

一种基于最小覆盖的复杂 Web 服务组合方法

朱正东¹, 伍卫国¹, 胡亚红², 李增智¹

(1. 西安交通大学计算机科学与技术系, 710049, 西安; 2. 浙江工业大学计算机系, 310032, 杭州)

摘要: 针对复杂 Web 服务的组合问题, 提出了基于最小覆盖的 Web 服务组合方法(CWSCM). 该方法从需求目标出发, 将逻辑理论中的逻辑函数化简的最小覆盖思想运用于基于本体的多层次服务匹配的过程之中, 从而建立了最小服务匹配集合, 同时构造了优化的有序服务组合图, 并将该图的形式化表示保存于扩充的 Web 服务描述之中, 以便复用. 实验表明, CWSCM 可成功地组合复杂的 Web 服务, 保证服务组合的成功率和效率, 当组合满意度接近 70% 时, 组合服务的成功率最高, 而满意度对组合效率的影响比较小.

关键词: Web 服务; 复杂 Web 服务; 服务组合; 最小覆盖

中图分类号: TP301 **文献标志码:** A **文章编号:** 0253-987X(2008)08-0945-05

Minimum Coverage Method for Complicated Web Service Composition

ZHU Zhengdong¹, WU Weiguo¹, HU Yahong², LI Zengzhi¹

(1. Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China;

2. Department of Computer, Zhejiang University of Technology, Hangzhou 310032, China)

Abstract: A minimum coverage method called CWSCM is presented to deal with complicated Web service composition. According to desired objectives, the minimum coverage theory that simplifies logic functions in mathematical logic is applied to the multi-level services matching process to discover the desired Web services, and establish a minimum service matching set. At the same time, an optimal sequenced service composition graph is constructed, and its formal expression is saved into the extended Web service description for reuse. Experiments show that CWSCM can compose complicated Web services successfully, and that the success rate and efficiency of Web service composition can be ensured. When the satisfaction degree of the composition reaches about 70%, the algorithm can achieve the highest success rate. However, the satisfaction degree has little impact on the efficiency of the composition.

Keywords: Web service; complicated Web service; service composition; minimum coverage

现有的 Web 服务组合方法主要包括基于关键词匹配的方法^[1-2]及基于服务语义的方法^[3-6]. 前者的组合效率较高, 但没有考虑服务语义, 所以它的匹配准确性较差, 且仅适用于概念简单、领域服务较少的场合; 后者中有一些利用了“最短路径”算法^[7], 也有些利用了基于领域本体的动态组合方法, 它们都考虑了组合的质量, 解决了服务组合中的 2 个异构性问题, 从而提高了服务组合的准确性, 获得了最优

的服务组合, 但却牺牲了算法的效率. 同时, 上述方法所针对的服务组合均不存在复杂的并发服务情况, 大多数与实际应用不符, 因此适用范围受到限制, 可行性及实用性也受到影响.

本文对通用 Web 服务描述进行了扩充, 并给出了复杂 Web 服务的描述, 同时结合现有基于本体的 Web 服务的动态组合方法, 重点研究了复杂 Web 服务的动态组合; 利用逻辑函数化简的最小

覆盖思想,动态地建立了优化的有序服务组合图,描绘了复杂 Web 服务的组合内容,并对有序服务组合图进行了形式化描述,且将描述结果存储于服务之中,以便于复用。

1 Web 服务的描述

1.1 Web 服务的表示模型

大多数 Web 服务描述语言在具体的描述上均存在着一些不足,如出现了异构性问题^[8],缺乏行为约束和非功能属性描述,不能满足复杂 Web 服务要求等。针对上述问题,本文基于 Rolland 等人提出的目标结构^[9],给出了可用于复杂 Web 服务的描述。

定义 1 Web 服务可用一个六元组 $S_{Wi} = (V_i, T_i, C_i, I_i, O_i, Q_i)$ 来描述,其中动词加对象 $(V_i + T_i)$ 构成基本描述, C_i 包含了有序服务组合图 G 的形式化描述和服务功能描述 2 个部分, I_i 和 O_i 是输入、输出参数的集合, Q_i 是服务提供的质量指标。对于原子服务, G 为空。复杂 Web 服务的描述模型如图 1 所示。

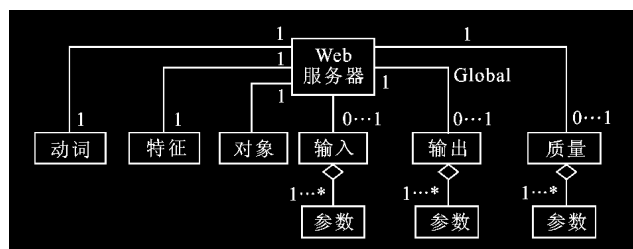


图 1 复杂 Web 服务的描述模型

1.2 Web 服务组合关系

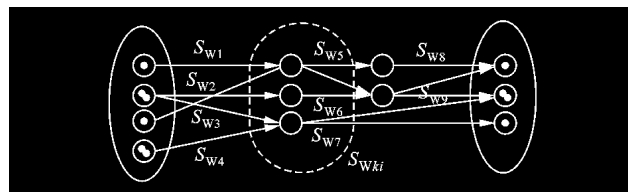
依据需求工程中的有关目标定义^[9]及构建服务过程的实际需要,服务之间应满足下列依赖关系和次序关系,以用于形式化地描述复杂 Web 服务的有序服务组合图 G 。

(1) 依赖关系。如果 S_{W1} 与 S_{W2} 为 2 个子服务, S_{W1} 必须满足的条件也是 S_{W2} 必须满足的,反之亦然,则称 S_{W1} 与 S_{W2} 之间为依赖关系,记为 $S_{W1} \wedge S_{W2}$ 。

(2) 次序关系。如果 S_{W1} 与 S_{W2} 为 2 个子服务, S_{W1} 满足的条件先于 S_{W2} ,则称 S_{W1} 相对于 S_{W2} 具有优先关系,记为 $S_{W1} \rightarrow S_{W2}$ 。

一个复杂 Web 服务的有序组合图 G 如图 2 所示。构成的服务组合图实际上也是一种服务状态图,所有起始节点的输入参数构成新服务的输入参数,所有结束节点的输出参数构成新服务的输出参数。

一个节点引出的所有边均属于一个过程,一个节点引入的所有边并不属于一个过程,2 个节点及其之间的连线均是一个成型的服务或服务的一部分。从图 2 可以看出,9 个服务之间的组合关系描述为 $((S_{W1} \wedge S_{W3}) \rightarrow S_{W5}) \wedge (S_{W2} \rightarrow S_{W6}) \wedge ((S_{W2} \wedge S_{W4}) \rightarrow S_{W7}) \rightarrow ((S_{W5} \rightarrow S_{W8}) \wedge ((S_{W5} \wedge S_{W6}) \rightarrow S_{W9}))$ 。



圆圈表示服务的开始和结束;黑点表示输入输出参数;箭头表示 Web 服务执行的过程

图 2 复杂 Web 服务有序组合图 G

2 复杂 Web 服务组合方法

2.1 最小覆盖思想

逻辑函数在卡诺图上的最小覆盖是:在卡诺图上,为了挑选数目最少的极大圈,优先选择必要的极大圈来覆盖卡诺图上全部标 1 的小方格的卡诺图,从而构成最小覆盖,见图 3。

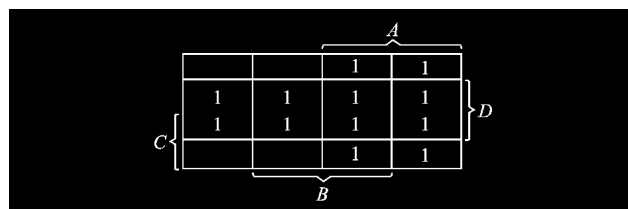


图 3 最简与或式 $A+D$

在基于最小覆盖的复杂 Web 服务组合方法 (CWSCM) 中,服务输出函数的参数与服务输入函数的参数有相同的部分,相同的部分称之为覆盖,那么最小覆盖是指用最少数量的并发服务输出函数覆盖某服务的输入函数,见图 4。图中的 3 个服务的输出函数分别为 $O_1(o_{11}, \dots, o_{15})$ 、 $O_2(o_{21}, \dots, o_{25})$ 及 $O_3(o_{31}, \dots, o_{38})$,小圆圈中的多个参数表示多个输出参数与一个输入参数的作用等同,即输入函数可被输出函数 $O_1(o_{11}, \dots, o_{15})$ 、 $O_3(o_{31}, \dots, o_{38})$ 覆盖。

2.2 基于最小覆盖的 Web 服务组合的概念和匹配方法

借鉴传统的构件匹配理论、中间件思想^[10]及文献^[7]中服务匹配层次的概念和方法,可将服务匹配划分为基本匹配、基调匹配和规约匹配 3 个层次。经

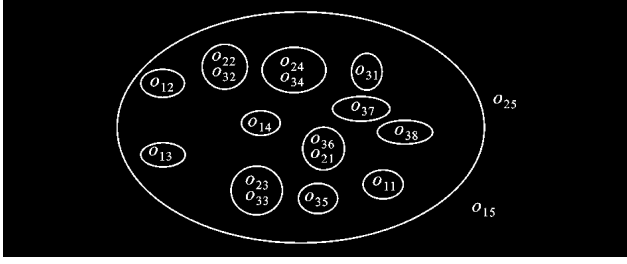


图4 输入函数的最小覆盖

由基本匹配筛选的候选服务优先参与基调匹配,再参与规约匹配,匹配中采用了领域本体及其推理,而在基调、规约匹配中使用了最小覆盖的思想。

定义2 对于2个概念 C_i, C_j ,如果在领域本体中 C_i 的属性集与 C_j 的属性集完全相同,即属性名称满足相似度要求,定义域完全一致,则称 C_i 和 C_j 语义相等,即 $C_i \equiv C_j$;如果 C_j 的属性集包含于 C_i 的属性集,则称 C_i 语义包含 C_j ,即 $C_i \supseteq C_j$;如果 C_i 的属性集与 C_j 的属性集部分相同,存在交集,则称 C_i 和 C_j 语义相交,即 $C_i \cap C_j \neq \emptyset$ 。

同理,对于2个概念集 S_{C_i}, S_{C_j} ,如果也存在 $S_{C_i} \equiv S_{C_j}, S_{C_i} \supseteq S_{C_j}$ 和 $S_{C_i} \cap S_{C_j} \neq \emptyset$,则称 S_{C_i} 与 S_{C_j} 语义相等、包含及相交。

定义3 服务请求可用一个五元组 $R_{WS} = (V, T, I, O, Q)$ 来描述,即由动词、对象、输入参数集、输出参数集和质量组成。动词反映请求者希望为客户所做的动作,对象是动作所影响的实体,输入参数是为实现服务请求所需要的条件,输出参数是在实现服务请求后提供给请求者的定量与定性指标,质量是客户提出的服务性能要求。

定义4 对于服务请求 $R_{WS} = (V, T, I, O, Q)$ 和 Web 服务集 $\{S_{W_i}\} = \{(V_i, T_i, C_i, I_i, O_i, Q_i), (i=1, \dots, n)\}$,如果 $O \subseteq O_1 \cup O_2 \cup O_3 \cup \dots \cup O_n$,且 $I \subseteq I', I' = I_1 \cup I_2 \cup I_3 \cup \dots \cup I_n$,则称 Web 服务集 $\{S_{W_i}\} (i=1, \dots, n)$ 满足服务请求,或部分满足服务请求,记为 $(R_{WS}, \{S_{W_i}\})$ 。如果 $O \subseteq O_1 \cup O_2 \cup O_3 \cup \dots \cup O_n$ 不存在,或者 $O \subseteq O_1 \cup O_2 \cup O_3 \cup \dots \cup O_n$ 且 $I_1 \cup I_2 \cup I_3 \cup \dots \cup I_n \subset I$,则没有 Web 服务 $\{S_{W_i}\}$ 满足服务请求 R_{WS} ,称为不满足服务请求。

定义5 对于服务请求 $R_{WS} = (V, T, I, O, Q)$,如果2个部分满足 R_{WS} 的 Web 服务集为 $\{S_{W_i}\}$ 和 $\{S_{W_j}\}$, $(O \cap O_i) \subseteq (O \cap O_j), O_i, O_j$ 分别为 $\{S_{W_i}\}$ 与 $\{S_{W_j}\}$ 输出的集合,则称 $\{S_{W_j}\}$ 是极大部分满足服务请求。对于 $\{S_{W_j}\}$,如果存在一个输出参数 $o_p \in (O \cap O_j)$,且 $o_p \notin O_i, O_i$ 是除 $\{S_{W_j}\}$ 以外的任意一个极大部分满足

R_{WS} 的 Web 服务集合 $\{S_{W_i}\}$ 的输出参数集合,则称 $\{S_{W_j}\}$ 是必要的极大部分满足服务请求。

定义6 对于服务请求 $R_{WS} = (V, T, I, O, Q)$ 的最小服务匹配,是由所有必要的极大部分满足 R_{WS} 的 Web 服务加上最少数量的极大部分满足 R_{WS} 的 Web 服务组成,则记为 $Last(R_{WS}, \{S_{W_1}, S_{W_2}, \dots, S_{W_i}, \dots, S_{W_n}\})$ 。最少数量的极大部分满足 R_{WS} 的 Web 服务是指包含不被所有必要的极大部分满足 R_{WS} 的 Web 服务所包含的输出参数的极大部分满足 R_{WS} 的 Web 服务。所以,最优服务组合是由所有最小服务匹配的集合构成的。

定义7 对于服务请求 $R_{WS} = (V, T, I, O, Q)$,假设 $\{\{S_{W_{k1}}\}, \{S_{W_{k2}}\}, \dots, \{S_{W_{ki}}\}, \{S_{W_{kj}}\}, \dots, \{S_{W_{kn}}\}\}$ 为最优服务组合,即所有最小服务匹配集合,则 $\{S_{W_{k1}}\}$ 为第一级最小服务匹配集合, $\{S_{W_{ki}}\}$ 为第 i 级最小服务匹配集合(见图2), $\{S_{W_{kn}}\}$ 为最后一级最小服务匹配集合。匹配满意度包括 R_{WS} 输入满足第一级最小服务匹配集合的所有输入的程度 ξ_0 ,各级所有输出满足下一级所有输入的程度 $\xi_i (i=1, \dots, n-1)$,以及最后一级最小服务匹配集合的所有输出满足 R_{WS} 输出的程度 ξ_n ,将所有匹配满意度中的最小值称为服务组合满意度 ξ ,则 $\xi = \min\{N(I_{k1} \cap I)/N(I_{k1}), N(O_{ki} \cap I_{kj})/N(O_{ki}), N(O_{kn} \cap O)/N(O)\}$,其中 $j=i+1 (i=1, \dots, n-1), N(\cdot)$ 表示集合中的参数个数, O_{ki}, I_{kj} 分别为第 ki 、第 kj 级的输出参数与输入参数的集合。

2.3 Web 服务组合算法

Web 服务的组合过程是从服务请求出发,使用构图法从后向前逐级找到最小服务匹配的过程,如果找到了最小服务匹配,则需要将找到的所有 Web 服务的输入作为前一级 Web 服务的需求,然后继续寻找前一级的最小服务匹配,直至找到所有服务的输入都包含用户需求,此时组合成功完成,否则组合宣告失败。可以看出,此过程与执行服务的过程相逆。本文将利用服务组合 (S_{LWC}) 满意度 ξ 来控制匹配效果。复杂 Web 服务组合算法如图5所示,需要说明的是:

- (1) 函数 (R_{WS}, S_W) 得到满足 R_{WS} 的服务集,并且必须满足 $\xi_i > \xi$,其中 S_W 为所有服务实例;
- (2) 函数 $Last(R_{WS}, \{S_{W_i}\})$ 得到的最小服务匹配为 $\{S_{W_{ki}}\}$;
- (3) 函数 $AddInput(I', I_{ki})$ 将服务集 $\{S_{W_{ki}}\}$ 的输入参数放入新服务集的参数集合之中;
- (4) 函数 $Push(S_{LWC}, \{S_{W_{ki}}\})$ 将 $S_{W_{1i}} \wedge S_{W_{2i}} \wedge \dots$

```

输入: 服务请求  $R_{WS}(V, T, I, O, Q)$ , 服务组合满意度  $\xi$ 
输出: 组合服务  $S_{LWC}$ 
1.  $LWC = \text{null}$ ;  $\text{Input} = \text{null}$ ;  $\text{Output} = \text{null}$ ;  $R_{WS0} = R_{WS}$ ;  $i = 0$ 
2.  $\{S_{W_i}\} = (R_{WS_i}, S_{W_i}, \xi_i > \xi)$ ;
3. if  $\{S_{W_i}\} = \text{null}$ , then 组合失败 else
4.    $\{S_{W_{ki}}\} = \text{Last}(R_{WS_i}, \{S_{W_i}\})$ ;  $\text{get}(k_i)$ 
5.    $\text{Input} = \text{AddInput}(I', I_{k_i})$ ; // 包含在  $I$  中的参数可
     作为需求参数集的无关项//
6.    $\text{Push}(S_{LWC}, \{S_{W_{ki}}\})$ ;
7.   if  $I \subseteq \text{Input}$  then  $\text{Output} = \text{get}(O_{k_i}, \{S_{W_{ki}}\})$  组合成功 else
8.     if  $\text{Input} \neq I$ , 组合失败 else
9.        $\text{Create}(R_{WS_{i-1}})$ , goto 2

```

图5 复杂 Web 服务组合算法

$\wedge S_{W_{ji}} \wedge \dots \wedge S_{W_{ki}}$ 压入 S_{LWC} ;

(5) 函数 $\text{Create}(R_{WS_{i-1}})$ 建立新请求, 并将 $\{S_{W_{ki}}\}$ 的输入参数集作为新需求的输出。

请求注册的 Web 服务要求客户用 XML 模式定义服务的输入、输出参数, 用 XML 文档表示用户人机界面。在客户执行服务的过程中, 组合成型的服务将根据输入、输出的参数及类型自动生成 XML 模式定义和 XML 文档。

3 实验结果与分析

本文参照文献[3]中的方法, 并取类似的数据集进行实验, 同时比较了本文方法(CWSCM)、基于服务语义的 Web 服务组合方法(BSCM)、基于领域本体的简单 Web 服务组合方法(DOSCM)的服务成功率和组合效率。

3.1 实验环境

硬件环境: CPU 为 Intel Pentium IV, 主频为 2.0 GHz, 内存为 1 GB, 操作系统为 Windows XP, 算法实现工具是 VC。实验数据: Web 服务实例为网上购物, 将 Google、Baidu 和 Amazon 等知名网站上的服务作为样本, 服务总数分别为 300、600、900、1 200、1 500、1 800、2 100、2 400、2 700。实验内容: 服务成功率和组合效率。

在实验中, 考察了服务组合满意度 ξ 分别为 10%、30%、50%、70%、90% 下的成功率和效率。

3.2 实验结果

实验 1 服务组合成功率实验。

在服务总数分别为 300、900、1 500、2 100、2 700 的 5 个数据集上对 100 个随机的服务请求进行服务组合, 组合的成功率如图 6 所示。从中看出, CWSCM 在某些满意度 ξ 上, 其成功率和其他 2 种

方法差距很小, 这说明基于本体的复杂 Web 服务动态组合是可行的。另外, 当满意度 ξ 逐步增大时, 成功率随之增大, 当满意度 $\xi = 70\%$ 时成功率最高, 继而成功率开始下降, 这说明满意度 ξ 对服务成功率的影响比较大。

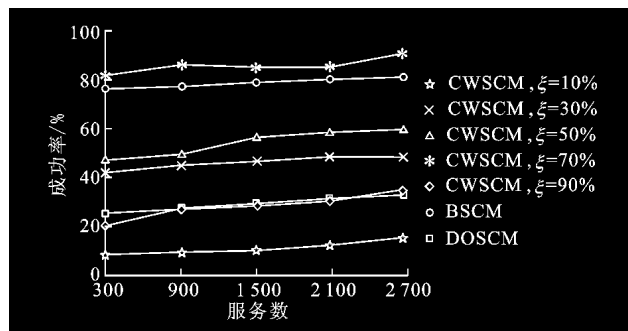


图6 服务组合成功率比较

实验 2 服务组合效率实验。

在服务总数分别为 300、600、900、1 200、1 500、1 800、2 100、2 400、2 700 的 9 个数据集上对 100 个随机的服务请求进行服务组合, 取平均组合时间作为组合效率, 结果如图 7 所示。从中看出, CWSCM 在某些满意度 ξ 上的效率低于 DOSCM 但高于 BSCM, 且曲线上速度非常慢, 这说明 CWSCM 与 DOSCM 在可扩展性方面具有类似的特性。BSCM 随服务总数的增加, 组合时间快速增长, 可扩展性较差。另外, 当满意度 ξ 较小时, CWSCM 响应时间很短, 效率很高, 随着满意度 ξ 的逐步增大, 效率随之降低, 但变化不是很大。

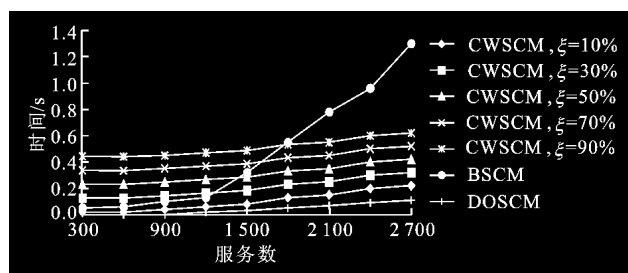


图7 服务组合效率比较

3.3 实验分析

BSCM 是基于语义的服务组合, 组合成功率较高, 但为了获取最优的结果, 需要反复扫描 WSC-Graph 中所有可能的路径, 所以算法受服务数量的影响很大, 可扩展性比较差, 效率也较低。DOSCM 与 CWSCM 在领域本体的支持下, 其服务组合可将语义与目标结构相结合, 所以二者的组合成功率差距很小。由于 DOSCM 和 CWSCM 一旦找到满足条件的服务组合, 其查找立即停止, 所以它们受服务总

数的影响较小,可扩展性也比较好。但是,CWSCM 适用性比 DOSCM 好,查全率更高,这说明利用最小满意度阈值可保证整个服务组合的质量。

通过实验结果分析可以看出,CWSCM 具有以下优点:①可扩展性好;②适用性广泛,查全率和效率比较高;③通过控制满意度 ξ 可使服务组合的成功率达到较高水平。

4 结 论

在 Internet 的应用中,Web 服务的组合非常重要。本文从需求目标出发,结合语义与层次匹配的方法,利用最小覆盖理论进行 Web 服务匹配,再利用构图法构造出有序服务组合图,从而实现了既适用于复杂 Web 服务,又可用于简单 Web 服务的服务组合,这样可最大程度地满足客户的需求。通过对有序服务组合图的形式化描述,为组合 Web 服务的复用和调用奠定了基础。实验表明,本文方法可以实现服务请求的自动组合,保证了服务的成功率、效率和实用性。今后的工作是,开发一种基于目标本体的 Web 服务组合管理系统,进一步完善目标本体,扩展组合算法的智能自适应能力。

参考文献:

- [1] PONNEKANTI S R, FOX A. SWORD: a developer toolkit for Web service composition [C]//Proceedings of International World Wide Web Conference, New York, USA: ACM,2002:83-107.
- [2] BENATALLAH B, DUMAS M. The self-serve environment for Web services composition [J]. IEEE Internet Computing, 2003,7(1):40-48
- [3] 李曼,王大治,杜小勇,等. 基于领域本体的 Web 服务动态组合 [J]. 计算机学报, 2005, 28(4):644-650
LI Man, WANG Dazhi, DU Xiaoyong, et al. Dynamic composition of Web service based on domain ontology [J]. Chinese Journal of Computers, 2005,28(4):644-650.
- [4] MA Jiangang, ZHANG Yanchun, LI Minglu. OM-WSC: an ontology-based model for Web service composition [C]//Proceedings of The 5th International Conference on Quality Software. Piscataway, NJ, USA: IEEE, 2005:464-469.
- [5] ZHANG Yue, YU Tao, RAMAN K, et al. Strategies for efficient syntactical and semantic Web services discovery and composition [C]//Proceedings of The 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing. Piscataway, NJ, USA: IEEE, 2006:72-72.
- [6] 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.
HU Jianqiang, ZOU Peng, WANG Huaimin, et al. Research on Web service description language QWSDL and service matching model [J]. Chinese Journal of Computers, 2005,28(4):505-513.
- [7] ZHANG Ruoyan, ARPINAR B, ALEMAN-MEZAN B. Automatic composition of semantic Web services [C]//Proceedings of International Conference on Web Services. Las Vegas, NV, USA: CSREA Press, 2003:38-41.
- [8] JORGE C, AMIT S. Semantic e-workflow composition [J]. Journal of Intelligent Information Systems, 2003, 1(3):191-225.
- [9] ROLLAND C, SOUVEYET C, BEN-ACHOUR C. Guiding goal modeling using scenarios [J]. IEEE Transactions on Software Engineering, 1998,24(12): 1055-1071.
- [10] HUANG Gang, LIU Xuanzhe, HONG Mei. SOAR: towards dependable service-oriented architecture via reflective middleware [J]. International Journal of Simulation and Process Modeling, 2007,3(1):55-65.

(编辑 苗凌)