

基于二维 QoS 模型的 Web 服务组合^{*})

孔维梁 刘清堂 杨宗凯 谈国新 何婷婷

(华中师范大学教育信息技术工程研究中心 武汉 430079)

摘要 在实际应用中,单个 Web 服务通常无法满足复杂应用的需求,如何组合已有的服务,从而提供更强大更完整的商业功能已成为此领域的研究热点。然而现行的 SOA 架构中,Web 服务组合方法中很少考虑 Web 服务质量(QoS)问题,或者对 QoS 考虑得不够全面,这样,组合出的服务不能确保能够满足用户的 QoS 需求。针对这一问题,提出了 Web 服务的二维 QoS 模型,并使用了改进的 UDDI 规范,在此基础上给出了基于 QoS 的 Web 服务组合的有色 Petri 网组合策略。在满足用户组合服务的功能需求的同时,也满足了用户对服务质量 QoS 的需求,实现了需求服务的优化。

关键词 Web 服务,二维 QoS 模型,有色 Petri 网,Web 服务组合,服务质量

Composition of Web Services Based on Two-dimension QoS Model

KONG Wei-liang LIU Qing-tang YANG Zong-kai TAN Guo-xin HE Ting-ting

(Engineering & Research Center for Information Technology on Education, Huazhong Normal University, Wuhan 430079, China)

Abstract In practical application, a single Web service is usually unable to meet the needs of complex applications. How to combine the existing service to provide a more complete and more powerful commercial function is becoming the key of this filed. However, in the existing framework of the SOA, the combination of Web services rarely considers Web quality of service (QoS). Thus, the composite service can not promise to meets users demands for QoS. On this issue, we proposed two-dimension Web services QoS Model, used the improved UDDI specification, then put forward the QoS-based Web services combination of colored Petri net portfolio strategy. It not only meets user's demands for the service functions but also meets user's QoS needs, and achieves the optimization of the demand services.

Keywords Web service, Two-dimension QoS model, Colored Petri net, Composition of Web services, Quality of service

1 引言

Web 服务是一个由 Web 服务描述语言(如 WSDL)描述的一种自包含、自描述、模块化的程序,并且可以由标准网络协议(如 SOAP)通过 HTTP 访问。它吸取了分布式计算、网格计算和 XML 等各种技术的优点,具有高度的互操作性、跨平台性以及松散耦合等特点,所以近年来 Web 服务得到了学术界和工业界的极大关注,然而在实际应用中,单个 Web 服务通常无法满足复杂应用的需求,因此,如何将现有的 Web 服务连接与合并起来,生成复杂的复合 Web 服务,从而提供更强大、更完整的商业功能成为为现有的服务集合增加价值的关键。

在 Web 服务组合的过程中,满足相同功能需求而具有不同 QoS 参数(如执行时间、费用、可靠性等)的 Web 服务存在多个,如何从中选择满足各服务结点功能需求的具体服务,形成一个可执行的服务链来完成用户的需求就成为 Web 服务组合中的一个关键问题。Kim 等人^[1]研究了 2003—2004 年公共 Web 服务的使用情况,大约只有 34% 的服务可用,而且每周大约有 16% 的已注册可用的 Web 服务失效,这就导致查找到的服务信息的实际可用性差,其主要原因在于没有考虑服务过滤和选择的问题,进而不能确保其发现的服务能够满足用户的 QoS 需求。

针对以上问题,通过分析组合服务的特性,本文提出了一种基于 QoS 的 Web 服务组合模型,包括反映基本服务质量的基本服务 QoS 和全面评价组合服务质量的组合服务 QoS。本文第 2 节介绍了 QoS 计算方法及组合结构;第 3 节介绍了基于 QoS 的应用模型;第 4 节给出了基于 QoS 的 Web 服务自动组合。

2 二维 QoS 模型及组合结构

2.1 Web 服务的二维 QoS 模型

Web 服务的 QoS 模型从很多方面刻画其非功能特性,它不仅有助于指导 Web 服务的高质量开发,而且用于支持服务的过滤和选择。一般来说,客户端调用远程目标服务进程的过程可以细化为:请求消息通过网络协议正确抵达服务提供者结点,进而抵达其上的 Web 服务容器,最终抵达 Web 服务进程。现有 Web 服务 QoS 模型单纯地从 Web 服务进程的角度研究其 QoS 属性,而忽视了更为基础的服务质量因素,即涵盖服务调用前两个环节的服务质量。因此本文提出了服务的二维 QoS 模型,从客观和主观 2 个维度对 Web 服务进行 QoS 建模。

(1)客观维度:表示服务是否正常,最近 T 时间段内是否可用,这与 Web 服务容器容量、最近 T 时间段内 Web 服务容器可用主存、处理器的频率、已有服务数、最大带宽以及最近

^{*})基金项目:湖北省自然科学基金项目创新团队计划(2006ABC011),国家“十一五”科技支撑计划重大项目资助(2006BAH02A24),部分受教育部/国家外国专家局高等学校学科创新引智计划(B07042)及 IBM 大学合作项目资助。孔维梁 硕士研究生;刘清堂 教授;杨宗凯 教授,博士生导师;谈国新 教授;何婷婷 教授,博士生导师。

T 时间段内带宽占用率等^[2]有关,这是一个定性值,表示 T 时间段内服务可用或者不可用,例如 T 时间段内服务器中服务数已满,服务则不可用。这有利于服务监控器对服务的实时监控,参见第3部分。

(2)主观维度:主观维度包括执行时间、费用、可靠性、信誉度等^[3-5],主观服务质量模型 $QoS = (time, cost, reliability, \dots, reputation)$ 是一个可扩充向量,这些质量的计算公式和说明如下表1。

表1 Web 服务质量计算公式及说明

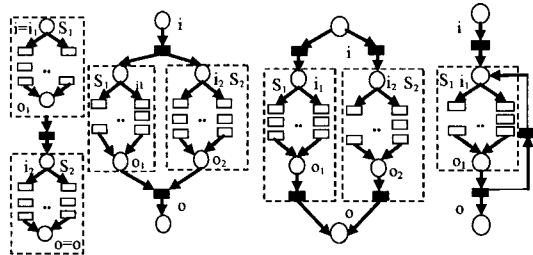
质量	说明	计算公式
QoS _{tim}	服务请求从发送到完成时间	$QoS_{tim} = T_{tran} + T_{wait} + T_{pro}$
QoS _{cs}	调用 Web 服务的相关费用	由服务提供者发布
QoS _{rel}	服务能正确执行的程度	$QoS_{rel} = \frac{\sum_{i=1}^n Rel_i}{n}$
QoS _{rep} ^[6]	服务的可信程序,取决于用户的使用情况	$QoS_{rep} = \frac{\sum_{i=1}^m R_i}{m}$

表1中,服务传输时间(T_{tran})可以通过该服务执行的历史记录确定,比如取几次传输时间的平均值(或者可不用,因为这个无法精确确定);服务等待时间(T_{wait})指服务从到达服务提供者端到服务开始执行的时间,这个时间是由服务提供者端通过计算当时状态给出;服务处理时间(T_{pro})是由服务提供者给出; n 是过去一段时间内服务被调用的次数; Rel_i 服务在所声明的时间内是否成功执行,若成功则 $Rel_i = 1$,否则, $Rel_i = 0$; m 是用户对服务的评价次数, R_i 是用户对服务的评价值。

2.2 组合结构及 QoS 计算方法

假设 Web 服务的主观维度包括4种 QoS 参数,即时间 T (time)、费用 C (cost)、可靠性 Rel (reliability)、信誉度 Rep (reputation),设 cs 为多个单服务 si 形成的组合服务,则 si 和 cs 的服务质量模型分别为 $Q_s = (Obj_i, (T_i, C_i, Rep_i, Rel_i))$, $Q_{cs} = (Obj_{cs}, (T_{cs}, C_{cs}, Rep_{cs}, Rel_{cs}))$,对于其中定性的客观维度可以对其进行量化,比如,如果服务可用, $Obj = 1$,否则 $Obj = 0$ 。

本文给出了基于有色 Petri 网^[7]的服务组合流程的4种基本模型及其相应的 QoS 计算方法,大部分服务组合流程都可以由这4种基本模型^[8,9,15]组合而成,组合流程的 QoS 参数也可以通过基本模型的 QoS 评价方法来获取,其中:变迁代表服务;变迁的前置库所和后置库所分别表示服务的输入和输出;静操作在图中用黑色方框表示相应的变迁。



(a)顺序结构 (b)并发结构 (c)选择结构 (d)循环结构

图1 基本组合结构

(1)顺序结构 顺序操作 $Seq(S_1, S_2)$ 指服务 S_1 和服务 S_2 顺序执行,顺序结构的典型情况是一个服务的输入依赖另一个服务的输出,给定 S_1 和 S_2 , $Seq(S_1, S_2)$ 的有色 Petri 网图形式表示如图1(a)所示。

$Obj_{cs} = \prod_{i=1}^n Obj_i$,但一般来说组合要用的单服务要满足 $Obj_i = 1$,所以 Obj_{cs} 可以认为是1。

$$T_{cs} = \sum_{i=1}^n T_i, C_{cs} = \sum_{i=1}^n C_i, Rel_{cs} = \prod_{i=1}^n Rel_i, Rep_{cs} = \frac{\sum_{i=1}^n Rep_i}{n}$$

(2)并发结构 并发操作 $Con(S_1, S_2)$ 指服务 S_1 和 S_2 并发执行,典型的并发结构是,一些互不相关的原子服务可以合并到一起形成一个大的合成服务。

$Obj_{cs} = \prod_{i=1}^n Obj_i$,但一般来说组合要用的单服务要满足 $Obj_i = 1$,所以 Obj_{cs} 可以认为是1。

$$T_{cs} = \text{Max}(T_1, T_2, T_3, \dots, T_n), C_{cs} = \sum_{i=1}^n C_i,$$

$$Rel_{cs} = \text{Min}(Rel_1, Rel_2, \dots, Rel_n), Rep_{cs} = \frac{\sum_{i=1}^n Rep_i}{n}$$

(3)选择结构 选择操作 $Choice(S_1, S_2)$ 使得服务 S_1 或 S_2 执行。一旦其中一个被执行,另一个就不再执行。

设第 i 个分支被选中的概率是 α_i , $\sum_{i=1}^n \alpha_i = 1$, $Obj_{cs} = \prod_{i=1}^n Obj_i \alpha_i$,但一般来说组合要用的单服务要满足 $Obj_i = 1$,所以 Obj_{cs} 可以认为是1。

$$T_{cs} = \sum_{i=1}^n T_i \alpha_i, C_{cs} = \sum_{i=1}^n C_i \alpha_i, Rel_{cs} = \sum_{i=1}^n Rel_i \alpha_i, Rep_{cs} = \sum_{i=1}^n Rep_i \alpha_i$$

(4)循环结构 循环操作 $Loop(S)$ 使得服务 S 执行一定的次数 n 。循环结构典型的情况是质量控制,服务需要执行多次。

$Obj_{cs} = \prod_{i=1}^n Obj_i$,但一般来说组合要用的单服务要满足 $Obj_i = 1$,所以 Obj_{cs} 可以认为是1。

$$T_{cs} = n * T_i, C_{cs} = n * C_i, Rel_{cs} = \prod_{i=1}^n Rel_i, Rep_{cs} = \frac{\sum_{i=1}^n Rep_i}{n}$$

3 基于二维 QoS 模型的应用模型

目前的 Web 服务组合采用可扩展标记语言(XML)定义了 Web 服务协议栈,通过 SOAP, WSDL, UDDI, BPEL4WS 等开放协议和标准,提供面向互联网应用的统一服务注册、发现、绑定及集成调用机制。虽然开放标准的采用使 Web 服务具有很好的互操作性及自描述、模块化的特性,允许采用任何编程语言在任何平台上开发松耦合的 Web 服务,并通过适当的服务组合支持面向服务应用的集成和开发;然而目前版本的 UDDI 规范更多地关注于基于功能约束的服务发现问题,没有为用户提供完成这些工作所需的足够的 QoS 信息。为了克服现有 UDDI 模型存在的这些缺点,本文强烈建议在保证与 UDDI 规范兼容的前提下,可以对其实现规范进行必要的扩展,如图2所示^[10]。

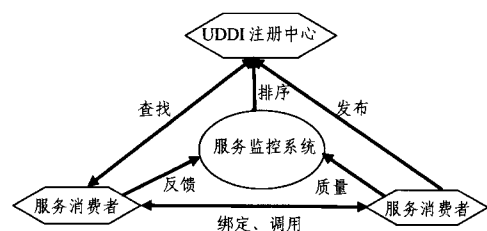


图2 UDD 扩展规范

在 UDDI 新模型中,有 4 种基本角色: Web 服务提供者、Web 服务消费者、UDDI 服务注册中心以及服务监控系统。Web 服务提供者提供一组 Web 服务,执行服务请求,在部署好服务之后,向 UDDI 服务注册中心发布符合 UDDI 数据结构规范的服务描述信息并向监控系统提供服务的 QoS 信息。Web 服务消费者会向 UDDI 服务注册中心提出服务发现请求并获得反馈,并且发现请求中可以包含对目标服务 QoS 指标的需求,这样可以根据约束条件从 UDDI 查找服务,接着与目标服务提供者进行协商,最后进行协议绑定,并调用目标服务。UDDI 服务注册中心提供服务注册和发现功能,并在继承原有功能的基础上,支持基于 QoS 约束的服务选择。服务监控系统接受服务提供者的 QoS 信息或服务消费者的 QoS 反馈,并根据反馈信息对 Web 的信誉度进行动态评估和调整,同时它还会动态地监视和反映服务的状态,如 T 时间段内服务容器中的服务数达到一定数量,则 T 时间段内服务就不可用等。

4 基于二维 QoS 模型的 Web 服务组合

4.1 Web 服务描述模型

Web 服务描述被认为是服务提供者和请求者的一种约定,使得服务请求者精确地、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,从而指导服务动态组合和交互。

定义 1 通用 Web 服务描述模型^[11] $WS = \{ S, C, P \}$, 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,主要包括费用(cost)、时间(time)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,常用的几种 Web 服务描述语言及其功能优缺点如表 2。

表 2 Web 服务描述模型比较

描述语言	优点	缺点
UDDI, WSDL	采用 XML Schema 机制,已具备编译和浏览工具支持	不考虑服务组合上下文描述,对服务的行为约束和属性描述缺乏进一步支持
SCDL ^[12]	支持服务组合上下文描述,支持 Web 服务行为的约束	缺乏对包括服务质量在内的非功能属性描述
OWL-s	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	对 Web 服务的行为约束和非功能属性描述是部分的、不完整的

本文采用的 Web 服务描述模型如下:

(1)基本描述 S :采用元素 Name 和 Description 分别定义 Web 服务名称和文本描述。

(2)功能描述 C :通过元素 Input 和 Output 描述服务的输入、输出接口。为了使系统具有良好可扩展性和支持上下文描述,利用本体来描述消息类型 Type。与传统的 UDDI 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-Cons 和 Post-Cons 分别定义前置约束和后置约束。

(3)属性描述 P :基于服务质量模型 $QoS = (Obj, (time,$

$cost, reliability, \dots, reputation))$ 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。

例 1 用上述 Web 服务描述模型描述 Buying Photocard 服务如下:

```

ServiceName:Buying Photocard
ServiceDescription:Buying Photocard 201
Type:Money;cash;Type:string;Number:string
Input:Money;Money;TypeofCard:Type
Output:changeMoney;Money;CardNumber:Number
Preconstraint:Money
Postconstraint:Number
QoS:
Obj = 1;
time=120;cost=21;reliability=0.81;reputation=0.65

```

4.2 基于 QoS 的 Web 服务组合图

依赖图可以提供一种自然直观的方式描述实体之间的复杂交互,因此本文采用依赖图的表示方法来表示服务之间的关系^[13]。

定义 2 一个依赖图 $G = (N, E, W)$ 是一个有向图,其中 N 是节点集合,表示服务的集合; E 是边的集合,表示服务之间的依赖关系; W 是权重的集合,表示边的权重,在本文中它表示 QoS 的值。

一个基本的服务可以表示为图 3 所示。

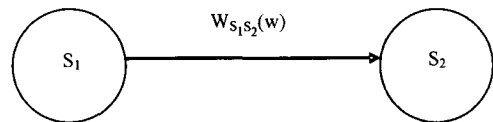


图 3 Web 服务的图表示

其中, S_1 表示输入服务, S_2 表示输出服务, w 表示服务的 QoS 值,对于 Web 服务 WS ,有多个非功能属性,本文把 Web 服务的这些非功能属性看作是一个综合性能的函数 $F(WS) = F(Obj, time, cost, reliability, reputation) = (1 - Obj) * NAN + \epsilon_1 * time + \epsilon_2 * cost + \epsilon_3 * (1 - reliability) + \epsilon_4 * (1 - reputation)$, $\sum_{i=1}^4 \epsilon_i = 1$,其中 NAN 表示一个足够大的正数, ϵ 表示各属性的权重,此权重可由用户指定,这样综合性能值越小表示服务的 QoS 越好。一个简单的基于 QoS 的 Web 服务图如图 4 所示。这样,Web 服务的组合是置换成求 S_N 到 S_M 可达集,求最优解也就成了求 S_N 到 S_M 的最短路径。

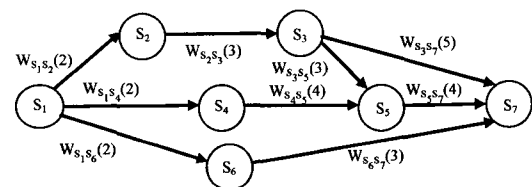


图 4 基于 QoS 的 Web 服务图

4.3 基于 QoS 的 Web 服务组合

为了快速计算,很多研究者^[14]倾向于使用 Petri 网,因为 Petri 网非常适合描述 Web 服务中的流关系。然而,基于库所/变迁网的 Web 服务合成模型^[15]不能体现输入输出数据的类型和个性,因此无法分析合成服务的有效性,有些专家提出 Web 服务组合的有色 Petri 网模型 CPN ^[9],所以本文即采

用基于有色 Petri 网的 Web 服务组合。要实现 Web 服务组合, 必须首先判断哪些服务以何种结构可以合成到一起。在 CPN 模型中, 是根据颜色 (数据类型) 来区分的, 因此设计者可以根据输入、输出库的颜色来判断合成服务的有效性。与面向对象程序中一样, 颜色被定义为可以继承的类, 例如, ‘C’和‘D’分别继承于‘A’和‘B’, 则表示为‘A-C’和‘B-D’, 因此, 当一个服务需要颜色为‘A’的输入时, ‘A’和‘A-C’都符合要求, 然而, 当一个服务需要颜色为‘A-C’的输入时, 只有‘A-C’符合要求。同时为了增强对服务行为约束的描述和避免语义异构和结构异构的产生, 如 4.1 所述, 我们引入了本体描述信息和前置约束 Pre-Cons 和后置约束 Post-Cons, 因此要保证服务的有效性还需要满足两个约束条件, 即服务的输入要满足前置约束 Pre-Cons, 服务的输出要符合后置约束 Post-Cons。通过以上分析, 可以得到以下规则:

(1) 如果合成服务 $S = Seq(S_1, S_2)$ 是有效的, 则必有 $C(i_2) \subseteq C(o_1)$;

$pre(S_2) \subseteq post(S_1)$ 且 $pre(S_1) \cap post(S_2) = \emptyset$;

(2) 如果合成服务 $S = Conc(S_1, S_2)$ 是有效的, 则必有 $C(i_1) \cup C(i_2) \subseteq C(i)$ 且 $C(o) = C(o_1) \cup C(o_2)$;

$(pre(S_1) \cup pre(S_2)) \subseteq post(S_p)$ 且 $pre(S_p) \cap (post(S_1) \cup post(S_2)) = \emptyset$, S_p 是 S 的前置服务;

(3) 如果合成服务 $S = Choice(S_1, S_2)$ 是有效的, 则必有 $C(i_1) \cup C(i_2) \subseteq C(i)$ 且 $C(o) = C(o_1) \cup C(o_2)$;

$(pre(S_1) \cup pre(S_2)) \subseteq post(S_p)$ 且 $pre(S_p) \cap (post(S_1) \cup post(S_2)) = \emptyset$, S_p 是 S 的前置服务;

(4) 如果合成服务 $S = Loop(S_1)$ 是有效的, 则必有 $C(i_1) \subseteq C(o_1)$; $pre(S_1) = post(S_1)$ 。

Web 服务组合图的生成: 对于给定的输入服务 $I(I \neq \emptyset, I = \emptyset)$ 和输出服务 $O(O \neq \emptyset, O = \emptyset)$, 服务组合图 $G = (N, E, W)$ 是由下述算法构造的图结构, 它的每一个节点都是一个服务, 定义 $\Phi \subseteq N$, 且 $N \cap \Phi = \{I, O\}$, 算法步骤如下:

(1) 定义 $MaxL = T$ (一般为 5-10)

(2) 初始状态, $N = \{I, O\}, E = \emptyset, W = \emptyset, \Phi = \emptyset$

(3) $\alpha = search(O)$, 令 $\alpha \subseteq \Phi, \langle \alpha, O \rangle \in E, W(\alpha, O) \in W$

(4) 节点 $X, (X \in \Phi)$ 且 $(X \neq O)$ 且 $(\cdot X = \emptyset)$

如果: $X = null$

结束

否则:

如果: $pace(X, O) > MaxL$

Goto(5)

否则:

Goto(6)

(5) 令 $\langle null, X \rangle \in E, W(null, X) = NAN$

(6) $\beta = search(X)$, 使 $\beta \subseteq \Phi, \langle \beta, X \rangle \in E, W(\beta, X) \in W$

(7) Goto(2)

其中, $search(X)$ 表示对于节点 X 查找符合条件的节点集, $pace(X, O)$ 表示节点 X 和节点 O 之间的步长。

5 实验结果及分析

为了检验本文所提出的基于二维 QoS 模型的 Web 服务组合在实践应用中的可行性和有效性, 设计并实现了一个 Web 服务组合系统, 此次实验从 Web 服务二维 QoS 模型中选择如下关键指标对多个候选服务进行选择: 1) 主观维度, T 时间段内服务是否可用; 2) 客观维度, 包括时间 T(time)、费用 C(cost)、可靠性 Rel(reliability)、信誉度 Rep(reputation)。

服务选择的主要目标是更好地为用户找到可用的服务, 为了对集成该算法的扩展 UDDI 原型系统进行评估, 定义并使用下列度量指标:

1) 查准率: 服务请求者通过 UDDI 查找符合条件的目标服务组合, 发现一个目标服务组合并成功调用的个数占调用的总数的比例。

2) 负载: 具备相同功能的多个原子服务在一定时间间隔内接受到的服务访问次数。

第 1 组评估的目的是为了说明本文提出的基于二维 QoS 服务组合算法对从 UDDI 上获取的服务信息准确性的影响。在系统测试阶段分别采用了普通组合算法 (也是利用有色 Petri 网生成服务组合图, 不过是在选择组合服务时只是利用普通的 QoS 模型, 即只考虑服务质量的客观维度, 并且“时间”属性只是考虑服务提供者发布的信息, 不考虑根据用户反馈信息动态更新“可靠性”属性值) 和基于二维 QoS 服务组合算法。从运行日志文件中分别提取 7 个测试周期的统计数据, 其发现一个目标服务组合并成功调用的个数占服务调用的总数的比例, 如图 5 所示。

第 2 组评估的目的是为了说明本文提出的基于二维 QoS 服务组合算法对多个原子服务的访问负载产生的影响。在系统测试阶段, 在一个测试周期内, 对采用普通服务组合和基于二维 QoS 组合两种情况下, 组合中的 3 个相同功能服务的访问负载各进行了 3 次随机抽样, 其访问负载的分布情况如图 6 所示。

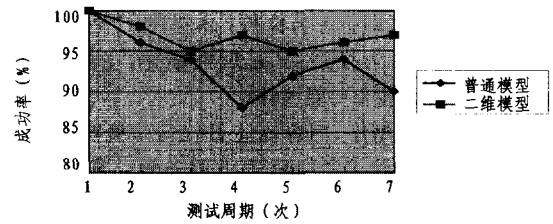


图 5 服务调用成功率

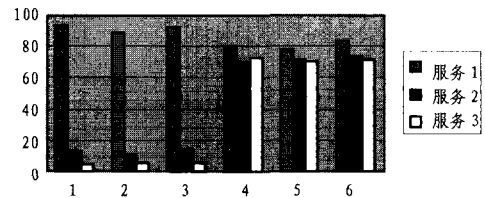


图 6 3 个 Web 服务的负载

结束语 Web 服务作为新兴技术有着广泛的应用前景, Web 服务是可以组合的, 如何动态地把现存的各种 Web 服务整合起来以形成新的、满足不同用户需求的、增值的复杂服务已成为新的应用需求和研究热点。本文在综合分析已经出现的几种 QoS 模型和组合模型的基础上, 提出了自己的改进方法: 基于 QoS 的 Web 服务组合。与已有的其它组合方法相比, 本文考虑了服务的质量属性, 以及提出了在现行 UDDI 标准上的改进方法, 使得可以对 Web 服务进行动态的监控, 在服务发现方面引入了语义和行为约束, 以期组合产生具有较好 QoS 的服务组合。

本文虽然在服务发现过程引入了语义和约束条件, 但只是一些简单表面的应用, 还需要不断改进; 其次, Web 服务自动组合的研究也是当今的热门话题, 如何实现服务的自动发

(下转第 143 页)

互信息度和概念词模式词相关性度量,特征维数是3;子集之二仅不包含互信息度量,特征维数是4。实验结果如图1所示。

如图1所示,在仅使用概念表示能力度量的情况下,3种分类器的精度都已达到83%以上。但是J48和LibSVM的精度随着输入维数增加不但没有增长反而降低。出现这种现象的原因是:LibSVM是把输入的特征数据映射到高维空间进行分类,J48是以决策树的方式分类。随着维数的增加,LibSVM在训练时,样本在高维空间变得非常稀疏;而J48训练出的决策树随着输入维数的增加决策树的复杂性上升,产生过拟合的现象。所以,在我们保持原有训练样本不变的情况下,随着维数的增加,精度出现不同程度的下滑。和LibSVM与J48不同,AdaBoostM1训练出的模型是一些弱分类器的线性组合,对训练样本数要求不高。从AdaBoostM1算法精度的变化可以看出,在分别增加了概念词和模式词相关性度量和互信息度量之后,精度约有1.3%的提升。

结束语 本文提出一种介词-动词结构的模式获取方法。通过介词集合和动词集合的笛卡尔积运算构造候选模式,并通过“什么”疑问词构造查询项,获取用于评价PV模式的句子。综合考虑PV模式的表示能力和模式词概念词之间的相关性,构造了6个数值特征,并采用监督学习的方法对PV模式进行预测。用交叉验证的方法估计出精度、召回率和F度量均在0.85左右。

在获取的PV模式的精度达到一定程度的基础上,还需要在以下几个方面做进一步的研究:

第一,从深度的角度来说,还需要增加对主语部分概念(SUB-CPT)或者宾语部分概念(OBJ-CPT)的语义限制,从而提高用PV模式获取语义关系的精度。

第二,从广度的角度考虑,虽然自然语言中的很多知识能通过PV模式进行表述,但是除了PV模式以外还存在很多其他结构的模式,这就需要考虑生成其他结构的模式候选集

合的方法。

参考文献

- [1] Wang Shi, Cao Yanan, Cao Xiny, et al. Learning Concepts from Text Based on the Inner-constructive Model // Proceedings of 2nd International Conference on Knowledge Science Engineering and Management. Melbourne, Australia, 2007
- [2] 余蕾,曹存根. 基于Web语料的概念获取系统的研究与实现. 计算机科学, 2007, 34(2): 161-165, 195
- [3] Hearst M. Automatic acquisition of hyponyms from large text corpora // Proceedings of the 14th Conference on Computational Linguistics. Nantes, France, 1992
- [4] Hearst M. Automated discovery of wordnet relations // Fellbaum C, ed. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press, 1998: 131-151
- [5] Riloff E. Automatically generating extraction patterns from untagged text // Proceedings of the 13rd National Conference on Artificial Intelligence. Oregon, USA, 1996
- [6] 刘磊,曹存根,王海涛,等. 一种基于“是一个”模式的下位概念获取方法. 计算机科学, 2006, 33(9): 161-165
- [7] Tian Guogang, Cao Cungen, Liu Lie, et al. MFC: A Method of Co-referent Relation Acquisition from Large-scale Chinese Corpora // Proceedings of Third International Conference on Fuzzy Systems and Knowledge Discovery. Xi'an, China, 2006
- [8] Surdeanu M, Turmo J, Ageno A. A Hybrid Approach for the Acquisition of Information Extraction Patterns // Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics. Trento, Italy, 2006
- [9] Nello C, John S-T, Huma L. Latent Semantic Kernels // Proceedings of the 18th International Conference on Machine Learning. MA, USA, 2001
- [10] Ian W, Eibe F. Data Mining: Practical Machine Learning Tools and Techniques. Second Edition. Burlington, MA: Morgan Kaufmann, 2005
- [11] Kim S M, Rosu M C. A survey of public Web services // Feldman S I, Uretsky M, Najork M, et al. eds. Proc. of the 13th Int'l Conf. on World Wide Web (WWW 2004). New York: ACM Press, 2004. 312-313
- [12] Tian M, Gramm A, Ritter H, et al. Efficient selection and monitoring of QoS-aware Web services with the WS-QoS framework // Liu J, Cercone N, eds. Proc. of the IEEE Int'l Conf. on Web Intelligence (WI 2004). New York: IEEE Press, 2004: 152-158
- [13] Bryson J, Martin D, McIlraith S, et al. Toward behavioral intelligence in the semantic Web. IEEE Computer, 2002, 25(11): 48-54
- [14] Cardoso J, Sheth A, Miller J, et al. Quality of service for workflows and Web service process. Journal of Web Semantic, 2004, 13: 281-308
- [15] Zeng L Z, Benatallah B, Ngu A H H, et al. QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327
- [16] 代鲸, 杨雷, 张斌, 等. 支持组合服务选取的QoS模型及优化求解[J]. 计算机学报, 2006, 29(7): 1167-1178
- [17] Jensen K. An introduction to the Theoretical Aspects of Colored Petri Nets[J]. Lecture Notes in Computer Science, 1994, 803: 230-272
- [18] Tan Z, Lin C, Yin H, et al. Approximate Performance Analysis of Web Services Flow Using Stochastic Petri Net[J]. Lecture Notes in Computer Science, 2004, 3251: 192-200
- [19] Narayanan S, McIlraith S. An analysis and simulation of Web Services[J]. Computer Networks, 2003, 42(5): 675-693
- [20] 杨胜文, 史美林. 一种支持QoS约束的Web服务发现模型[J]. 计算机学报, 2005, 28(4): 589-594
- [21] 胡建强, 邹鹏, 王怀民, 等. Web服务描述语言QWSDL和服务匹配模型研究[J]. 计算机学报, 2005, 28(4): 505-513
- [22] Gao Xiang, Yang Jian, Papazoglou M P, et al. The capability matching of Web services // Proceedings of the IEEE Four International Symposium on Multimedia Software Engineering (MSE'02). California, USA, 2002: 56-63
- [23] Ashemian V H S. A Graph2Based Approach to Web Services Composition // Proceedings of the 2005 Symposium on Applications and the Internet Society. 2003: 191-200
- [24] Milanovic N, Malek M. Current Solutions for Web Service Composition[J]. IEEE Internet Computing, 2004, 8(6): 51-59
- [25] Hamadi R, Benatallah B. A Petri Net - Based Model for Web Service Composition [C] // Proceedings of the 14th Australasian Database Conference. Adelaide, Australia, Australian Computer

(上接第134页)

现、调用、组合以及自动优化也是今后工作的重点。

参考文献