

Web 服务描述语言 QWSDL 和服务匹配模型研究

胡建强 邹 鹏 王怀民 周 斌

(国防科学技术大学计算机学院网络技术与信息安全研究所 长沙 410073)

摘 要 现有的 Web 服务发现方法存在两方面的问题:一方面,采用语法级 Web 服务描述语言,因语义信息不足和依赖关键字匹配,容易造成查准率低,影响服务复用和服务组合的相容性.另一方面,采用语义级 Web 服务描述语言,因缺乏服务质量描述和灵活、有效的服务匹配算法,而难以保证服务组合的性能和质量.作者首先在系统研究 Web 服务描述语言的基础上,设计了一种基于服务质量的轻量级 Web 服务描述语言 QWSDL(QoS-based Web Service Description Language),全方位描述 Web 服务的功能、行为约束以及服务质量.其次,提出“三层次,五类型”的匹配模型,引进相似函数来度量松弛匹配的服务相似程度.最后,对比实验证明 QWSDL 和松弛匹配是可行和有效的.

关键词 模型服务质量;本体;松弛匹配;相似函数

中图法分类号 TP311

Research on Web Service Description Language QWSDL and Service Matching Model

HU Jian-Qiang ZOU Peng WANG Huai-Min ZHOU Bin

(Institute of Network Technology & Information Security, School of Computer Science,
National University of Defense Technology, Changsha 410073)

Abstract Currently, syntactic-level service discovery methods, due to insufficient semantic information and the adoption of keyword-based match, easily lead to low precision and difficult reuse and composition of Web services. Meanwhile, semantic-level service discovery methods, due to the adoption of ontologies without quality of service (QoS) and logical reasoning-based match, result in low matching efficiency and failure in guaranteeing the performance and quality of Web service composition. In this paper, on basis of systematic studying on Web service descriptions, a QoS-based Web Service Description Language (QWSDL) is outlined, which supports overall description of the functionality, behavior restraints and QoS attributes. Then, a matching model named “three levels and five kinds” is proposed to discover Web services and similarity functions is introduced to rank Web services in Relaxed match. Finally, QWSDL and the Relaxed match are proved to be feasible and effective by some comparative experiments.

Keywords quality of service; ontology; relaxed match; similarity function

1 引 言

开放的网络化应用和“软件作为服务”的理念

必将导致基于 Internet 环境下软件系统的主要形态、运行方式、生产方式和使用方式发生巨大的变化.未来网络软件的一种趋势表现为构造若干 Web 服务动态组合(composition)、渐趋稳定的软件

应用系统^[1]. Web 服务是一种新型的 Web 应用程序,具有自包含、自描述、模块化和松耦合等特点,可以通过 Web 发布、查找和调用. 简单地说,Web 服务是“一次发布,到处可用”的新型 Internet 组件,支持服务复用和组合,对提高网络软件生产效率和产品质量具有重要意义.

在动态、多样的 Internet 服务市场中,发现合适的服务是实现服务共享、复用的重要前提. Web 服务发现的效果直接关系服务复用的质量,影响到服务组合的相容性(compatibility)和可替换性(substitutability),关系到能否真正实现服务的“即插即用”. 因此,Web 服务发现作为面向服务体系结构的一个重要组成部分,倍受工业界和学术界关注,并以查准率、查全率和匹配效率作为评价其效能的主要指标.

Web 服务描述语言和匹配算法是解决 Web 服务发现问题的两把钥匙. 针对 Web 服务描述信息的丰富程度的不同,目前 Web 服务发现方法的主要研究可分为两类:

(1) 语法级

在描述语言上,其着重描述 Web 服务接口的语法,对行为约束缺乏有效支持;在匹配算法上,大多是基于关键字匹配. 典型系统有 IBM, Microsoft, SUN 等公司各类 UDDI 系统^①(基于预定义分类和 WSDL 描述语言). 语法级服务发现方法实现相对简单,但查准率低,很难保证服务组合的相容性,对 Web 服务复用、验证和管理的支持远远不够.

(2) 语义级

在描述语言上,采用本体论来解决传统语法级 Web 服务描述的异构性,增强对 Web 服务的功能、行为的语义描述;在匹配算法上,依赖于逻辑演绎和推理,具有查准率高、匹配效率不佳、实用性差等特点. 典型研究有卡内基·梅隆大学的 augment UDDI Registry^[2] 系统(基于本体 DAML 的非轻量级语言 DAML-s), 乔治亚大学的 SpeedR 项目^[3](基于本体 OWL 的 Web 服务描述语言 OWL-s^②, 即 DAML-S 后续版本). 总体来说,现有语义级方法采用的 Web 服务描述语言过于复杂而不全面(普遍缺乏服务质量描述),且缺乏灵活、有效的服务匹配算法,不利于组合过程的系统监控、性能分析和动态调整.

针对上述问题,本文首先在分析通用 Web 服务描述模型的基础上,引入本体和服务质量模型,提出基于服务质量的轻量级 Web 服务描述语言 QWSDL

(QoS based Web Service Description Language),全方位支持 Web 服务功能、行为约束和服务质量的描述;其次,将现有匹配算法归纳为基本、基调和规约三个层次,精确、可替代、包含、相交和松弛匹配五种类型,着重运用相似函数构造松弛匹配算法,在保证较高查准率和查全率的前提下,显著地提升服务匹配的效率和提高了服务匹配的灵活性;最后,构造对比实验证明匹配模型的可行性和有效性.

2 Web 服务描述模型

2.1 通用 Web 服务描述模型

在面向服务体系结构中,Web 服务描述被认为是服务提供者和请求者的一种约定. 虽然这种约定在完整性和精确程度上会有差别,但最终目的是让服务请求者精确地、无二义地理解服务的功能和行为,使每个 Web 服务发挥最大的效能. 事实上,Web 服务描述通常不仅仅是简单地定义一系列可用的操作,还描述在特殊情况下请求者对服务所期望的行为,约束服务允许的状态,从而指导服务动态组合和交互.

定义 1. Web 服务描述模型 $WS = \{S, C, P\}$, 其中 S 是基本描述,即服务名称和文本描述; C 是服务功能描述,即服务功能、行为描述; P 是属性描述,主要包括代价(cost)、响应时间(time)、服务优先级(priorities)等非功能属性.

在上述描述模型中,内容描述 S 往往是概要性的描述;功能描述 C 是服务请求者判断 Web 服务能否满足其功能需求的主要依据,希望行为足够严格以及交互时有精确的和可预见的输出;属性描述 P 为服务请求者选择服务提供积极、有意义的参考. 目前大多数 Web 服务描述语言都自发地遵循 $\{S, C, P\}$ 模型:

(1) UDDI, WSDL 是由 IBM, Microsoft 等公司向 W3C 组织提交并被工业界广泛采用的标准. UDDI 使用关键字和预定分类管理 WSDL; WSDL 除简单定义服务名称和文本描述外,还采用变量类型(Types)、消息(Message)、端口类型(PortType)、绑定(Binding)和端口(Port)五种主要元素来描述接口的语法. 优点:采用 XML Schema 机制,已具备编

① UDDI.org. UDDI spec TC, Version 3.0.2, 2004. http://www.uddi.org/pubs/uddi_v3.htm

② The OWL Services Coalition. Semantic markup for Web services (OWL-S), Version 1.0, 2004

译和浏览工具支持. 不足:存在服务描述的“两个异构性”(见 2.2 节),对服务的行为约束和属性描述缺乏进一步支持,不考虑服务组合上下文描述.

(2)SCDL^[4]是一种轻量级 Web 服务能力描述语言,基本描述包括服务名称和文本描述,并使用数据类型(Type)、输入(Input)、输出(Output)、前置条件(Pre-conditions)和后置条件(Post-conditions)等元素来描述服务功能和行为语义. 优点:支持服务组合上下文描述;着力刻画 Web 服务行为的约束关系;不足:存在服务描述的两个异构性,缺乏对包括服务质量在内的非功能属性描述.

(3)OWL-s 是基于本体 OWL 的描述语言,采用明确的、计算机可以理解的语言标识来描述 Web 服务,试图利用语义描述和逻辑推理最终实现服务匹配的自动化和智能化. Service Profile 是 OWL-s 重要组成部分,除基本描述之外,使用输入(Input)、输出(Output)、前提条件(Precondition)和结果(Result)等元素描述 Web 服务. 优点:引进本体描述 Web 服务功能和行为语义,支持部分服务组合上下文描述. 不足:对 Web 服务的行为约束和非功能属性描述是部分的、非完备的;没有明确地给出单个 Web 服务的服务质量模型;语言比较复杂,普通用户不易掌握.

2.2 基于服务质量 Web 服务描述语言 QWSDL

由 2.1 节分析可知,目前主流的描述语言对通用 Web 服务描述模型 $\{S, C, P\}$ 尚缺乏统一、全面的支持,主要表现在以下几个方面:

(1)语法级 Web 服务描述语言普遍存在“两个异构性”^[5],即结构异构(structural heterogeneity)和语义异构(semantic heterogeneity). 结构异构是指用不同的数据结构和类层次来定义接口参数;语义异构是指使用不同术语(terminology)来表达同一接口参数. 缺少同样的语义解释(interpretation)会引发语义冲突.

(2)Web 服务行为描述信息不足,只重视服务功能的语法,忽略对 Web 服务的行为约束的刻画,不可避免地导致服务组合不相容问题.

定义 2(Web 服务组合不相容). 设 Web 服务某个 XML 消息类型 message 的出现称为一个事件,若存在 Web 服务 WS_1 ,要求输入事件 $msgX$ 先于输出事件 $msgY$;在 Web 服务 WS_2 中,要求输入事件 $msgY$ 先于输出事件 $msgX$,输入和输出的次序关系导致 WS_1 和 WS_2 无法有效组合,称为服务组合不相容.

例 1. 假设售票服务 OrderingTicket 和客户服务 Customer 进行服务组合. Customer 有一个输入 *ticket* 和一个输出 *money*; OrderingTicket 有一个输入 *money* 和一个输出 *ticket*. 不考虑服务行为约束,因为客户服务 Customer 的输出是售票服务 OrderingTicket 的输入,容易认为服务组合是有效的. 事实上存在 Customer 输入前提是先得票 *known* (*ticket*)才能保证输出钱 *money*, OrderingTicket 输入前提是先收钱 *known* (*money*)才能保证输出 *ticket*,导致两个服务死锁而服务组合不相容.

(3)普遍缺少对服务质量的描述,不利于服务选择和服务组合的性能评价,影响 Web 服务复用的质量. 服务质量显式表达服务可用性和可靠性信息,尤其在开放网格服务体系结构 OGSA (Open Grid Service Architecture)、移动自组网络 (mobile Ad hoc network) 等服务协同应用中扮演极其重要的角色^[6].

针对上述问题构造一种全新的基于服务质量 Web 服务描述语言 QWSDL,引入本体将异地术语映射到一共享本体来解决二义性,从而避免语义冲突;遵循通用 Web 服务描述模型,在保持服务名称和文本描述的基础上,通过前置约束和后置约束进一步增强对服务行为的描述;基于服务质量模型 *QoS* 完整描述 Web 服务的质量相关属性,从而有利于服务组合的质量分析和监控.

(1)QWSDL 的主要内容:

①基本描述. 采用元素 *serviceName* 和 *textDescription* 分别定义 Web 服务名称和文本描述;

②基调描述. 通过元素 *Input* 和 *Output* 描述类型和方向来定义服务功能,即表达接口的基调. 为了使系统具有良好可扩展性和避免“结构异构”和“语义异构”,利用本体来描述消息类型 *Type*. 与传统的 IDL 和 WSDL 语法级描述语言不同的是,还支持行为约束的描述,即采用元素 *Pre-constraint* 和 *Post-constraint* 分别定义事前约束和事后约束.

③服务质量描述. 基于服务质量模型 *QoS* = $\langle time, cost, reliability, \dots, Fidelity \rangle$ 定义服务质量属性,满足服务组合的质量需求和系统质量分析.

定义 3. 服务质量模型 *QoS* = $\langle time, cost, reliability, \dots, Fidelity \rangle$ 是一个可扩充向量,其中时间 (*time*) 是一个度量性能的通用尺度,包括服务请求到达和完成的时间,包括延迟时间、处理时间;费用 (*cost*) 是指执行 Grid/Web 服务的相关费用;可靠性 (*reliability*) 定义为成功执行次数与调度执行总次

数的比率,即 $reliability = \frac{s}{s+f}$, 其中 s 表示成功执行次数, f 为失败执行次数; 真实度 **Fidelity** = $\langle fid_1, fid_2, \dots, fid_n \rangle$ 定义为 n 个真实属性组成的向量, 其中第 i 个真实属性被用于确定服务满足第 i 个服务质量属性的真实程度。

例 2. 设本体 Thing 存在的概念类关系如图 1 所示. 为简化描述的复杂性且能清晰说明 QWSDL 的结构, 省略了底层本体的描述并采用“:”表示概念类限定关系. 采用 QWSDL 描述例 1 中的售票服务 OrderingTicket.

```

serviceName: Ordering Ticket Service
textDescription: Ordering air tickets from Flight Company.
Type: Date={year,month,day}; Money: real;
      Ticket: string
Input: yourMoney: Money; ValidateDate: Date
Output: changeMoney: Money; ticket: Ticket
Pre-constraint: Money
Post-constraint: Ticket
QoS Model:
    timemin = 100, timeavg = 120, timemax = 139;
    costmin = 42, costmin = 63, costmax = 83;
    reliability = 0.88;
    fid1 = 0.89, fid2 = 0.94, fid3 = 0.87

```

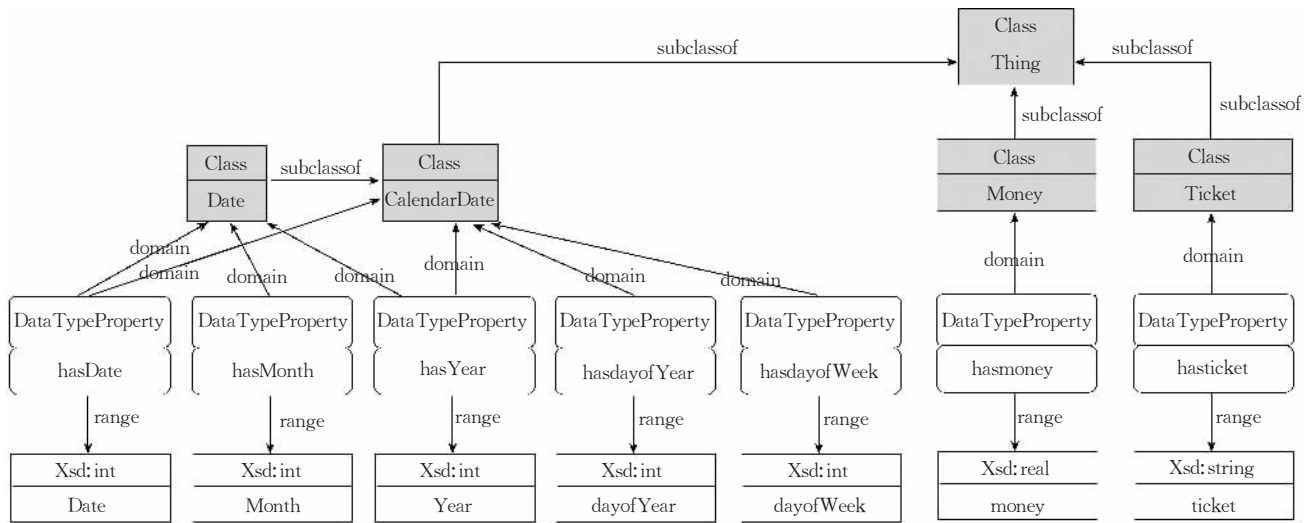


图 1 本体 Thing

(2) QWSDL 的主要优点:

① 良好的开放性和可扩充性: 主要表现在基于本体和可扩充服务质量模型;

② 全方位描述服务: 遵循通用服务描述模型 $\{S, C, P\}$, 突出服务行为约束、条件和服务质量模型, 避免潜在的服务组合不相容. 这些 Web 服务应具备的约束和条件为服务组合提供了部分共享上下文, 可以有效地指导和监督服务组合。

QWSDL 不是将服务视为简单的功能“黑盒”, 而是通过清晰暴露服务的行为前提和后果, 避免服务请求者发现和选择与自身存在行为前提、后果相互制约的服务, 从而阻止发生服务组合不相容的情况. 如例 1 中的 OrderingTicket 的行为前提 (Pre-constraint: Money) 是 Customer 行为后果 (Pre-constraint: Money), 而 Customer 服务行为前提 (Pre-constraint: Ticket) 是 OrderingTicket 的行为后果 (Pre-constraint: Ticket), 两者互相制约, 无法保证两个服务成功执行和交互。

3 Web 服务匹配模型

3.1 服务匹配类型

在“服务请求者—服务提供者—服务注册中心”的 Web 服务架构中, 服务提供者向服务注册中心发布遵循 $\{S, C, P\}$ 的广告描述; 服务注册中心负责维护和管理服务广告描述; 服务请求者向服务注册中心发出查询请求时, 服务注册中心将请求映射为服务请求描述并与广告描述进行匹配. 于是, 服务发现的问题就转化为请求描述与广告描述之间的匹配问题. 基于 QWSDL 描述语言和本体论, 按照匹配精确程度差异性, 可定义如下匹配类型。

定义 4 (Web 服务匹配类型). 设存在请求描述 A 与广告描述 A' , 若 A 与 A' 是两个相同的概念, 即 $A \equiv A'$, 称为精确匹配 $Exact(A, A')$; 若 A 是 A' 的子概念, 即 $A \subseteq A'$, 称为可替代匹配 $PlugIn(A, A')$; 若 A 是 A' 的超级概念, 即 $A' \subseteq A$, 称为包含匹配

$Subsume(A, A')$; 若 A' 与 A 的交集是可满足的, 称为相交匹配 $Overlap(A, A')$; 其它情形匹配失败。

由定义 4 可知, 精确匹配是可替代匹配和包含匹配的特殊情形, 并且各种匹配类型之间存在如下关系。

命题 1. “ \Rightarrow ”表示蕴含关系, “ \vee ”表示逻辑并关系, 则存在 $Exact(A, A') \Rightarrow PlugIn(A, A')$; $Exact(A, A') \vee PlugIn(A, A') \vee Subsume(A, A') \Rightarrow Overlap(A, A')$ 。

命题 2. “ \rightarrow ”表示匹配精确度下降, 则各服务匹配类型存在 $Exact(A, A') \rightarrow PlugIn(A, A') \rightarrow Subsume(A, A') \rightarrow Overlap(A, A')$ 。

以上定义的四种匹配类型, 从本体概念类之间的各种关系(如 $supclass, subclass$ 等)出发, 以服务发现的自动化和智能化为目标, 采用逻辑推理方法进行服务匹配, 具有查准率(precision)高的优点, 不可避免地存在查全率(recall)低、实现复杂、灵活性差等不足。

通常情况下, 服务请求者的需求是多层次的, 服务发现需要一定的灵活的匹配能力, 保证一定的查准率尽可能提高查全率。同时需要兼顾请求描述往往是不完全描述的, 对服务匹配应有一定的张弛能力, 不仅要求返回匹配结果还要求能返回相应的匹配程度, 为用户复用、组合提供有意义的参考信息。应对这些问题, 定义如下松弛匹配(relaxed match)。

定义 5(松弛匹配). 采用数值计算度量请求描述 A 与广告描述 A' 的相似程度, 称为松弛匹配 $Relaxed(A, A')$ 。

3.2 服务匹配层次

传统语法级服务描述采用关键字匹配, 并主要针对服务名称和文本描述, 因而查准率低。服务请求者的需求是多层次的, 不仅要求考虑通用 Web 服务描述模型中的服务名称、文本描述和服务功能, 还要考虑服务行为的事前约束和事后约束以及服务质量。借鉴传统的构件匹配理论^[7,8], 按是否涉及服务行为和匹配严格程度, 将服务匹配划分为基本匹配、基调匹配(signature match)和规约匹配(specification match)三个层次, 很容易得到如下定义。

定义 6(Web 服务匹配). 设有请求描述 A 与广告描述 A' , 则存在 A 与 A' 的服务匹配断言

$Match: Web\ Service, Web\ Service \rightarrow Bool;$

$Match(A, A') = match_{bas}(A, A') \wedge match_{sig}(A, A') \wedge match_{spec}(A, A')$ 。

其中基本匹配 $match_{bas}(A, A')$ 表示比较两服务的服

务名字和文本描述的相关性; 基调匹配 $match_{sig}(A, A')$ 表示比较两服务的输入(输出)类型和方向的相关性; 规约匹配 $match_{spec}(A, A')$ 表示比较两服务行为和约束信息的相关性。

由定义 6 和通用服务模型可知, 服务匹配可以是一个逐步精化过程: 经由基本匹配筛选的候选服务再参与基调匹配, 合格的服务进一步参与规约匹配。这就可以满足服务请求者不同层次的需要。另一方面, 这三个匹配层次采用的算法可以是相互独立的, 匹配类型可以互不相同。对于基本匹配一般采用关键字计算相似程度, 即松弛匹配; 而对于基调匹配和规约匹配则依赖于逻辑推理的方法。

(1) 类型推断规则^[4,9]

对于基调匹配, 除考虑输入输出方向一致外, 类型推断是判断请求描述 A 与广告描述 A' 是否满足精确、可替代、包含和相交匹配的逻辑推理基础。

定义 7(子类型推理规则). 设类型 t_1, t_2 分别是 A 和 A' 的输入或输出类型; $t_1 \triangleleft t_2$ 表示 t_1 是 t_2 的子类型; $t_1 \equiv t_2$ 表示同样类型当且仅当满足 $t_1 \triangleleft t_2$ 且 $t_2 \triangleleft t_1$, 并且具有: (a) 同一性. $t_1 = t_2$; (b) 对称性. $t_1 | t_2 \equiv t_2 | t_1$; (c) 结合性. $(t_1 | t_2) | t_3 \equiv t_1 | (t_2 | t_3)$ 。则存在类型 t_1 是类型 t_2 的子类型推断规则:

① $t_1 \triangleleft t_2$; ② $\frac{t_1 \equiv t_2}{t_1 \triangleright t_2}$; ③ 若 t_1, t_2 是集合, $\frac{t_1 \subseteq t_2}{t_1 \triangleright t_2}$;

④ $t_1 \triangleleft t_1 | t_2$; ⑤ $t_2 \triangleleft t_1 | t_2$; ⑥ $\frac{t_1 \triangleleft t_2, s_1 \triangleleft s_2}{t_1 | s_1 \triangleleft t_2 | s_2}$;

⑦ $\frac{t_1 \triangleleft t_2, s_1 \triangleleft s_2}{(t_1, s_1) \triangleleft (t_2, s_2)}$; ⑧ $\frac{t_1 \triangleleft t_2}{Setof(t_1) \triangleleft Setof(t_2)}$ 。

(2) 规约匹配

借鉴规约匹配理论^[8]和文献^[9], 可知判断请求描述 A 是否规约匹配广告描述 A' 的条件是: 基调匹配的基础上能否同时满足: ① 请求描述 A 的前置条件、约束是否逻辑蕴含广告描述 A' ; ② 广告描述 A' 的前置条件、约束是否逻辑蕴含 A 。形式表达为如下定义。

定义 8. 请求描述 A 与广告描述 A' 满足规约匹配当且仅当同时满足规约匹配和 $(pre-A \Rightarrow pre-A') \wedge (post-A' \Rightarrow post-A)$ 关系。

例 3. 设存在基于本体 Thing 的售票服务 OrderingAirTicket, 其描述为

serviceName: Ordering Air Ticket Service

textDescription: Ordering air tickets from Flight Agent Company.

Type: CalendarDate = {dayofWeek, dayofYear, year, month, day};

Money: real; Ticket: string

Input: yourMoney: Money;

ValidateCalendarDate: CalendarDate

Output: changeMoney: Money; ticket: Ticket

Pre-constraint: Money

Post-constraint: Ticket

QoS Model:

$time_{min} = 98, time_{avg} = 117, time_{max} = 140;$

$cost_{min} = 40, cost_{min} = 62, cost_{max} = 83;$

$reliability = 0.95;$

$fid_1 = 0.96, fid_2 = 0.92, fid_3 = 0.91$

考虑例 2 中的 OrderingTicket 为请求描述与 OrderingAirTicket 为广告描述的规约匹配过程. 对于 OrderingTicket 服务的输入 $t_1 = \{\text{yourMoney}\}$, $t_2 = \{\text{ValidateDate}\}$ 和 OrderingAirTicket 服务的输入 $t'_1 = \{\text{yourMoney}\}$, $t'_2 = \{\text{ValidateCalendarDate}\}$ 的类型方向是一致的, 显然 $t_1 \equiv t'_1$, 满足 $t_1 \triangleleft t'_1$; 因为 $\{\text{year, month, day}\} \subseteq \{\text{dayofWeek, dayofYear, year, month, day}\}$, 运用规则③可知 $t_2 \triangleleft t'_2$. 对于两服务的输出 $t_3 = \{\text{yourMoney}\}$, $t_4 = \{\text{ticket}\}$ 和 $t'_3 = \{\text{yourMoney}\}$, $t'_4 = \{\text{ticket}\}$ 运用的类型方向是一致的, 显然 $t_3 \equiv t'_3$, $t_4 \equiv t'_4$, 满足 $t_3 \triangleleft t'_3$, $t_4 \triangleleft t'_4$. 由定义 4 和定义 6 可知, 满足可替代匹配 PlugIn(OrderingTicket, OrderingAirTicket).

4 相似函数

通常情况下, 用户对服务的理解是局部和片面的, 精确、可替代和包含等匹配类型大多是建立在完整信息的严格推理上, 因而缺少灵活性和实用性差. 松弛匹配采用数值计算来度量服务的相似程度, 通过判断是否大于预定阈值来选择服务. 按照定义 6, 构造了如下的相似函数:

$$Sim(A, A') = \xi_1 \times Sim_{bas}(A, A') + \xi_2 \times Sim_{sig}(A, A') + \xi_3 \times Sim_{spec}(A, A'), \sum_i \xi_i = 1, 0 \leq \xi_i \leq 1, i = 1, 2, 3 \quad (1)$$

其中 Sim_{bas} , Sim_{sig} 和 Sim_{spec} 分别是基本信息(包括服务名字、基本描述)、服务基调和规约匹配相似函数; ξ_i 是相似函数对应的权值. 并且定义 Sim_{qos} 服务质量匹配相似函数为选择服务提供参考.

(1) 基本相似函数

$$Sim_{bas}(A, A') = \mu_1 \times Sim_{name}(A, A') + \mu_2 \times Sim_{text}(A, A'), \sum_i \mu_i = 1, 0 \leq \mu_i \leq 1, i = 1, 2 \quad (2)$$

其中 Sim_{name} , Sim_{text} 分别是采用关键字匹配算法得

到的基于服务名称、文本描述的相似函数; μ_i 是权值, 体现服务请求者对服务名字或基本描述的重视程度.

(2) 基调和规约相似函数

设 a^p, b^q 是分别来自本体 p 和 q 的实体类, Sim_w , Sim_u 和 Sim_n 分别表示本体类基于同义词、属性和语义关系的相似函数, ω_w, ω_u 和 ω_n 是权值, 则 a^p, b^q 的实体类相似函数

$$Sim(a^p, b^q) = \omega_w \times Sim_n(a^p, b^q) + \omega_u \times Sim_u(a^p, b^q) + \omega_w \times Sim_n(a^p, b^q), \omega_w, \omega_u, \omega_n \geq 0 \quad (3)$$

基于 Tversky^[10] 模型和 Andrea 相似函数及集合论, 设 A, B 是 a^p, b^q 本体类描述集合(同义词集、属性和语义关系集), $\cap, |$ 分别是集合的并、模运算, δ 是非公共特性的相对重要程度, 则 Sim_w, Sim_u 和 Sim_n 都可以通过下列公式计算^[11]

$$Sim_{w,u,n}(a^p, b^q) = \frac{|A \cap B|}{|A \cap B| + \delta |A/B| + (1 - \delta)(a^p, b^q) |B/A|}, 0 \leq \delta \leq 1 \quad (4)$$

基调匹配需要计算输入、输出类型的相似度, 可定义基调相似函数为

$$Sim_{sig}(A, A') = \mu_1 \times Sim_{in}(A.input, A'.input) + \mu_2 \times Sim_{out}(A.output, A'.output), \sum_i \mu_i = 1, 0 \leq \mu_i \leq 1, i = 1, 2 \quad (5)$$

其中 Sim_{in}, Sim_{out} 分别是两个服务关于输入集(input sets)、输出集(output sets)的相似函数, μ_i 是权值. 可以通过式(4)和以下递归公式(7), (8)计算

$$Sim_{in}(A.input, A'.input) = Sim_{in}(A.input - a^{A.input}, A'.input - b^{A'.input}) + \mu_i \times Sim_{in}(a^{A.input}, b^{A'.input}) \quad (6)$$

$$Sim_{out}(A.output, A'.output) = Sim_{out}(A.output - a^{A.output}, A'.output - b^{A'.output}) + \mu_i \times Sim_{out}(a^{A.output}, b^{A'.output}) \quad (7)$$

其中 $\sum_i \mu_i = 1, 0 \leq \mu_i \leq 1$.

规约匹配需要计算前置条件和约束、后置条件和约束的相似性, 定义规约相似函数为

$$Sim_{spec}(A, A') = \mu_1 \times Sim_{pre-}(A.pre-, A'.pre-) + \mu_2 \times Sim_{post-}(A.post-, A'.post-), \sum_i \mu_i = 1, 0 \leq \mu_i \leq 1, i = 1, 2 \quad (8)$$

其中 Sim_{in}, Sim_{out} 分别是两个服务关于前置条件和约束集、后置条件和约束集的相似函数, μ_i 是权值.

可以通过式(4)和类似(6),(7)的递归公式计算。

(3) 服务质量相似函数

$$Sim_{qos(i)}(A, A') = \sum_i \mu_i Sim_{qos(i)}(A, A'),$$

$$\sum_i \mu_i = 1, 0 \leq \mu_i \leq 1, i = 1, 2, \dots, n \quad (9)$$

其中 $Sim_{qos(i)}(A, A')$ 表示对于第 i 个服务质量属性(包括时间、费用、可靠性等)的相似函数,

$$Sim_{qos(i)}(A, A') = \sqrt{Sim_{min} Sim_{avg} Sim_{max}} \left(Sim_{min} = 1 - \frac{abs(fid_i A.qos(i)_{min} - fid'_i A'.qos(i)_{min})}{fid'_i A'.qos(i)_{min}}, \right.$$

$$Sim_{avg} = 1 - \frac{abs(fid_i A.qos(i)_{avg} - fid'_i A'.qos(i)_{avg})}{fid'_i A'.qos(i)_{avg}},$$

$$Sim_{max} = 1 - \left. \frac{abs(fid_i A.qos(i)_{max} - fid'_i A'.qos(i)_{max})}{fid'_i A'.qos(i)_{max}} \right);$$

Fidelity = $\{fid_i\}_{i=1,2,\dots,n}$, 其中 $0 \leq fid_i \leq 1$ 表示每个服务质量对应的真实度, 反映用户对服务属性的依赖程度, μ_i 是权值。

例 4. 设例 2 中的 OrderingTicket 为请求描述, 例 3 中 OrderingAirTicket 为广告描述, 采用相似函数分别计算基本、基调和规约三个匹配以及 QoS 的相似度。

(1) 基本匹配

采用关键字匹配算法, 容易得到

$$Sim_{name}(OrderingTicket, OrderingAirTicket) = 0.89,$$

$$Sim_{text}(OrderingTicket, OrderingAirTicket) = 0.97,$$

取权值 $\mu_1 = \mu_2 = 0.5$, 由式(1)得到基本匹配相似度

$$Sim_{bas} = 0.93.$$

(2) 基调匹配

根据递归公式(6)和式(4), 可知

$$Sim_{in}(OrderingTicket, OrderingAirTicket) = 0.5 \times Sim(Money, Money) + 0.5 \times Sim(Date, ValidateCalendarDate) = 0.5(1 + 3/5) = 0.8;$$

$$Sim_{out}(OrderingTicket, OrderingAirTicket) = 0.5 \times Sim(Money, Money) + 0.5 \times Sim(Ticket, Ticket) = 1.$$

因此, 取权值 $\mu_1 = \mu_2 = 0.5$, 由式(1)得到基本匹配相似度 $Sim_{bas} = 0.9$ 。

同理可得, 规约匹配的相似度为 1。

(3) 服务质量

对于时间相似度, 因为

$$Sim_{min} = 1 - \frac{abs(0.89 \times 100 - 0.96 \times 98)}{0.96 \times 0.98} = 0.9460,$$

$$Sim_{avg} = 1 - \frac{abs(0.94 \times 120 - 0.92 \times 117)}{0.92 \times 117} = 0.9521,$$

$$Sim_{max} = 1 - \frac{abs(139 - 140)}{140} = 0.9492,$$

所以

$$Sim_{time} = \sqrt{Sim_{min} Sim_{avg} Sim_{max}} = 0.9246;$$

同理可得

$$Sim_{cost} = 0.9461, \quad Sim_{reliability} = 0.8484.$$

由式(9)可得,

$$Sim_{qos}(OrderingTicket, OrderingAirTicket) = 0.9064.$$

5 实验结果及分析

为了检验本文所提出的 QWSDL 描述语言及其松弛匹配在实践应用中的可行性和有效性, 设计并实现一个 Web 服务发现的原型系统 StarWSDS (StarWebService Discovery System). 实验选取 Internet 上的 Web 服务实例, 如 Google, Amazon 和 GlobalWeath 等知名服务共 120 个作为样本, 分别考查遵循 UDDI 规范实现系统 JAXR Registry、集成 DAML-s 语言的 augment UDDI Registry、原型系统 StarWSDS 的查全率、查准率和性能. 为了尽可能说明匹配效率和减少网络因素的影响, 所有系统都运行在 Windows 2000, JDK1.4.1 环境下, 统一通过 StarWebService 2.0^① 发布和查找服务: (1) 使用 WSDL 描述和基于分类 NAICS 标准将服务发布到 JAXR Registry^②, 查询结果为基于关键字匹配和符合 NAICS 分类的 Web 服务; (2) 使用 DAML-s 描述服务并发布到 augment UDDI Registry^[2], 查询结果为满足基于 Prolog 推理的精确匹配的 Web 服务; (3) 使用 QWSDL 描述服务并发布到 StarWSDS, 查询结果为满足一定阈值的松弛匹配的 Web 服务。

图 2 给出三个系统的 Web 服务查全率、查准率和性能的实验对比结果, 总结如下:

(1) JAXR Registry 具有最高的查全率, 但查准率却最低. 相反, Augment UDDI Registry 具有最高的查准率, 但查全率相对较低. 这表明基于关键字匹配算法和基于逻辑推理的匹配都不能很好满足服务发现的多方面的需求。

(2) StarWSDS 具有与 JAXR Registry 相近的

① <http://www.starmiddleware.net/ws>

② <http://www.sun.com/xml/jaxr>

查全率和 Augment UDDI Registry 相近的查准率(预定义阈值为 0.90),平均查准率为 84%,平均查全率为 95%。这表明基于相似函数的松弛匹配和 QWSDL 语言是可行的。

(3) JAXR Registry 的响应曲线 S_1 稳定 80ms 左右, Augment UDDI Registry 的平均响应时间为

739ms, 曲线 S_3 有明显的抖动, StarWSDS 的响应曲线 S_2 在 324ms 左右波动。这体现匹配算法的复杂性,即 Prolog 逻辑推理比相似度计算复杂,相似度计算比关键字匹配复杂,同时表明松弛匹配较好地满足服务发现效率的要求,当然 StarWSDS 系统还存在性能优化空间。

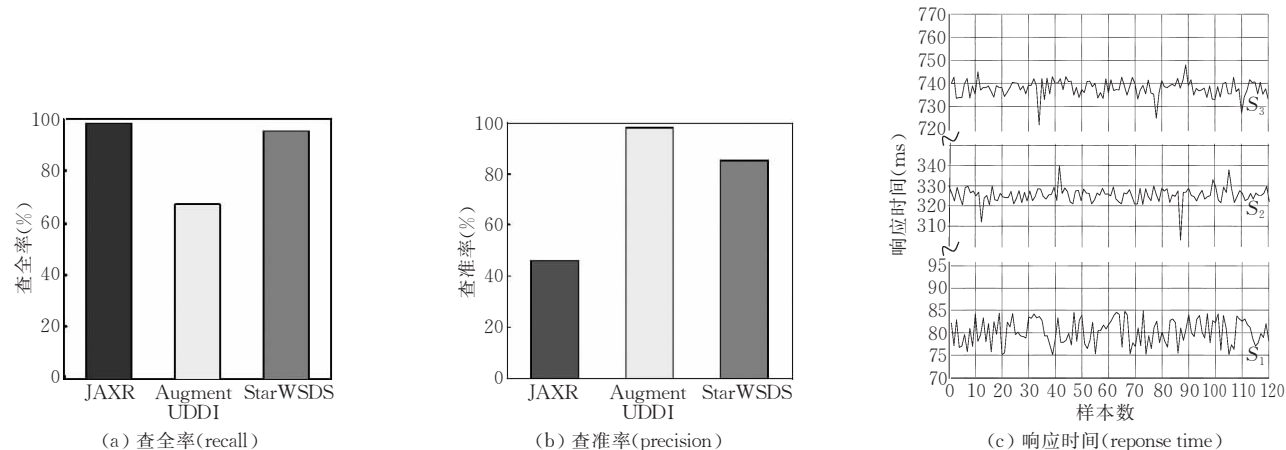


图 2 3 种不同匹配模式的查全率、查准率和性能实验结果

6 结 论

本文提出一种全新的 Web 服务描述语言,全方位支持 Web 服务的功能和行为约束及服务质量描述,并将现有匹配算法归纳为“三层次,五类型”的服务匹配模型。针对现有服务匹配算法过多依赖于逻辑推理和缺乏匹配灵活性,构造了相似函数度量服务相似程度。该方法将匹配问题转化为数值计算,在大大降低计算复杂性的同时提高匹配的灵活性和易用性,并具备很强的松弛匹配能力。

通过在原型系统上进行的服务发现实验,表明基于相似函数的松弛匹配具有较高的服务查全率和查准率以及足够的服务匹配效率。这证明了本文所设计的 Web 服务描述语言、匹配模型及算法在实践应用中是可行的、有效的。

致 谢 感谢国防科学技术大学计算机学院的郭长国博士,空军指挥学院的张志伟博士和湖南大学的罗凤梅对本文提出的宝贵意见!

参 考 文 献

1 Jorge Antonio, Cardoso Silva. Quality of service and semantic composition workflows [Ph. D. Dissertation]. Department of

Computer Science, University of Georgia, GA, 2002

2 Paolucci Massimo, Kawamura Takahiro, Payne Terry R., Sycara Katia. Importing the semantic Web in UDDI. In: Proceedings of Web Services, E-business and Semantic Web Workshop (CAiSE Workshop), Toronto, Canada, 2002, 225~236

3 Sivashanmugam K., Verma K., Mulye R., Zhong Z.. Speed-R: Semantic P2P environment for diverse Web services registries. Final Presentation, CSCSI: 8350, Enterprise Integration, Department of Computer Science, University of Georgia, GA, 2002

4 Gao Xiang, Yang Jian, Papazoglou Midke P.. The capability matching of Web services. In: Proceedings of the IEEE Four International Symposium on Multimedia Software Engineering (MSE'02), California, USA, 2002, 56~63

5 Cardoso Jorge, Sheth Amit. Semantic e-Workflow composition. Journal of Intelligent Information Systems (JIIS), 2003, 21 (3): 191~225

6 Sheth A., Cardoso J., Miller J., Kochut K.. QoS for service-oriented middleware. In: Proceedings of the 6th World Multi-conference on Systemics, Cybernetics and Informatics, Orlando, FL, 2002, 528~534

7 Zaremski A. M., Wing J. M.. Specification matching of software components. ACM Transactions on Software Engineering and Methodology, 1997, 6(4): 333~369

8 Zaremski Amy Moormann. Signature and specification matching [Ph. D. dissertation]. School of Computer Science, Carnegie Mellon University (CS-CMU-96-103), Pittsburgh, USA, 1996

9 Hu Jian-Qiang, Guo Chang-Guo, Wang Huai-Min, Jia Yan,

- Zou Peng. The matching model of Web services based on QoS description. In: Proceedings of the 10th Joint International Computer Conference (JICC'04), Kunming, 2004, 266~271
- 10 Tversky A.. Features of similarity. *Psychological Review*, 1977, 84(4): 327~352
- 11 Rodriguez M. Andrea, Egenhofer Max J.. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 2003, 15(2): 442~456
- 12 Zhang Bo, Feng Yu-Lin, Huang Tao. The architecture description language XADL and architecture mismatch checking. *Journal of Software*, 2002, 13(12): 2238~2243(in Chinese) (张波,冯玉琳,黄涛.体系结构描述语言 XADL 和组合失配检测. *软件学报*, 2002, 13(12): 2238~2243)
- 13 Xu Ru-Zhi, Qian Le-Qiu, Cheng Jian-Ping, Wang Yuan-Feng, Zhu San-Yuan. Research on matching algorithm for XML-based software component query. *Journal of Software*, 2003, 14(7): 1195~1202(in Chinese) (徐如志,钱乐秋,程建平,王渊峰,朱三元.基于 XML 的软件构件查询匹配算法研究. *软件学报*, 2003, 14(7): 1195~1202)



HU Jian-Qiang, born in 1971, Ph. D. candidate. His current research interests include distributed object-oriented technologies and software engineering.

ZOU Peng, born in 1957, professor, Ph. D. supervisor. His current research interests include real-time systems,

software engineering and distributed object-oriented technologies.

WANG Huai-Min, born in 1962, professor, Ph. D. supervisor. Her current research interests include software engineering, Agent technologies and distributed object-oriented technologies.

ZHOU Bin, born in 1971, Ph. D.. His current research interests include distributed object-oriented technologies and software engineering.

Background

Faced with decreasing time-to-market and increasing requirement volatility, software-development processes are increasingly relying on reuse of existing software. Moreover, Web applications based on the Web are developed from localization to globalization, from B2C (business-to-customer) to B2B (business-to-business), from centralized fashion to decentralized fashion. Web services are Internet-based software components that provide standard interfaces as well as communication protocols aiming at efficient and effective service sharing and integrating. A critical step in the process of reusing and integrating existing web service components is the discovery of potentially relevant components based on functional, operational and QoS requirements. The authors' projects are supported by the National Natural Science Founda-

tion of China and the National High Technology Research and Development Program (863 Program) of China and aims to provide a flexible and efficient mechanism for Web service publication and discovery. The project names are "Research on key technologies of application integration based on Web services", "The architecture, protocol and implementation mechanism for the new generation network middleware".

The research group has presented a dynamic, scalable Web service discovery infrastructure called Stratus. The major aim of this paper is to enhance the precision and efficiency of Web service discovery and guide Web service selection. Moreover, it lays down a sound theoretical basis for the further research framework of process-based Web service composition.