

基于活动关联度的服务构件识别方法

吴 雷^{1,2}, 魏 臻¹, 岳 峰¹, 邱述威¹

(1. 合肥工业大学计算机与信息学院, 合肥 230009; 2. 安徽教育网络出版有限公司, 合肥 230061)

摘 要: 采用自顶而下的分析方法, 结合面向服务架构(SOA)下企业业务流程模型组成元素的特点, 提出一种基于活动关联度的服务构件识别方法。利用业务流程模型元素间的关联关系, 通过业务活动之间关联度计算公式, 形成关联度矩阵, 采用基于矩阵变换的层次聚类分析方法对矩阵进行处理, 获取高内聚、低耦合、可复用的业务构件。实例分析结果表明, 与基于遗传算法的服务识别方法和基于聚类分析的企业信息系统业务构件方法相比, 该方法识别出的服务构件综合度量指数最优。

关键词: 服务构件; 构件识别; 业务模型; 活动关联度; 聚类分析; 关联度矩阵

Service Component Identification Method Based on Activity Correlation

WU Lei^{1,2}, WEI Zhen¹, YUE Feng¹, QIU Shu-wei¹

(1. School of Computer & Information, Hefei University of Technology, Hefei 230009, China;

2. Anhui Education Network Publishing Co., Ltd., Hefei 230061, China)

【Abstract】 This paper uses the top-down analysis method and considers the characteristics of the elements in enterprise Service-oriented Architecture(SOA) model, it proposes a method to identify service components based on business model. It can generate relevance matrix by activity relevance formula based on the association between elements in the business process model. Then analyzing the matrix using the cluster algorithm, it gets the high-cohesion, low-coupling and reusing business components. Example analysis results show that this method can identify service components with optimizer comprehensive measuring index than service identification method based genetic algorithm and enterprise business component method based clustering analysis.

【Key words】 service component; component identification; business model; activity correlation; clustering analysis; correlation matrix

DOI: 10.3969/j.issn.1000-3428.2013.04.016

1 概述

随着软件开发技术的不断发展, 面向服务架构(Service-oriented Architecture, SOA)已逐渐广泛应用于企业信息系统中, 成为继面向过程、面向对象、面向构件等技术后的主流软件开发架构, 能较好地实现应用系统的业务敏捷性, 保持信息系统的灵活性。

如何在 SOA 框架下从企业的业务模型中获取高内聚低耦合、具有一定可复用性的服务构件, 构建面向服务的系统架构模型, 以满足企业应用灵活开发的需要, 成为该领

域国内外学者研究的热点问题。

业务构件识别的相关因素包括识别目标、输入模型形态、识别策略、识别过程等, 这些因素的差异产生了不同的识别方法^[1]。基于企业业务模型的构件识别大多以聚类分析算法为基础, 目前的研究主要分为基于业务元素关联度的聚类分析^[2-3]和基于创建-读取-写入-删除矩阵的聚类分析方法^[4-5]2 种。

前者通过分析构成业务模型的业务元素之间的联系, 计算出业务元素之间的关联度, 将业务模型转化为数学模型, 如矩阵形式、加权有向图的形式, 然后采用聚类分析

基金项目: 国家自然科学基金资助项目(61070220); 国家“863”计划基金资助项目(2011AA060406); 高等学校博士学科点专项科研基金资助项目(20090111110002)

作者简介: 吴 雷(1981—), 男, 博士研究生, 主研方向: 企业信息系统, 模型驱动开发; 魏 臻, 教授、博士生导师; 岳 峰、邱述威, 博士研究生

收稿日期: 2012-04-12 **修回日期:** 2012-07-14 **E-mail:** youhuagu@163.com

技术识别出大粒度的构件; 后者通过分析业务活动和业务实体之间的 4 种关系——创建、读取、写入和删除, 将具有创建或删除关系的业务活动与业务实体合并到一个业务构件中。

目前, SOA 下的构件识别问题主要体现在对企业业务模型缺乏形式化的描述, 在识别过程中通常需要依靠领域专家的经验, 以致无法从企业业务模型中自动识别出企业级的业务构件。

本文采用自顶而下的分析方法, 结合 SOA 架构下企业业务模型组成元素的特点, 首先, 对 SOA 下的企业业务元模型进行分析, 分别从不同特征侧面给出了活动关联度的计算公式, 克服了依靠领域专家经验设定的局限性; 其次, 为了识别可复用的服务构件, 提出一种基于矩阵变换的聚类分析算法, 并且给出对服务构件进行度量的公式。

2 服务构件的总体识别过程

企业业务模型是对组成系统的业务元素以及元素之间关系的抽象化描述, 文献[6-7]对 SOA 架构下企业业务建模的元模型做了描述, 元模型构成元素包括业务流程、业务活动、操作以及执行角色等, 这些元素之间的关系见图 1。从图 1 可以看出构成服务的核心元素组活动及其之间的关系。因此, 从业务模型中识别出高内聚、低耦合的服务构件的过程, 其主要工作是对其中的活动进行划分, 将具有较高关联度的组活动划分到一个构件中。

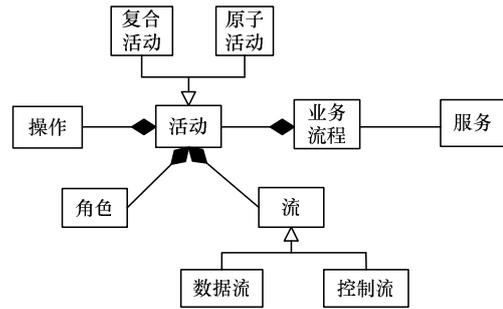


图 1 SOA 下的业务建模元模型

说明: (1)业务流程与服务之间属于双向关联, 双向关联在 UML 类图中可以用无方向箭头(如图 1 所示)或双向箭头表示。(2)复合活动与原子活动同活动的关系属于泛化, 也就是继承关系, 因此, 用空心的箭头。数据流与控制流同流的关系同理。

基于上述分析, 本文提出一种基于活动关联度的服务构件识别方法, 识别过程如图 2 所示。

该方法以 UML 语言描述的 SOA 下的业务模型作为输入, 分别从不同侧面提出活动的关联度信息, 形成关联度矩阵, 将此模型作为服务构件识别的输入模型; 然后, 采用基于矩阵的转换方法对矩阵作相应变化, 将关联度较强(通常为大于一个阈值)的一组活动形成一个服务构件; 最后, 通过本文所提的度量计算公式, 从构件粒度、内聚性和耦合性 3 个方面对服务构件进行评估, 从而得到服务构件模型的总体度量指数。

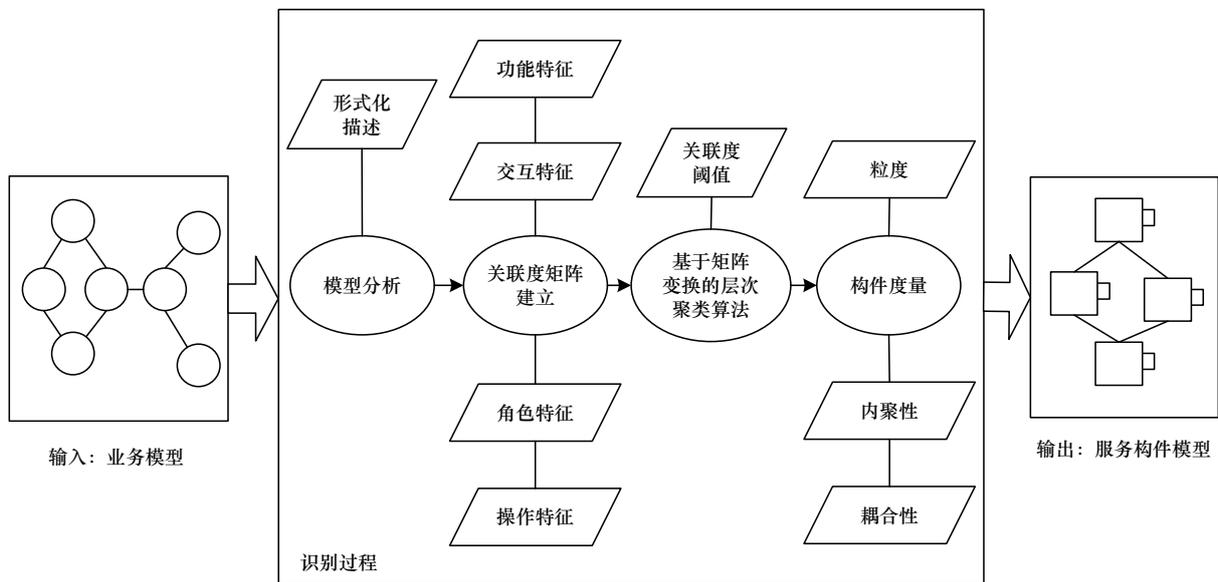


图 2 服务构件的识别过程

3 基于活动的关联度矩阵构建

服务构件是对构成该构件的功能实体所包含的业务活动的封装。通过图 1 所示的元模型可以看出, 影响 2 个活动之间关联情况的因素主要包括: 2 个活动是否属于同一业务流程; 2 个活动之间是否有数据关联; 2 个活动共同的执

行角色情况以及 2 个活动所包含的公共操作情况等方面, 本文将这 4 种因素归纳为功能特征、交互特征、角色特征和操作特征。

(1)功能特征。业务层面的任何活动都隶属于某一业务流程中, 而业务流程是由一组完成特定功能目标的业务活

动构成。因此,业务活动都具有功能特征,属于同一业务流程的2个活动之间的相关性要高于属于2个不同业务流程的活动。功能关联度计算公式可以定义为:

$$Rel_1(a_i, a_j) = \frac{Count(Process(a_i) \cap Process(a_j))}{Count(Process(a_i) \cup Process(a_j))} \quad (1)$$

其中, $Count(Process(a_i) \cap Process(a_j))$ 表示活动 a_i 和 a_j 共同属于的业务流程数目; $Count(Process(a_i) \cup Process(a_j))$ 表示活动 a_i 和 a_j 所有的附属业务流程数目。

(2)交互特征。活动之间的交互特征反映了活动间另一层面的关联性。如果2个活动之间存在交互关系,那么这2个活动之间的相关性要高于不存在交互关系的活动。交互关联度计算公式可以定义为:

$$Rel_2(a_i, a_j) = \frac{isInteractive(a_i, a_j)}{Count(Interact(a_i) \cup Interact(a_j))} \quad (2)$$

其中, $isInteractive(a_i, a_j)$ 函数表示活动 a_i 与 a_j 之间是否存在交互关系,当存在交互关系时值为1,当不存在时值为0; $Count(Interact(a_i) \cup Interact(a_j))$ 表示活动 a_i 和 a_j 对外交互的所有活动数目。

(3)角色特征。角色特征反映执行活动的人员角色情况。活动与执行角色之间是多对多的关系,每个业务活动都对应多个角色,因此,可以通过活动的数学模型计算出角色相关性:

$$Rel_3(a_i, a_j) = \frac{Count(Actors(a_i) \cap Actors(a_j))}{Count(Actors(a_i) \cup Actors(a_j))} \quad (3)$$

其中, $Count(Actors(a_i) \cap Actors(a_j))$ 表示活动 a_i 和 a_j 共同的执行角色数目; $Count(Actors(a_i) \cup Actors(a_j))$ 表示活动 a_i 和 a_j 所有的执行角色数目。

(4)操作特征。操作特征包括该活动所执行的操作集合情况。2个不同的业务活动包含的相同操作比例越高,两者之间的相关性就越高。因此,可以通过服务的数学模型计算操作相关性:

$$Rel_4(a_i, a_j) = \frac{Count(Operations(a_i) \cap Operations(a_j))}{Count(Operations(a_i) \cup Operations(a_j))} \quad (4)$$

其中, $Count(Operations(a_i) \cap Operations(a_j))$ 表示活动 a_i 和 a_j 共同操作数目; $Count(Operations(a_i) \cup Operations(a_j))$ 表示活动 a_i 和 a_j 所有的操作数目。

通过式(1)~式(4)分别对以上4类特征相关度进行计算,将其结果代入式(5),可以得出任意2个服务间的关联度:

$$Rel(a_i, a_j) = \sum_{k=1}^4 w_k \times Rel_k(a_i, a_j) \quad (5)$$

其中, w_k 为该特征的关联度权值系数。

依据上文给出的活动间的关联度计算公式,可以计算出任意2个活动之间的关联度,从而建立关联度矩阵。由

于活动关联的双向性,因此该矩阵为上三角矩阵。例如, n 个活动构成的关联矩阵 $A_{n \times n}$ 可以表示为:

$$A_{n \times n} = \begin{bmatrix} Rel(a_1, a_1) & Rel(a_1, a_2) & \cdots & Rel(a_1, a_n) \\ Rel(a_2, a_1) & Rel(a_2, a_2) & \cdots & Rel(a_2, a_n) \\ \vdots & \vdots & & \vdots \\ Rel(a_n, a_1) & Rel(a_n, a_2) & \cdots & Rel(a_n, a_n) \end{bmatrix}$$

4 基于矩阵变换的服务构件识别算法

聚类分析将具有高度聚合性的元素聚起来形成某种模式,聚类结果依赖于样本点集合的选取以及对样本点之间关联值的理解,被广泛应用在数据挖掘、模式识别等领域^[1]。

K-means 算法聚类方法中有一种典型的基于距离的算法,但是该算法在应用时,需预先确定聚类数 k ,在进行构件识别时,能够识别出的构件数目是无法预先设置的。本文在 K-means 方法的基础上,提出一种基于矩阵变换的聚类算法,通过对矩阵行列做相应初等变换,把矩阵分块成子矩阵聚类。

4.1 聚类识别算法

该算法首先设定一个关联度阈值 θ 。然后从矩阵的首行依次对关联矩阵进行按列遍历,将权值大于等于 θ 的服务所在行与列同该行首个权值小于 θ 的服务所在行与列作行列的初等交换,以至于多次交换后,前移的行、列能够构成一个所有元素都大于 θ 的上三角矩阵。在这样的上三角矩阵内,其任意服务间的关联度都大于等于设定的关联度阈值,因此,可以构成一个候选构件集合。具体算法流程如下:

算法1 $shift(A, i, \theta)$

作用:对矩阵 A 中的第 i 行的各列依次遍历,将关联度大于等于阈值 θ 的活动所在行及列同当前列中首个小于阈值 θ 的活动所在行与列作对调变换;

已知:关联度矩阵 $A_{n \times n}$, 需要遍历的行数 i , 关联度阈值 θ ;

结果:经过一次行遍历变换后的矩阵 $B_{n \times n}$ 。

算法描述:

For $j=1:n$

IF 找到小于 θ 的列

记录此列所在的列数 lesscol 及对应的活动数;

Else IF 找到大于等于 θ 的列

记录此值所在的列数 greatcol 及对应的活动数;

IF 此列前存在小于 θ 的列

交换矩阵 A 中第 lesscol 行与第 greatcol 行的值;

交换矩阵 A 中第 lesscol 列与第 greatcol 列的值;

End IF

End IF

End

算法2 $getCompSET(A, \theta)$

作用:获取服务构件的集合

已知: 关联度矩阵 $A_{n \times n}$, 关联度阈值 θ ;

结果: 对 n 个服务重新划分, $SET = \{SET_1, SET_2, \dots, SET_m\}$,

$0 < m \leq n$ 。

算法描述:

初始化: $SET = \Phi$, $TempSET = \Phi$; $k, flag = 0$;

For $i = 1: n$

 执行算法 $shift(A, i, \theta)$, 输出变换后的矩阵 A ;

 IF 矩阵 A 中第 $1+flag$ 行、第 $1+flag$ 列到第 i 行、第 i 列所构成的上三角子矩阵中, 所有的元素都大于或等于已知的关联度阈值 θ

 将该行所在的活动插入到临时集合 $TempSET$ 中;

 继续向后遍历 $k++$;

 Else

 将临时集合 $TempSET$ 作为一个新的子集合插入到 SET 中, 形成一个构件集合;

 将此时的 k 值赋给 $flag$;

 将临时集合 $TempSET$ 置空;

 将此时代行所在的活动插入到临时集合 $TempSET$ 中;

 End If

End

4.2 时间复杂度分析

对于一个矩阵 $A_{n \times n}$, 下面分析用上述算法对该矩阵进行处理时的时间复杂度。当执行算法 1 对任意一行的所有列进行一次迭代时, 对矩阵中大于等于阈值行、列同符合条件的小于阈值的行列作对换时, 在最坏情况下, 时间复杂度为 $O(n^2)$; 当执行算法 2 时, 对某一行进行是否小于阈值的判断, 最坏的情况为 $O(n)$ 。因此, 算法的总体时间复杂度最大为 $O(n^3)$ 。实际上, 由于本文生成的关联度矩阵是对称矩阵, 这样的矩阵作行、列对换时的时间复杂度会降低。因此 $O(n^3)$ 只是时间复杂性的一个上界, 目前的计算机处理能力完全可以处理这类识别问题。

5 服务构件的性能度量方法

文献[2-3]提出对业务构件的粒度进行度量的方法, 能够用于指导业务构件的优化。但构件识别目标一般要求粒度良好, 在比较识别结果时, 粒度这个因素无法用大或者小的量化指标来判断构件的性能, 也就是说, 粒度的大小不能正确地反映构件识别的效果。文献[8]提出从粒度、耦合性、内聚性等多个方面度量服务的方法, 本文通过分析图 1 所示面向服务架构下的企业建模元模型, 对文献[8]中的度量方法作了改进, 形成了从 3 个方面对服务构件进行度量的计算公式。

(1) 构件粒度: 服务构件的粒度描述了构成该构件的所有活动的功能属性, 所以, 粒度的计算公式同构件所包含活动的操作集合相关。服务构件粒度可以定义为:

$$Granu(SM) = \frac{1}{|SM|} \sum_{i=1}^{|SM|} \frac{SumOperation(SC_i)}{|SC_i|} \quad (6)$$

其中, $SumOperation(SC_i)$ 表示服务构件 SC_i 中所包含活动的操作数目; $|SM|$ 表示模型中所有服务构件的数目; $|SC_i|$ 表示服务构件 SC_i 中包含的活动数目。在一般情况下, $SumOperation(SC_i) > 0, 0 < |SM| \leq SumOperation(SC_i)$ 。

(2) 耦合性: 服务构件的耦合性说明了该构件同其他构件之间的关联程度, 反映了 2 个构件之间的依赖属性。构件中的活动通过数据流同其他构件内的活动交换信息。所以, 服务构件的总体耦合性可以定义为:

$$Coupl(SM) = \frac{1}{|SM|} \sqrt{\sum_{i=1}^{|SM|} \frac{OutOperations(SC_i)^2}{|SC_i|^2}} \quad (7)$$

其中, $OutOperations(SC_i)$ 表示服务构件 SC_i 中关联外部构件的活动数目。

(3) 内聚性: 服务构件的内聚性反映了构件内部所有活动之间的关联属性, 构件内部活动之间的关联值越高, 构件的内聚性越强。所以, 服务构件的总体内聚性可定义为:

$$Cohes(SM) = \frac{1}{|SM|} \times \sqrt{\sum_{i=1}^{|SM|} \frac{(Weight_{DataFlow}(SC_i) + Weight_{ControlFlow}(SC_i))^2}{|SC_i|^2}} \quad (8)$$

其中, $Weight_{DataFlow}(SC_i)$ 和 $Weight_{ControlFlow}(SC_i)$ 分别表示服务构件 SC_i 在本构件内的数据关联操作数目和控制关联操作数目。在实际应用系统开发过程中, 上述 3 个度量因素通常是互斥的。因此, 在衡量服务构件的综合性能指数时, 为每个因素引入权值, 通过权值描述因素的重要性。

6 实例分析

下面以某出版集团 ERP 系统中的出版选题管理子系统项目为背景^[9], 将出版社“图书选题管理”的业务流程模型作为实例对象, 验证本文方法的可行性。通过分析该业务流程模型所包含的业务活动及其关联的数据、操作、角色、业务流程等元素, 通过式(1)~式(5)计算出该模型所包含的任意 2 个活动之间的关联度, 建立的关联度矩阵如图 3 所示, 关联度阈值 θ 为 0.7。

	s_1	s_2	s_3	s_4	s_5	s_6
s_1	1	0.7	0.5	0.9	0.6	0.5
s_2		1	0.4	0.8	0.6	0.1
s_3			1	0.1	0.6	0.8
s_4				1	0.3	0.2
s_5					1	0.5
s_6						1

图 3 业务活动关联度矩阵

采用第 4 节介绍的基于矩阵变换的服务构件识别算法

对该关联度矩阵进行变换,其变换过程如图4所示。

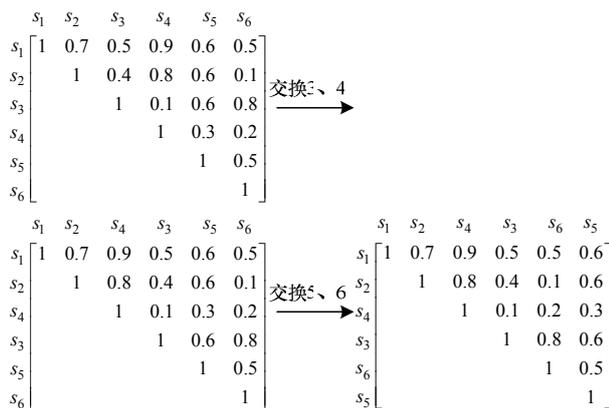


图4 活动关联度矩阵的移动变换过程

经过聚类识别算法处理,可以得到经过移动变换过的关联度矩阵。因此,可以得出候选构件集合为 $\{a_1, a_4, a_2\}$ $\{a_5, a_3\}$ $\{a_6\}$ 。

为验证本文识别方法(简称方法1)的有效性,针对上述实例,采用基于遗传算法的服务识别方法^[10](简称方法2)和基于聚类分析的企业信息系统业务构件识别方法^[2](简称方法3)进行构件识别,并对结果进行比较。这2种方法是目前比较有代表性的构件识别方法,在文献中虽然都是应用在传统的业务构件识别领域,但其识别算法同样可运用在服务构件识别领域,方法2和方法3的参数设置按照文献^[10]与文献^[2]中的相关说明。方法2采用遗传算法自顶而下对业务模型进行识别,取得了不错的效果;方法3应用聚类算法识别企业信息系统的业务过程构件和业务实体构件,从而获得高内聚、低耦合的业务构件。

3种方法的识别结果如表1所示。

表1 3种方法的识别结果比较

名称	识别结果	构件粒度	内聚性	耦合性
方法1	$\{s_1, s_2, s_4\}$ $\{s_3, s_6\}$ $\{s_5\}$	3.4	4.5	2.5
方法2	$\{s_1, s_2, s_4, s_6\}$ $\{s_5, s_3\}$	3.6	2.7	3.3
方法3	$\{s_1, s_2, s_4\}$ $\{s_5, s_3\}$ $\{s_6\}$	4.0	3.2	4.6

实验结果表明,方法1在内聚性和耦合性两项指标上,要好于方法2和方法3;在构件粒度指标上,方法2和方法3的粒度均略高于方法1。但考虑到构件粒度三者之间相差甚微,对综合指数的影响较小,在此可以忽略不计。因此可以得出结论:方法1识别的服务构件模型综合度量指数要好于方法2和方法3。

7 结束语

从业务模型中提取可复用的服务构件是SOA下基于构件的软件开发的重要手段,本文在分析面向服务架构下业务建模元模型构成元素及其关系的基础上,提出一种基于活动关联度的服务构件识别方法。这种方法考虑了影响活

动之间关联度的因素,从4个方面计算活动之间的关联性,将具有较高关联度的一组活动聚类为一个符合规范的服务构件,为SOA下的基于服务构件的软件系统重构奠定了基础。通过实际系统开发中的应用,将本文方法的识别结果同现有方法相比,证明本文方法设计的构件更合理,更符合实际需求,从而验证了本文方法的实际应用价值。

参考文献

- [1] 王忠杰, 战德臣, 徐晓飞. 业务构件识别方法研究综述[J]. 计算机集成制造系统, 2007, 13(4): 797-805, 811.
- [2] 徐玮, 尹宝林, 李昭原. 企业信息系统业务构件设计研究[J]. 软件学报, 2003, 14(7): 1213-1220.
- [3] 孟凡超, 战德臣, 徐晓飞. 基于领域业务模型的可重用构件设计方法[J]. 计算机集成制造系统, 2006, 12(9): 1402-1410.
- [4] Jamshidi P, Sharifi M, Mansour S. To Establish Enterprise Service Model from Enterprise Business Model[C]//Proc. of IEEE International Conference on Services Computing. Honolulu, USA: IEEE Press, 2008: 93-100.
- [5] Arch-Int S, Batanov D N. Development of Industrial Information Systems on the Web Using Business Components[J]. Computers in Industry, 2003, 50(2): 231-250.
- [6] Alahmari S, Zaluska E, de Roure D. A Service Identification Framework for Legacy System Migration into SOA[C]//Proc. of the 7th International Conference on Services Computing. Miami, USA: IEEE Press, 2010: 614-617.
- [7] Alahmari S, de Roure D, Zaluska E. A Model-driven Architecture Approach to the Efficient Identification of Services on Service-oriented Enterprise Architecture[C]//Proc. of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops. [S. l.]: IEEE Press, 2010: 165-172.
- [8] Ma Qian, Zhou Nianjun, Zhu Yanfeng, et al. Evaluating Service Identification with Design Metrics on Business Process Decomposition[C]//Proc. of IEEE International Conference on Services Computing. Bangalore, India: IEEE Press, 2009: 160-167.
- [9] Wu Lei. The Model-driven Development Method Based on the Improved Feature Analysis[J]. Journal of Advanced Manufacturing Systems, 2011, 10(1): 151-158.
- [10] Kazemi A, Rostampour A, Jamshidi P. A Genetic Algorithm Based Approach to Service Identification[C]//Proc. of IEEE World Congress on Services. Washington D. C., USA: IEEE Press, 2011: 339-346.

编辑 陆燕菲

