

# 一种利用继承关系的业务服务实例化方法

张鹏<sup>1</sup> 韩燕波<sup>2</sup> 王菁<sup>2</sup> 陈旺虎<sup>2</sup> 丁维龙<sup>1</sup> 高晶<sup>1</sup>

(山东科技大学信息科学与工程学院 青岛 266510)<sup>1</sup>

(中国科学院计算技术研究所网格与服务计算研究中心 北京 100190)<sup>2</sup>

**摘要** 随着面向服务计算技术的日渐普及,通过互联网提供的科研领域的 Web 服务也越来越多,目前仅在生物信息领域可统计到的 Web 服务数量就超过 3000 个。然而,怎样在业务层上将最终用户的个性化需求快速实例化到具体的 Web 服务,成为很多专业工作者关心的问题,生物信息领域尤其突出。为了解决这个问题,提出一种利用继承关系的业务服务实例化方法,该方法基于我们前期工作中提出的抽象服务形式——业务服务,通过对基类业务服务的继承操作构建表达最终用户个性化需求的子类业务服务,同时建立了两个业务服务之间的继承关系,并且利用这种继承关系将最终用户的个性化需求快速实例化到具体的 Web 服务。

**关键词** 面向服务计算,业务服务,继承关系,实例化

**中图分类号** TP311 **文献标识码** A

## Instantiation Method of Business Service Utilizing Inherited Relationship

ZHANG Peng<sup>1</sup> HAN Yan-bo<sup>2</sup> WANG Jing<sup>2</sup> CHEN Wang-hu<sup>2</sup> DING Wei-long<sup>1</sup> Gao Jing<sup>1</sup>

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China)<sup>1</sup>

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100910, China)<sup>2</sup>

**Abstract** As the prevalence of Service-oriented Computing, the number of Web services through internet in scientific research field gradually grows more and more. At present, the number of Web service exceeds 3000 in biological field. However, how to map the personalized requirement of end-user into corresponding Web service rapidly on business layer is concerned by many specialists, especially in biologic field. To solve the problem, the paper proposed an instantiation method of business service utilizing inherited relationship. The method is based on an achieved business model——business service, is constructing subclass business service which expresses the personalized requirement of end-user by inherited operations on base class business service, at the same time, inherited relationship is established between the two business services, and it can map the requirement of end-user into corresponding Web service rapidly utilizing the inherited relationship.

**Keywords** SOC (Service Oriented Computing), Business service, Inherited relationship, Instantiation

## 1 引言

Web 服务作为一种新型的分布式计算模型,是 Web 上数据集成的有效机制,但是由于 Web 服务缺少语义信息以及不支持最终用户自主“编程”的构造,最终用户难以直接使用。业务服务<sup>[1]</sup>作为一种具有业务语义的抽象服务形式,它的出现解决了 Web 服务缺少语义信息以及无法在业务层上表达最终用户个性化需求的问题,但是业务服务需要通过 Web 服务实现具体的功能,而实例化的作用就是在业务服务和对应的 Web 服务之间建立关联,使业务服务能够实现具体的功能,其本质问题是业务服务与 Web 服务的匹配。目前的工作中,一方面业务服务实例化的匹配算法没有保证实例化结果的准确性——即任何一种满足最终用户需求的 Web 服务一定在业务服务实例化的结果中,并且任何不满足最终用户需

求的 Web 服务一定不在业务服务实例化的结果中,另一方面业务服务实例化的方法是匹配底层所有 Web 服务,因此,随着底层 Web 服务数量的增多必然会出现大量的冗余匹配,这个问题制约着业务服务实例化的效率,于是,它在保证业务服务实例化结果正确性的基础上减少这种冗余匹配的个数,提高业务服务实例化的效率是合理的也是必要的。

本文的工作主要包括两点,一个是提出了判断 Web 服务是否是业务服务实例的匹配算法,保证了业务服务实例化结果的正确性。另一个是利用继承关系减少了业务服务实例化过程中的冗余匹配,提高业务服务实例化的效率。

文章的组织结构如下,第 2 部分是利用继承关系的业务服务实例化方法的基本原理,第 3 部分是业务服务实例化的相关定义,第 4 部分是业务服务继承关系的性质和继承操作,第 5 部分是业务服务实例化的匹配算法,第 6 部分利用继承

到稿日期:2008-03-19 本文受国家自然科学基金项目(60573117),山东省“泰山学者”专项基金,国家重点基础研究发展计划项目:需求导向的服务资源交互与协同(2007CB310805)资助。

张鹏(1984—),男,硕士研究生,主要研究领域为面向服务的网格计算,E-mail:zhangpeng@software.ict.ac.cn;韩燕波(1962—),男,博士,研究员,主要研究领域为服务网格、软件集成。

关系的业务服务实例化方法的实验分析,第7部分是相关工作。最后是结束语。其中第3部分是整个业务服务实例化过程的基础,它一方面给出了业务服务与Web服务匹配规则,另一方面提供了证明业务服务可以进行的继承操作的依据。

## 2 基本原理

图1描述了业务层结构的变化,通过继承操作构建的业务服务在能力上与之前的业务服务相同,但是它们中形成了一定的逻辑关系。其中每一个长方体都是一个业务服务。根据领域存在内聚性、稳定性的特点,按照领域分析原理,可对某确定领域内相似业务活动根据其共性进行抽象,得到描述业务领域内最小集的业务活动集合,其数量相对固定,反映了领域内最基本的业务活动<sup>[1]</sup>。因此,在业务层的构造上,首先,领域专家通过组合动作和客体来表示领域内最小集的业务活动的概念,在特征层上通过输入、输出、服务质量3个方面的特征建模<sup>[3]</sup>完成对这些最小集的业务活动所表达的领域内相似的业务活动的共性特征的表示。其次,最终用户通过继承操作进行个性化需求的业务服务的构建。这里的特征模型不仅可以表达最终用户的某一具体的需求,而且还可以表达最终用户变化性的需求——即一个特征模型可以表达多种实例,有效提高了业务服务的抽象能力。最终,业务服务通过实例化建立与Web服务的关联,在业务层上完成最终用户的需求。

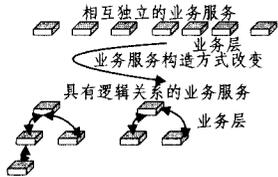


图1 业务层结构的变化

图2表示最终用户从重新构建符合个性化需求的业务服务到通过继承操作构建符合个性化需求的业务服务,业务服务在实例化过程中同样需要与Web服务匹配,然而,由于业务服务的继承操作保证了构建的子类业务服务实例化结果一定包含在基类业务服务的实例化结果中,因此子类业务服务需要与Web服务进行匹配的次数少于重新构建业务服务需要进行匹配的次数,此方法有效复用了基类业务服务实例化的结果,减少了业务服务的冗余匹配,提高了业务服务实例化的效率。

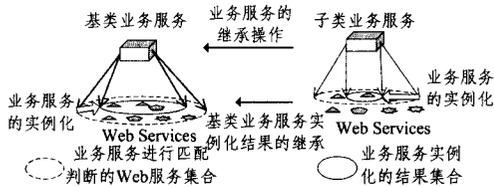


图2 利用继承关系的业务服务实例化方法的原理

## 3 业务服务实例化的相关定义

业务服务实例化的关键步骤是业务服务和Web服务的匹配。由于业务服务的特征模型具有表达变化性需求的能力,所以它的匹配不是一个简单的判断,需要有一个合理的判断规则。这里,我们通过定义给出匹配的规则,并且在第5部

分根据这些规则描述了具体的匹配算法。

### 3.1 特征模型的定义

**定义1(特征模型)** “可选”、“必选”、“或”、“异或”4种特征结点组成的层次结构模型。

**定义2(“可选”特征结点)** 如果“可选”特征结点的父亲特征结点可解,那么“可选”特征结点可以是可解结点。

**定义3(“必选”特征结点)** 如果“必选”特征结点的父亲特征结点可解,那么“必选”特征结点必须是可解。

**定义4(“异或”特征结点)**

1)当“异或”特征结点都是“必选”特征结点时,如果“异或”特征结点的父亲特征结点是可解结点,那么有且仅有一个“异或”特征结点是可解结点。

2)当“异或”特征结点都是“可选”特征结点时,如果“异或”特征结点的父亲特征结点是可解结点,那么最多有一个“异或”特征结点是可解结点。

**定义5(“或”特征结点)** 如果“或”特征结点的父亲特征结点可解,那么“或”特征结点的任何非空集合中的特征结点是可解结点。

**定理1(特征模型表示等价定理)**

两个特征模型表示是等价的,如果从其中一个特征模型表示所有概念实例的集合与从另一个特征模型表示得到的所有概念实例的集合相等。

**推论1** 如果在“异或”特征结点集合中存在可选特征结点等价于“异或”特征结点全是“可选”特征结点。

**推论2** 如果在“或”特征结点集合中存在可选特征结点等价于特征结点全是“可选”特征结点。

### 3.2 业务服务实例的定义

**定义6(业务服务的实例)** Web服务是业务服务的一个实例当且仅当Web服务标注的业务活动概念是业务服务名称概念或其子概念,Web服务的输入、输出、服务质量参数项标注的概念实例集合分别能使业务服务输入特征模型、输出特征模型、服务质量特征模型可解,并且Web服务中所有必选输入参数项标注的概念实例都在输入特征模型可解的概念实例集合中。

**定义7(特征模型可解)** 如果特征模型的根结点可解,则该特征模型可解。

**定义8(特征模型叶子结点可解)** 如果Web服务的某一方面参数项标注的概念实例集合中存在参数项标注的概念实例是这个叶子结点概念或其子概念,则称该叶子结点在Web服务条件下为可解结点。

**定义9(特征模型非叶子结点可解)** 如果Web服务的某一方面参数项标注的概念实例集合在以下两种情况下满足相应的条件,则称该非叶子结点在Web服务条件下为可解结点。

(1)当非叶子结点的子结点中含有必选特征时

该集合为当非叶子结点可解时,根据特征分析得到的所有子特征可解情况下,其中一种情况下子特征对应的概念实例集合。

(2)当非叶子结点的子特征中都是可选特征时

该集合存在参数项标注的概念实例是这个非叶子结点概念。该集合为当非叶子结点可解时,根据特征分析得到的所有子特征可解情况下,其中一种情况下子特征对应的概念实例

例集合。

## 4 业务服务继承关系的性质和继承操作

### 4.1 业务服务继承关系的性质

**性质 1** 子类业务服务实例化的结果一定包含在基类业务服务实例化的结果中。

业务服务继承关系的性质一方面为继承操作的正确性提供了验证基础,另一方面为子类业务服务在实例化过程中减少冗余匹配提供了理论依据。

### 4.2 业务服务的继承操作

业务服务通过展示层的动作和客体组合形成的业务服务名称来表达最终用户需要实现的业务活动概念,通过定制层的特征模型表达最终用户在功能和非功能方面的个性化需求。因此业务服务的继承包括对业务服务动作和客体概念的继承以及特征模型的继承。

#### 4.2.1 业务服务动作和客体概念的继承操作

继承业务服务的动作和客体的概念必须是基类业务服务的动作和客体的概念或其子概念。

**推论 3** 如果继承业务服务的概念不是基类业务服务的概念或其子概念,则继承业务服务与基类业务服务之间不具有继承关系的性质。

**证明:**如果继承业务服务的概念不是基类业务服务的概念或其子概念,根据定义 6,继承业务服务的实例不能满足基类业务服务的实例的定义,则继承业务服务实例化的结果中包含不在基类业务服务实例化的结果中的实例。因此,它们之间不具有继承关系的性质,得证。

#### 4.2.2 特征模型的继承操作

以下是为了保证继承关系的性质而制定的输入、输出、服务质量 3 个方面的特征模型的操作规则。

##### 1) 增加特征

$$(\forall a) Leaf(a) \rightarrow capableAdd(a);$$

$Leaf(x)$  表示  $x$  是特征模型的叶子结点;  $capableAdd(x)$  表示  $x$  可以增加子特征操作。

**推论 4** 如果非叶子结点增加参数特征,则继承业务服务与基类业务服务之间不具有继承关系的性质。

**证明:**如果非叶子结点增加参数特征,根据定义 9,存在使继承业务服务的非叶子结点可解的概念实例不能使基类业务服务的该结点可解,则根据定义 6,继承业务服务的实例化的结果中包含不在基类业务服务的实例化的结果的实例,因此,它们之间不具有继承关系的性质,得证。

##### 2) 删除特征

$$(\forall a) is\_optional(a) \rightarrow capableDelete(a)$$

$is\_optional(x)$  表示  $x$  是可选特征结点;  $capableDelete(x)$  表示  $x$  可以删除。

$$(\forall a)(\forall b) Leaf(a) \cap is\_mandatory(a) \cap is\_parent(a, b) \cap (is\_alternative(a) \cup is\_or(a)) \cap (number(b, subNodeSet) > 1) \rightarrow capableDelete(a)$$

其中  $Leaf(x)$  表示  $x$  为特征模型的叶子结点;  $is\_mandatory(x)$  表示  $x$  是必选特征结点;  $is\_parent(x, y)$  表示  $y$  是  $x$  的父结点;  $is\_alternative(x)$  表示  $x$  是“异或”特征结点;  $is\_or(x)$  表示  $x$  是“或”特征结点;  $number(x)$  表示  $x$  的个数;  $capableDelete(x)$  表示  $x$  可以删除。

**推论 5** 如果子特征为“异或”或“或”特征,且子特征中存在必选特征,那么子特征都是必选特征(应用特征模型等价定理)。

**推论 6** 如果不是“异或”或“或”的必选特征结点被删除,则继承业务服务与基类业务服务之间不具有继承关系的性质。

**证明:**如果不是“异或”或“或”的必选特征结点被删除,根据定义 8 或定义 9,则存在使继承业务服务中删除了该结点的父结点可解的概念实例不能使基类业务服务中相应的父结点可解。根据定义 6,继承业务服务的实例化的结果中包含不在基类业务服务的实例化的结果的实例,因此,它们之间不具有继承关系的性质,得证。

**推论 7** 如果当只有一个是“异或”或“或”的必选结点时,删除该结点,则继承业务服务与基类业务服务之间不具有继承关系的性质。

**证明:**如果当只有一个是“异或”或“或”的必选结点时,删除该结点,根据定义 8 或定义 9,则存在使继承业务服务中删除了该结点的父结点可解的概念实例不能使基类业务服务中相应的父结点可解。根据定义 6,继承业务服务的实例化的结果中包含不在基类业务服务的实例化的结果的实例,因此,它们之间不具有继承关系的性质,得证。

#### 3) 改变特征类型

$$(\forall a) is\_optional(a) \rightarrow capable\_mandatory(a)$$

$is\_optional(x)$  表示  $x$  是可选特征结点;  $capable\_mandatory(x)$  表示子特征结点  $x$ ; 可以改变为“必选”特征结点。

**推论 8** 如果改变“必选”特征结点为“可选”特征结点,则继承业务服务与基类业务服务之间不具有继承关系的性质。

**证明:**如果改变“必选”特征结点为“可选”特征结点,根据定义 8 或定义 9,则存在使继承业务服务的该结点的父结点可解的概念实例不能使基类业务服务的该结点的父结点可解。根据定义 6,继承业务服务的实例化的结果中包含不在基类业务服务的实例化的结果的实例,因此,它们之间不具有继承关系的性质,得证。

**推论 9** 如果改变“异或”或“或”特征结点,则继承业务服务与基类业务服务之间不具有继承关系的性质。

**证明:**如果改变“异或”或“或”特征结点,根据定义 8 或定义 9,则存在使继承业务服务的该结点的父结点可解的概念实例不能使基类业务服务的该结点的父结点可解。根据定义 6,继承业务服务的实例化的结果中包含不在基类业务服务的实例化的结果的实例,因此,它们之间不具有继承关系的性质,得证。

### 4.2.3 业务服务继承操作的正确性验证

通过实例分析来验证上面的继承操作可以保证业务服务之间继承关系的性质,如图 3 所示。

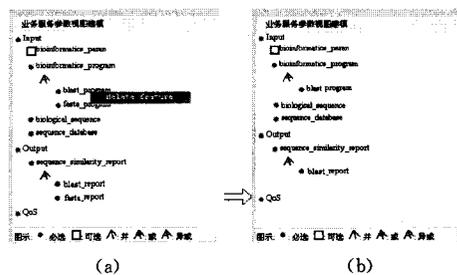


图 3 继承操作保证业务服务继承关系性质的实例验证

范例：“序列相似性查找”业务服务及继承于它的“Blast 序列相似性查找”业务服务。

图 3(a)是序列相似性查找业务服务,图 3(b)是通过继承操作得到的 Blast 序列相似性查找业务服务。

分析:图 3(a)图表达的概念的实例结果包括 8 种类型,分别是 {A, C, E, F, H}, {A, D, E, F, H}, {A, C, E, F, I}, {A, D, E, F, I}, {C, E, F, H}, {D, E, F, H}, {C, E, F, I}, {D, E, F, I};图 3(b)表达的概念的实例结果包括 2 种类型,分别是 {A, C, E, F, H}, {C, E, F, H}。其中 A 表示 bioinformatics\_param 概念,B 表示 bioinformatics\_program 概念,C 表示 blast\_program 概念,D 表示 fasta\_program 概念,E 表示 biological\_sequence 概念,F 表示 sequence\_database 概念,G 表示 sequence\_similarity\_report 概念,H 表示 Blast\_report 概念,I 表示 Fasta\_report 概念。

结论:继承的业务服务实例化的结果全部包含在基类业务服务实例化的结果中,保证了业务服务继承关系的性质。

## 5 业务服务实例化的匹配算法

第 3 部分的定义给出了算法的判断规则,其中定义 6 给出了判断一个标注语义的 Web 服务是否是业务服务的实例依据。算法描述如下:

Algorithm:

输入:业务服务(BS),语义标注的 Web 服务(WS)

输出:业务服务的实例集合(BS, InstanceSet)

过程:

```
If (Web 服务标注的业务活动概念! = 业务服务名称概念
    && Web 服务标注的业务活动概念 ∈ 业务服务名称概念的子概念集合)
```

```
break;
```

```
For each FeatureModel ∈ {业务服务的输入,输出,服务质量特征模型} do
```

```
int j = 特征模型的深度;
```

```
Set ConceptInstances = Web 服务相应标注的概念实例集合;
```

```
int i = j;
```

```
For each i > 0 do
```

```
If(i = j)
```

```
{judgeLeafFeatureNodes(特征模型第 i 层特征结点集合, ConceptInstances);应用定义 8
```

```
i--;}
else
```

```
{judgeNonLeafFeatureNodes(特征模型第 i 层特征结点集合, ConceptInstances);应用定义 9
```

```
i--;}
end
```

```
end
```

```
Node root = FeatureModel.getRoot();
```

```
If(! root.getResolve())
```

```
break;
```

```
end
```

```
if(FeatureModel = 输入特征模型)
```

```
if(Web 服务必选输入参数项标注的概念实例 ∈ 输入特征模型可解的概念实例集合)
```

```
break;
```

```
BS, InstanceSet.add(WS);
```

```
return BS, InstanceSet;
```

```
End Algorithm
```

业务服务实例化的匹配算法保证了业务服务实例化结果的正确性。即任何一种满足最终用户需求的 Web 服务一定在业务服务实例化的结果中,并且任何不满足最终用户需求的 Web 服务一定不在业务服务实例化的结果中。这种正确性为子类业务服务实例化过程中减少冗余匹配提供了前提。同时,也保证了实现业务服务的具体功能。

## 6 实验分析

应用实例:业务人员需要使用 Fasta 的序列相似性查找 Web 服务。目前服务社区注册的 33 个 Web 服务中有 2 个是实现此功能的 Web 服务。

实验目标:本实验通过比较在人工判断 Web 服务、普通方式的业务服务实例化、利用继承关系的业务服务实例化 3 种策略下实现最终用户的个性化需求的平均时间来验证利用继承关系的业务服务实例化的效果。

实验数据:实验数据包括 DDBJ, EBI, MyGrid, 本地程序等多种形式的生物领域的 Web 服务。

实验步骤:

a)注册实验所用 Web 服务并利用领域本体进行语义标注;

b)按普通方式构造业务人员个性化需求的 Fasta 序列相似性查找业务服务;

c)通过对序列相似性查找业务服务的继承操作来构建业务人员个性化需求的 Fasta 序列相似性查找业务服务;

d)分别进行业务服务的实例化过程。

注:为观察评价指标随 Web 服务注册个数增加的规律,实验记录注册 Web 服务个数分别为 10, 20, 30 时各种策略下实现用户需求的平均时间(方法 1 的实验数据是根据本人进行人工判断需要的时间得到的,方法 2 和方法 3 的实验数据是通过程序得到的)。

实验结果(如图 4 所示):

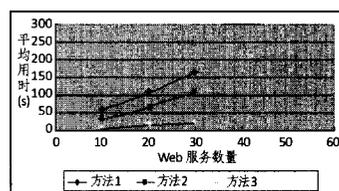


图 4 3 种方法的平均用时的实验结果

方法 1:人工判断 Web 服务;方法 2:普通方式的业务服务实例化;方法 3:利用继承关系的业务服务实例化。

实验结果分析(如图 5 所示):我们分析一下当 Web 服务数量为 30, 3 种方法实现用户个性化需求的平均时间。

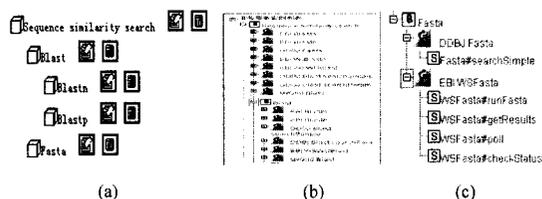


图 5 序列相似性查找业务服务继承关系及实例化结果

图 5(a)是显示通过继承操作建立的业务服务的继承关系,图 5(c)是实现序列相似性查找功能的 8 个实例化的 Web

服务,图 5(b)是实现 Fasta 序列查找功能的 2 个 Web 服务,其中 EBISWFasta 服务的功能是由 4 个操作交互而形成,具体实现可参考文献[15]。

在策略 1 下,最终用户通过判断每一个 Web 服务的注册信息去人工筛选出满足其个性化需求的具体应用,这样,最终用户平均判断次数为  $\{\sum_{N=1}^{30} N/30\}/2 = 7.75$ (次),判断一个 Web 服务是否满足其个性化需求的平均时间  $\approx 20s$ ,实例化平均用时 = 平均判断次数 \* 平均判断一次的时间  $\approx 155s$ ,并且最终用户需要每次去判断。在策略 2 下,最终用户通过构建 Fasta 序列相似性查找业务服务来表达其个性化的需求,并且通过业务服务实例化的匹配算法去判断每个 Web 服务是否满足需求,由于业务服务实例化的正确性,因此,虽然业务服务实例化过程需要进行 30 次匹配判断,但是,一方面由于机器的正确性和高效性,在时间上会少于人工判断的时间,平均一次的匹配判断时间  $\approx 3.5s$ ,实例化平均用时  $\approx 105s$ 。另一方面,最终用户不需要每次进行重新的匹配,可以利用已有的业务服务实例化结果。因此,在长期效果方面,策略 2 好于策略 1。在策略 3 下,业务人员通过对序列相似性查找业务服务进行继承操作构建表达其个性化需求的 Fasta 序列相似性查找业务服务,由于我们验证了继承操作保证业务服务之间继承关系的性质,因此,Fasta 序列相似性查找业务服务实例化过程只需要进行 8 次匹配判断,实例化平均用时  $\approx 28s$ 。并且最终用户不需要每次进行重新的匹配判断,可以利用已有的业务服务实例化结果。

结论:实验结果显示策略 3 无论在实例化时间上还是在结果的重用上都好于策略 1 和策略 2,并且随着 Web 服务数量的增多对于策略 1 和策略 2 平均判断次数会线性增长,对于策略 3 平均判断次数为基类业务服务实例化结果的个数,由于特征模型具有表达变化性的特点,因此,可以有效降低平均判断次数增长的幅度,减少业务服务实例化的平均用时,提高业务服务实例化的效率。

## 7 相关工作

传统的业务建模的方法,通常将重点放在对业务模型所涉及的实体、过程、目标和功能进行充分和全面的表达上。近年来,有些研究使用 UML 并采用 4+1 对业务模型进行描述,有些也以业务规则形式体现模型的灵活和重构,但它们缺少足够的细节,没有形成相对清晰和完整的体系,而作为服务集成的业务模型不仅要求足够的业务细节,而且要是个集成化、可执行的模型。在这些方面业务服务不但表现出具有的表达变化性的高度集成化的性质,而且还可以实现具体的功能。在领域工程中有些工作例如文献[4]虽然引入了特征建模的理论,但所用的业务模型主要在描述体系结构的共性和变化性上,即使有些应用在 Web 服务的描述上,但也没考虑怎样实例化到相应的具体应用上,缺少一种适应特征模型的匹配算法。目前,有人也提出按需即时构造业务模型的观点例如文献[5],但他们关心的是最终用户提出需求后构造的业务模型,而我们关注的不仅是即时构造业务模型更注重业务服务的实例化。

在服务发现和服务匹配上,文献[6]扩展了 UDDI 使其能够基于服务的语义信息提供检索功能。文献[7,8]分别提出了 Service Domain 和 Service Container 的概念,探讨了具有相

同功能服务的组织和管理。我们的工作与他们的区别是我们的业务服务面向的是最终用户,因此在功能上体现出对最终用户的个性化支持。文献[9]中作者提出了基于扩展 DL 的 PI-Calculus 服务匹配方法,在文献[10]中,作者提出了基于抽象的有限状态自动机的方法来比较两个服务的能力,他们都只关心过程的匹配,很难适应服务请求的个性化及 Web 服务的多样性的情况。而我们的匹配算法,是基于业务服务的,它不仅适应 Web 服务的多样性,而且也适应服务请求的个性化。

我们的前期工作,虽然提出了一种表达最终用户个性化需求的业务模型——业务服务,以及业务服务的复用概念,但是没有深入研究实现复用的方式,在业务层上表现的仍然是一个个相互独立的模型,使得在模型的实例化过程中,平均需要的时间随 Web 服务数量的增加保持  $O(n)$  的线形增长,最终导致业务服务实例化的效率出现瓶颈。

本文针对领域内服务资源及最终用户使用特点,通过继承操作不仅构造了表达最终用户个性化需求的业务服务,而且也提高了业务服务实例化的效率。

**结束语** 业务服务作为一种抽象服务形式,让最终用户更加容易理解和使用。通过业务服务的继承操作,业务服务在表达最终用户个性化需求的同时,减少了实例化的时间,不仅如此,最终用户可以通过业务服务实现具体的功能,真正实现了屏蔽底层 Web 服务的多样性和异构性,相关工作参考文献[2]。关于业务服务的继承的研究,我们还有很多工作要做,比如业务服务是否可以多继承等等,这些问题将是我们以后工作的重点。

## 参考文献

- [1] 王建武. 面向领域及业务用户的服务模型研究. 博士学位论文. 北京:中科院计算所,2007
- [2] Wang Jing, Han Yanbo, et al. VINCA4 Science: A Personal Workflow System for e-Science ICICSE. 哈尔滨工程大学,2008,1: 28-29
- [3] Dipl.-Inf. Krzysztof Czarnecki, Generative Programming. October 1998
- [4] 彭鑫,赵文耘,刘奕明. 基于特征模型和构件语义的概念体系结构设计. Journal of Software,2006,17(6):1307-1317
- [5] 李云长,阳爱民,满君丰,等. 一种面向按需继承服务的业务模型. 计算机学报,2006(7)
- [6] Paolucci M, Kawamura T, Payne T, et al. Semantic Matching of Web Services Capabilities // Proceedings of the International Semantic Web Conference, Sardinia, Italy, June 2002:333-347
- [7] Khalaf R, Leymann F. On Web Services Aggregation // Proceedings of the 4th International Workshops on Technologies for E-services in Conjunction with the VLDB conference. LNCS 2819. Berlin; Germany,2003:1-13
- [8] Benatallah B, Dumas M, Maamar Z. Definition and Execution of Composite Web Services; The SELF-SERV Project. Data Engineering Bulletin, IEEE Computer Society,2002,25(4):47-52
- [9] Agarwal S, Aankolekar A. Automatic matchmaking of Web services[A] // Proceedings of the 15th International Conference on World Wide Web (WWW 2006) [C]. New York: ACM Press, 2006:1057-1058
- [10] Gao Xiang, Yang Jian, Papazoglou M P. The Capability Matching of Web Services[A] // IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02) [C]. Los Alamitos: IEEE Computer Society,2002:56-63