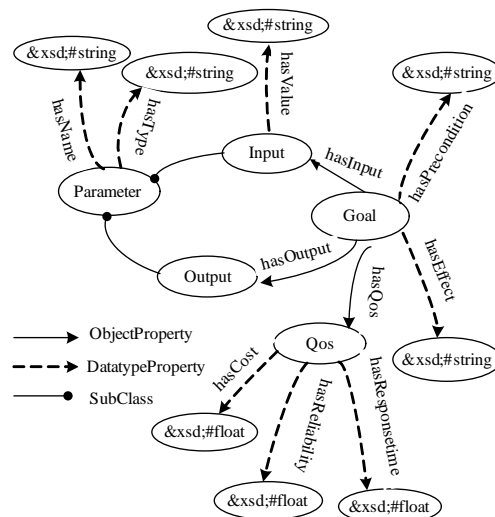


图 1 目标 Web 服务描述本体

基金项目: 国家自然科学基金资助项目(60432010)

作者简介: 刘传昌(1976 -), 男, 博士研究生, 主研方向: 智能 Web 服务; 陈俊亮, 教授、博士生导师

收稿日期: 2006-11-21 **E-mail:** chuanchang.liu@gmail.com

(1)输入参数,属性由 hasName,hasType,hasValue 组成。其中,hasName 是对参数的语义描述,其值为领域本体中预先定义的概念;hasType 是对参数的数据类型描述;hasValue 是为动态验证发现的 Web 服务所设置的参数值。

(2)输出参数,属性由 hasName,hasType 组成,含义与输入参数的相同。

(3)前置条件,用领域本体中的概念表示。

(4)后置效果,用领域本体中的概念表示。

(5)QoS,目标 Web 服务的非功能指标非常多,笔者只考虑响应时间、使用费用和可靠性。目标 Web 服务的 QoS 模型是一个一维向量,可以根据需要进行扩充。

例 要查找机票预定,并取消服务,对目标 Web 服务的功能需求如下:输入参数为 AirTicket,输出参数为 ResultInfo,前置条件为 HaveAirTicket,效果为 NoAirTicket。功能需求描述使用的概念来自旅游领域本体,如图 2 所示(采用 Protégé 的 Ontoviz 插件绘制)。目标 Web 服务的非功能需求为:响应时间小于 5s,费用为 1 元/次,可靠性不低于 0.8。

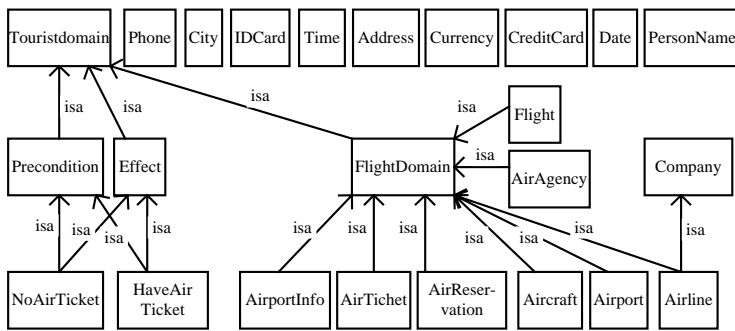


图 2 旅游领域本体的一个片段

根据目标 Web 服务需求描述本体的定义,需求描述为:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:goal="http://skl.bupt.edu.cn/goal.owl#"
  xml:base="http://skl.bupt.edu.cn/goal1.owl"
  xml:domain="http://skl.bupt.edu.cn/touristontology.owl#">
  <goal:Goal rdf:ID="">
    <goal:hasPrecondition>
      <goal:Precondition rdf:ID="precondition1">HaveAirTicket
    </goal:Precondition>
    </goal:hasPrecondition>
    <goal:hasInput>
      <goal:Input rdf:ID="input1">
        <goal:hasName>AirTicket</goal:hasName>
        <goal:hasType>Class</goal:hasType>
        <goal:hasValue>
          <ticketNumber>200</ticketNumber>
          <seatClass>Economy</seatClass>
          <hasDateOfIssue>DateInstance</hasDateOfIssue>
          <hasPersonName>PersonNameInstance</hasPersonName>
          <hasFormOfPayment>CreditCardInstance
        </hasFormOfPayment>
          <hasIDCard>IDCardInstance</hasIDCard>
          <hasFlight>FlightInstance</hasFlight>
        </goal:hasValue>
      </goal:Input>
    </goal:hasInput>
    <goal:hasOutput>
      <goal:Output rdf:ID="output1">
```

```
<goal:hasName>OperationResult</goal:hasName>
<goal:hasType>boolean</goal:hasType>
</goal:Output>
</goal:hasOutput>
<goal:hasEffect>
  <goal:Effect rdf:ID="effect1">NoAirTicket</goal:Effect>
</goal:hasEffect>
<goal:hasQos>
  <goal:hasResponsetime>5</goal:hasResponsetime>
  <goal:hasCost>1</goal:hasCost>
  <goal:hasReliability>0.8</goal:hasReliability>
</goal:hasQos>
</goal:Goal>
</rdf:RDF>
```

2 目标 Web 服务发现模型

2.1 Web 服务特征元素数据库设计

根据目标 Web 服务描述本体和发现算法需要设计的 Web 服务特征元素数据库见图 3,它由 5 个表构成:WebService, InputParameter, OutputParameter, Precondition, Effect。

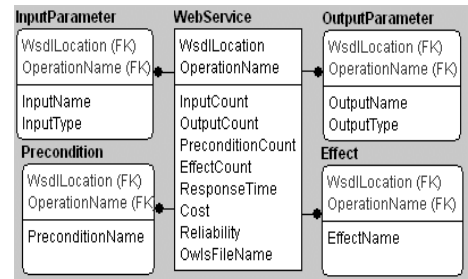


图 3 数据库结构界面

其中,WebService 表存储 Web 服务的各种参数的个数、性能特征以及动态执行时需要的参数;InputParameter 表、OutputParameter 表、Precondition 表和 Effect 表分别存储 Web 服务的输入参数、输出参数、前置条件和效果。

2.2 本体概念间语义匹配

领域本体中概念之间的语义匹配度是计算 Web 服务功能特征匹配度的基础,计算公式定义为

$$\text{Sim}(C_i, C'_i) = \frac{\alpha}{k + \alpha}$$

其中, $\text{Sim}(C_i, C'_i)$ 表示概念 C_i 和 C'_i 的语义匹配度; C_i 和 C'_i 分别属于目标 Web 服务需求描述中和 Web 服务特征数据库中使用的概念, $\text{Sim}(C_i, C'_i)$ 的取值范围为 [0,1]。如果 2 个概念语义相等,则其匹配度为 1;如果 2 个概念之间不满足包含关系,则其匹配度为 0。其中, α 是一个可调节的参数; k 是一个整数,取值随着 C_i 属于 I, O, P, E 的不同而变化。

如果 C_i 属于 I 或 P k 的取值定义见式(1),如果 C_i 属于 O 或 E, k 的取值定义见式(2)。

$$k = \begin{cases} 0 & \text{if } C_i \equiv C'_i \\ 1 & \text{if } C_i \supset C'_i \\ 2 & \text{if } C_i \subset C'_i \end{cases} \quad (1)$$

$$k = \begin{cases} 0 & \text{if } C_i \equiv C'_i \\ 1 & \text{if } C_i \subset C'_i \\ 2 & \text{if } C_i \supset C'_i \end{cases} \quad (2)$$

2.3 Web 服务的功能特征匹配

$WS_g(I_g, O_g, P_g, E_g, Q_g)$ 和 $WS_f(I_f, O_f, P_f, E_f, Q_f)$ 表示任意 2 个 Web

服务，它们之间的功能特征匹配度计算公式定义如下：

$$\text{functionMatch}(WS_g, WS_j) = \lambda_i * \text{SimI}(I_g, I_j) + \lambda_o * \text{SimO}(O_g, O_j) + \lambda_p * \text{SimP}(P_g, P_j) + \lambda_e * \text{SimE}(E_g, E_j)$$

输入参数匹配度为

$$\text{SimI}(I_g, I_j) = \frac{1}{|I_j|} \sum_{i=1}^{|I_j|} \text{Sim}(C_i, C'_i), C_i \in I_g, C'_i \in I_j$$

输出参数匹配度为

$$\text{SimO}(O_g, O_j) = \frac{1}{|O_g|} \sum_{i=1}^{|O_g|} \text{Sim}(C_i, C'_i), C_i \in O_g, C'_i \in O_j$$

前置条件匹配度为

$$\text{SimP}(P_g, P_j) = \frac{1}{|P_j|} \sum_{i=1}^{|P_j|} \text{Sim}(C_i, C'_i), C_i \in P_g, C'_i \in P_j$$

效果匹配度为

$$\text{SimE}(E_g, E_j) = \frac{1}{|E_g|} \sum_{i=1}^{|E_g|} \text{Sim}(C_i, C'_i), C_i \in E_g, C'_i \in E_j$$

其中， $\lambda_i, \lambda_o, \lambda_p, \lambda_e$ 分别表示输入参数匹配度、输出参数匹配度、前置条件匹配度和效果匹配度在 Web 服务功能特征匹配度中所占的比重，且 $\lambda_i + \lambda_o + \lambda_p + \lambda_e = 1, 0 \leq \lambda_i \leq 1, 0 \leq \lambda_o \leq 1, 0 \leq \lambda_p \leq 1, 0 \leq \lambda_e \leq 1$ 。

2.4 Web 服务的非功能特征匹配

非功能特征的匹配结果只有 2 种情况：成功和失败。成功时的匹配值为 1，失败时的匹配值为 0。匹配算法如下：

```
qosMatch(goalWS, WS){
    qosMatchDegree=0;
    if(goalWS.responseTime>=WS.responseTime
    && goalWS.cost>=WS.cost
    && goalWS.reliability<=WS.reliability)
    then qosMatchDegree =1;
    return qosMatchDegree;}
```

2.5 Web 服务匹配算法

Web 服务匹配算法的流程如图 4 所示。

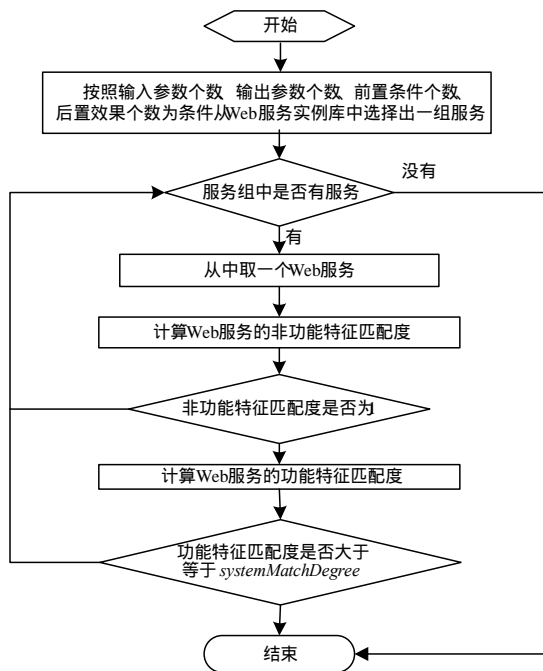


图 4 Web 服务匹配算法流程

实际上，对 Web 服务请求者来说，需要找到一个能够满足其需求的服务，而不是一定要找出最优的那个服务。为找出最优服务而付出巨大的时间代价是不可取的，尤其是在备

选 Web 服务数量很大的情况下。笔者在设计匹配算法时遵循以下 2 个原则：(1)尽可能提高服务发现效率，即缩短服务发现的时间；(2)尽可能提高服务发现的质量，即保证匹配 Web 服务的可用性。Web 服务匹配包括功能特征匹配和非功能特征匹配，与目标 Web 服务相匹配的 Web 服务必须满足以下 2 个条件：(1)功能特征的匹配度不小于系统设置的最小功能特征匹配度值 systemMatchDegree (2)非功能特征的匹配度为 1。

3 原型系统的实现

笔者结合旅游领域，实现了一个 Web 服务发现的原型系统，系统结构如图 5 所示。

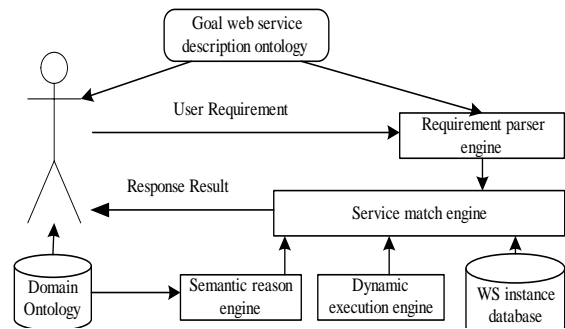


图 5 原型系统的系统结构

旅游领域本体是对旅游领域内概念关系的精确描述，是描述 Web 服务的语义特征的基础。旅游领域本体的一个片段如图 2 所示。服务请求者使用目标 Web 服务描述本体和旅游领域本体定义的概念，来描述 Web 服务需求。需求解析引擎对 Web 服务需求进行解析，并把解析结果作为服务匹配引擎的输入。服务匹配引擎使用该 Web 服务匹配算法，在 WS 实例库中寻找满足需求的 Web 服务。语义推理引擎使用 RACER 推理机制[4]获取旅游领域本体内概念之间的关系，并完成概念间语义匹配度的计算。动态执行引擎使用 Web 服务调用框架(WSIF)动态地执行 Web Services，以此来验证服务的可用性。

4 结束语

Web 服务发现是 Web 服务系统架构中的一个重要部分。笔者设计了一个能从功能特征、性能特征和语义特征 3 个方面来描述 Web 服务需求的目标 Web 服务描述本体，提出了一个基于 Web 服务特征元素的服务发现算法。算法充分利用了 Web 服务的特征元素和领域本体概念间的语义关系，缩短了服务发现的时间，保证了服务的可用性，并结合旅游领域实现了一个原型系统。

智能 Web 服务的目标为：服务发现，服务调用，服务组合，服务执行监控的自动化。因此，研究服务组合及服务执行监控的自动化是未来的主要工作。

参考文献

- Antonio J, Silva C. Quality of Service and Semantic Composition Workflows[D]. Terrell Hall Athens, GA: Department of Computer Science, University of Georgia, 2002.
- Massimo P, Takahiro K, Payne T R, et al. Importing the Semantic Web in UDDI[C]//Proceedings of Web Services, E-business and Semantic Web Workshop, Toronto, Canada. 2002: 225-236.
- Sivashanmugam K, Verma K. Speed-R: Semantic P2P Environment for diverse Web Services Registries[D]. Terrell Hall Athens, GA: Department of Computer Science, University of Georgia, 2002.
- Haarslev V, Möller R. RACER User's Guide and Reference Manual[EB/OL]. (2004-02). <http://www.fh-wedel.de/~mo/racer/>.