按需集成服务的发现算法研究

李 伟,吴岳忠,李长云
(湖南工业大学,湖南 株洲 412008)

摘 要:为满足用户按需交换动态信息的要求,Web服务应当可以动态按需协同工作,完成任务,提供信息,即动态地按需集成Web服务。服务发现是服务组合的关键问题。针对现有的服务发现方法要么依赖关键字查找,查准率低;要么基于语义描述,难以保证服务组合的性能和质量。在场景驱动的业务模型构造方法基础上,构建了一个基于场景的按需服务集成方法框架,提出了一种依据"先筛选后匹配"原则的服务自动发现方法。测试验证该方法具有高查全率和低查错率的特性。

关键词:按需集成;业务模型;服务发现
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2008)06-0036-04

Research of Based-Requirement Integrating Services Discovering

LI Wei, WU Yue-zhong, LI Chang-yun
(Hunan University of Technology, Zhuzhou 412008, China)

Abstract: In order to satisfy the acquirements of based-requirements exchanging dynamic information, Web services should cooperate dynamically, accomplish task and provide information, seemed as based-requirements integrating Web services. services discovering is the key of services combining. Based on the method of scenario-driven business model constructing, a frame of scenario-driven services integrating and a method of automatic Services discovering was proposed, which keeps to the principle of "first riddling, second mapping". The experiment proves that it has higher recall and lower false positives.

Key words: based-requirements integrating; business model; services discovering

0 引 言

Web服务是一种新兴的应用模式和分布计算模型,是Web上业务数据和信息集成的有效机制。目前Web服务研究面临的一个更具挑战性的问题是如何动态地按需集成Web服务,即当需要的功能不能通过已有的服务实现时,已有的服务能被合理地组合来满足这个要求。按需集成Web服务蕴涵两层意思:如何方便、准确地表达用户的需求;如何选择、集成合适的Web服务来满足和解决用户的需求。针对第一个问题,提出了基于场景的业务模型构造方法[1]和场景驱动的工作流模型构造方法[2],在此基础上针对第二个方面,即服务发现方法展开研究。

服务发现是按需集成服务的重要工作,现有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[3,4]。针对上述问题,提出的按需集成服务发现方法包含两个方面:

(1)为服务提供者和服务请求者提供丰富的语义描述,并兼容当前的UDDI注册机制。

(2)依据构建好的业务模型,然后通过添加一些必要的语义信息以及质量属性,即可通过它来进行服务的发现、集成和执行,最终满足和解决用户的需求。

1 基于场景的按需服务集成方法框架

在文献[1]中提出了一种基于场景的按需服务集成方法(Scenario Base Model Construction for service integration,SBMC),其中场景指的是系统执行时的一系列可观察行为序列,使用图示语言UCM来对其进行

收稿日期:2007-09-23
基金项目:国家科学自然基金(60773110);湖南省教育厅优秀青年项目(06B023);湖南省学位与研究生研究课题(06B28);湖南工业大学博士基金
作者简介:李 伟(1983-),女,河南沁阳人,硕士研究生,研究方向为软件体系结构;李长云,博士,教授,硕士生导师,研究方向为软件体系结构、软件自动化。

描述。UCM主要由路径(path)、构件(component)、责任(responsibility)和存根(stub)构成。路径表示场景流,用来连接起始点、责任点和终点;场景构件是责任点的逻辑执行者,可以代表不同的系统实体,如构件、组织和伙伴;责任点表示处理(行为、任务或功能)。存根表示暂时不能确定的行为,起"占位符"的作用。SBMC方法的本质是一种客户驱动的模型设计方法,是客户直接参与和驱动的交互过程,该方法通过对初始场景的演化形成具体的业务模型。

如图1中虚线以上部分所示,首先终端用户将初始的功能性需求和系统行为以UCM图描绘出来,确定好责任点和承担责任点的场景构件,不能确定的行为路线以stub符号表示;同时,质量需求以特征需求表的形式表示。依据特征需求表,用户到模式库中搜索一个或多个合适的设计模式,根据模式使用方法,应用模式到初始场景,得到蕴涵专家解决办法的方案场景,同时评估模式结构对质量属性需求的影响,修改特征需求表中每个质量属性的满足度,然后根据方案场景中stub符号定义的需求目标,搜索插件库中具有与其匹配的操作目标的plug,将这些plug替换相应的stub符,得到情形化场景。接着,将没有充分满足的质量属性需求权衡分配到情形化场景的每一处责任点上,使得责任点不仅表征功能和任务,还附属了质量属性约束,得到带约束场景。依据带约束场景对每一个责任点进行服务选择、匹配和绑定,使得每一个责任点都有一个执行体。

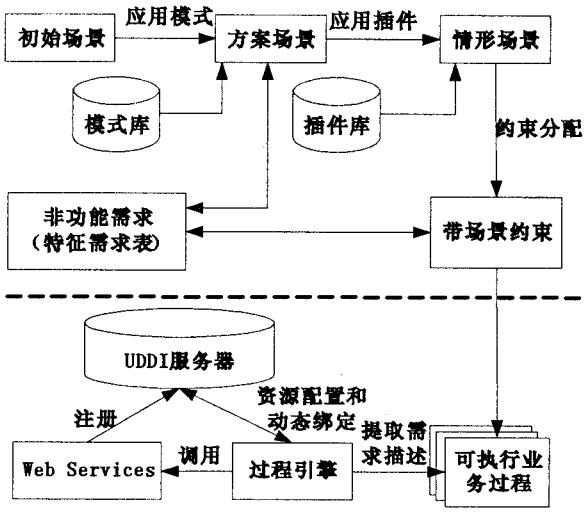

图1　按需集成服务方法框架

至此在构建好的业务模型基础上,按照以下的流程进行服务的发现和绑定:

(1)服务提供者首先向UDDI服务器中发布并注册自己的服务,在UDDI服务器中根据服务语义描述模型,提取出该服务的语义描述,并且根据领域本体对注册的服务分类管理。

(2)通过执行业务模型,产生一系列可执行业务过程,接着依据服务语义描述模型,提取出责任的需求描述,再通过和UDDI服务器中注册的服务按照服务发现匹配方法,自动协商和推理进行资源匹配,找出最符合活动需求描述的服务,然后进行资源的绑定,设置调用的输入参数和输出参数,准备调用具体的服务。

(3)业务流程实例中的责任调用匹配的服务,并根据Grounding文件进行相应的参数传递,完成任务实例,同时按照业务模型的控制流进行服务的集成。

2　服务发现机制

2.1　服务的语义描述

实现服务的自动发现首先要对服务进行语义描述。结合OWL-S描述语义的性质,将语义分为三类:数据语义、功能语义和质量属性语义。

(1)数据语义。

服务发现过程中,对服务操作中的数据使用本体进行注解,可用于输入、输出数据语义与服务请求中的输入、输出数据语义进行匹配,有助于提高系统数据检索和服务互操作的能力[5]。文中则采用领域本体对输入、输出数据进行注解。

(2)功能语义。

文中引入功能语义来细粒度刻画服务。功能语义除了输入语义和输出语义外,还包括服务操作的功能分类、前置条件和后置条件。用s表示服务,用o表示服务的一个操作,那么操作的功能语义可定义为F(s,o)=<N(s,o), Rq-function(s,o), I(s,o), O(s,o), Precondition (s,o), Effect (s,o)>,其中N(s,o)表示服务名称;Rq-function(s,o)表示功能分类;I(s,o)表示输入;O(s,o)表示输出;Precondition (s,o)表示前置条件;Effect (s,o)表示后置条件。

(3)质量属性语义。

服务质量(Quality of Service, QoS)表征了服务定性与定量方面的问题。目前已有一些通用的QoS规格[6,7],而领域具体QoS规格要根据具体领域进行定义。文中采用质量展开法来分配服务的性能(performance)、可靠性(reliability)和扩展性(expansibility),并利用本体来建立QoS规格。该QoS规格由一个二元组表示,即rq-performance=<Nafp, Vafp>。表1的QoS说明服务的可靠性一般、扩展性好而性能较差。

由此,一个服务操作的完整语义定义如下:

OP=<name, rq-function, in, out, precondition, effect, rq-performance>,其中:<name, rq-function, in, out, precondition, effect>是功能语义,告诉服务去

做什么；rq－performance 是 QoS 规格说明，保证服务完成得更好。语义伴随服务的整个生命周期，参与复杂的交互操作，为进行高质量、高效率的服务查询提供保障。

表 1 QoS 语义应用举例

Nafp	Vafp
Reliability	6
expansibility	8
performance	4

2.2 服务提供者和服务请求者的描述模板

根据服务语义描述模型提取出的服务提供者描述模板定义为：

$SA = <SLP(SA), OS(SA, op_1), \cdots, OS(SA, op_m) >$

从业务模型提取出的服务请求者描述模板定义为：

$SR = < SLP(SR), OS(SR, op_1), \cdots, OS(SR, op_m) >$

在上述两个模板定义中，SLP(SA) 和 SLP(SR) 是服务层参数(Service Level Parameter)，由于构件是责任的逻辑执行者，因此文中定义构件为服务层；OS(SA, o_i) 和 OS(SR, o_i) 为操作语义(Operation Semantic)，责任为构件的操作。

而 SLP(SR) = < N(SR), L(SR), D(SR), S(SR) >，其中 N(SR)指服务提供商名称(Name)；L(SR)指服务提供商的地理位置(Location)；D(SR)指服务领域(Domain)；S(SR)指服务提供商的客户满意度。

2.3 服务发现方法

依据“先筛选后匹配”的原则，从服务注册库中选择满足需求的候选服务集，然后通过以下四个步骤选取最佳服务：

1)用本体的领域分类技术筛选与服务请求相关的领域；

2)根据场景构件发现一系列满足条件的粗粒度服务；

3)根据责任的功能语义从满足条件的粗粒度服务中选取一系列细粒度服务；

4)根据质量属性语义，对服务进行匹配，选取最优服务。

根据两个本体概念在本体语法树中的关系，定义两个参数间的匹配度 match(A, B) 如下：

(1) 精确匹配(exact match)，其匹配度值为 3：记做 $A = B$，表示请求与注册的服务参数在语义层次上完全匹配；

(2) 不完全匹配(not exact match)，其匹配度值为 2：记做 $A \geqslant B$，表示请求与注册的服务参数在语义层次上是不完全匹配，因为 A 是 B 的父类；

(3) 勉强匹配(reluctance match)，其匹配度值为 1：记做 $A \leqslant B$，表示请求与注册的服务参数在语义层次上是勉强匹配，因为 A 是 B 的子类；

(4) 完全不匹配(unmatch)，其匹配度值为 0：记做 $A \infty B$，表示请求与注册的服务参数在语义层次上是完全不匹配。

对于服务的自动发现，用 UDDI API 实现其算法，其指导思想是在 SR 和 SA 相似性上进行量化处理，具体步骤如下：

首先是服务层参数的粗粒度匹配，该层次匹配的结果是由服务层参数的各部分(包括服务提供商名称、位置、所属领域和客户满意度) 比较后的值相乘得到的一个 SA 线性序列集。

$$Match(SLP(SR), SLP(SA)) = [result(SLP_i(SR), SLP_i(SA))], SLP_i \in SR$$

$$result(SLP_i(SR), SLP_i(SA)) = \begin{cases} S(SA) & (\text{相等}) \\ 0 & (\text{其他}) \end{cases}$$

然后在这个 SA 序列集中进行功能操作层的匹配，这个层次匹配是细粒度的，结果得到一个范围较小的 SA 线性序列集。

功能操作层参数的匹配结果是由 SR 中的每个操作(具体对应到每个责任) 分别与 SA 中所有功能操作比较的值(用 Matchopf 表示)，这样每一个功能操作都对应有一个从大到小排列的 SA 序列集。

$$Matchopf(OS(SR), OS(SA)) = Matchopf(OS(SR, o_i), OS(SA, o_j)), \cdots, Matchopf(OS(SR, o_i), OS(SA, o_j))$$

其中 $i = 1$ to m, j = number of operation in SA

Matchopf 则是通过操作类、输入、输出、前置条件和后置条件匹配的加权平均值得到。

$$\begin{aligned} Matchopf(OPF(SR, o_i), OPF(SA, o_j)) = {} & [w_1 * match(OP(SR, ope(o_i)), OP(SA, ope(o_j))) \\ & + w_2 * match(OP(SR, out(o_i)), OP(SA, out(o_j))) + \\ & w_3 * match(OP(SR, in(o_i)), OP(SA, in(o_j))) + \\ & w_4 * match(OP(SR, pre(o_i)), OP(SA, pre(o_j))) + \\ & w_5 * match(OP(SR, eff(o_i)), OP(SA, \\ & eff(o_j)))] / (w_1 + w_2 + w_3 + w_4 + w_5) \end{aligned}$$

其中 w_1、w_2、w_3、w_4 和 w_5 权重可以根据 SR 和 SA 中操作类、输入、输出、前置条件和后置条件的优先权设定。

最后，在经过服务层和功能操作匹配后得到一系列较小候选服务序列集中，进行质量属性层次的匹配，选取最优服务。该层次的匹配结果是由 SR 中的每个操作分别与已定序列集里 SA 的所有质量属性操作比较

得到(用 Matchopp 表示),这样每一个质量属性操作都对应于一个从大到小排列的 SA 序列集。

$$\text{Matchopp}(OS(SR),OS(SA)) =$$
$$\text{Matchopp}(OS(SR,o_i),OS(SA,o_j)),\cdots,$$
$$\text{Matchopp}(OS(SR,o_i),OS(SA,o_j))$$

j = number of operation in SA

SR 与 SA 中两个操作的质量属性参数匹配的 Matchopp 结果是由其性能、可靠性和扩展性匹配结果的加权平均值得到。

$$\text{Matchopp}(OP(SR,o_i),OP(SA,o_j)) =$$
$$[w_1 * \text{match}(OP(SR,p(o_i)),OP(SA,p(o_j))) +$$
$$w_2 * \text{match}(OP(SR,r(o_i)),OP(SA,r(o_j))) +$$
$$w_3 * \text{match}(OP(SR,e(o_i)),OP(SA,e(o_j)))]/$$
$$(w_1 + w_2 + w_3)$$

其中 w_1、w_2 和 w_3 权重可以根据 SR 和 SA 中服务对性能、可靠性和扩展性的重视程度设定,得到的候选服务集是按线形评估值排序的,值最高者为最佳服务。

3　服务发现测试

传统的服务发现是通过在 UDDI 库中以关键字匹配的方法进行查找来完成的。为了验证文中提出的基于场景的按需集成服务发现算法的性能,做如下测试:

(1)从 Internet 上下载 30 个 WSDL 文档;

(2)下载 12 个不同领域的本体文件(OWL 格式);

(3)采用相关的领域本体概念注解 WSDL 文件;

(4)用 UDDI API 将注解过的 WSDL 文件发布到 UDDI 库中;

(5)这样每个 Web 服务都有一个与存储在 UDDI 库中相对应的服务描述;

(6)根据服务描述,分别作基于关键字和基于场景的服务发现操作;

(7)重点测试每个服务发现的精确率(Precision)、查全率(Recall)和查错率(False Positives),选取 4 种测试情况:基于具体概念的查找;基于一般概念的查找;查找位置;查找领域。测试结果如图 2,图 3 所示。

由图 2 和图 3 可以看出基于场景服务发现的查全率要明显高于基于关键字的服务查找。并且在查错率上,基于场景查找结果几乎为零(除了第一种情况以外)。

4　结束语

在基于场景的按需集成服务的业务模型构造方法的基础上,提出了按需集成服务自动发现方法。该方法按照"先筛选后匹配"原则进行服务发现,给出的服务匹配算法根据本体定义了 4 个匹配度值,方便定量地发现合适的服务。为了验证本方法,做了服务发现的试验。

测试结果表明该方法在很大程度上提高了服务处理的精准度和效能,为动态服务组合提供了坚实的基础。

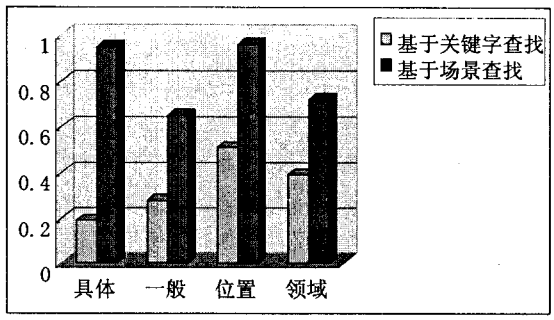

图 2　服务发现的查全率

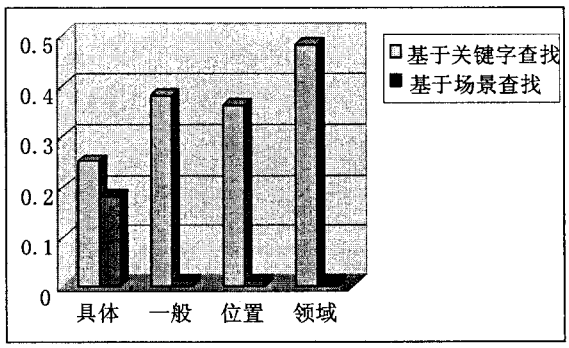

图 3　服务发现的查错率

参考文献:

[1] 李长云,阳爱民,满君丰,等.一种面向按需集成服务的业务模型构造方法[J]. 计算机学报,2006,29(7):1095-1104.

[2] 吴岳忠,李长云,何频捷,等.场景驱动的工作流模型构造方法[J].科学技术与工程,2007,6(7):1007-1011.

[3] 胡建强,邹　鹏,王怀民,等. Web 服务描述语 QWSDL 和服务匹配模型研究[J]. 计算机学报,2005,28(4):505-513.

[4] 许卓明,石　磊.基于语义的 Web 服务发现算法研究[J].计算机应用与软件,2006,23(5):21-23.

[5] McGuinness D L, van Harmelen F. Web Ontology Language (OWL)[EB/OL]. 2002. Web-Ontology(WebOnt) Working Group. http://www.w3.org/2001/sw/WebOnt/.

[6] Cardoso J, Sheth A. Semantic E-Workflow Composition [J]. Journal of Intelligent Information Systems,2003,21(3):191-225.

[7] Cardoso J, Sheth A, Miller J. Quality of Service for Workflows and Web Service Processes[J]. Journal of Web Semantics,2004,24(3):25-32.