

一种关联感知的组合服务重选取方法

张岳松¹, 张 斌¹, 张长胜¹, 张明卫²

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 东北大学 软件学院, 辽宁 沈阳 110819)

摘 要: 已有的组合服务重选取方法大都假定任务之间互相独立,然而在实际场景中,由于基本服务间存在着 QoS 关联关系,使得某些任务之间也存在着关联关系,从而导致这些已有的方法难以获取最佳的调整方案. 针对该问题,通过扩展 OWL-S,给出了一个支持服务间关联关系的 QoS 模型. 基于该模型,给出了一个支持任务间关联关系的重选取算法,该算法将具有关联关系的任务作为一个任务单元,以关联服务作为任务单元的备选服务进行重选取. 实验结果表明,与按照任务间独立的方式进行重选取相比,该方法能有效提高重新选取出的组合服务质量.

关 键 词: 重选取; QoS; 关联关系; OWL-S; 组合服务

中图分类号: TP 311.5

文献标志码: A

文章编号: 1005-3026(2015)04-0469-05

Composite Service Reselection Method of Correlation-Aware

ZHANG Yue-song¹, ZHANG Bin¹, ZHANG Chang-sheng¹, ZHANG Ming-wei²

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. School of Software, Northeastern University, Shenyang 110819, China. Corresponding author: ZHANG Bin, professor, E-mail: zhangbin@mail.neu.edu.cn)

Abstract: Most of the existing researches for composite service reselection method are based on the assumption that the tasks involved are independent. However, in practical scenarios, the QoS of some candidate services have correlations with other services, which leads to the reselected results not to be the optimal one in the real executing environment for these existing reselection methods. To solve this problem, a QoS model supporting the correlations between services was given by extending the OWL-S framework. Based on the proposed model, a composite service reselection method considering the correlations among tasks was provided, in which the correlated tasks were regarded as a task unit and the correlated services of each task unit were regarded as its candidate service set. Experimental results showed that compared with the reselection methods considering the tasks independent, the proposed approach could effectively improve the quality of composite service reselected.

Key words: reselection; QoS; correlation; OWL-S; composite service

组合服务的重选取是指重新选取某些服务绑定受到失效影响的任务流程片段,一方面这些服务需要满足任务描述的功能和接口结构等要求,另一方面重新选取出的服务不能偏离用户设定的约束. 文献[1-3]在重选取的过程中假定 workflow 中的任务是相互独立的,每个任务绑定的服务与其他任务无关. 然而在实际的执行过程中普遍存在着一个场景:考虑一个订购机票旅行的计划,中

国航空与假日酒店之间具有合作伙伴关系,选择假日酒店的情况下,中国航空可以提供打折的机票. 因此为了获得优惠的价格,订票任务的选取结果决定了酒店预订任务的选取结果,进而使得两个任务之间具有关联关系. 以任务间互相独立的方式考虑组合服务的重选取无法获得服务间的关联 QoS,进而降低了重选取的质量. 文献[4-7]给出了服务间关联关系的具体定义及相应的处理

收稿日期: 2014-03-09

基金项目: 国家自然科学基金资助项目(61100090); 中央高校基本科研业务费专项资金资助项目(N110604002, N120604003, N120804001); 沈阳市科技计划项目(F12-277-1-80).

作者简介: 张岳松(1986-),男,河南濮阳人,东北大学博士研究生; 张 斌(1964-),男,辽宁开原人,东北大学教授,博士生导师.

方法,但没有给出一个清晰的服务间关联关系的描述和声明,难以将服务间的关联关系应用于组合服务的选取或重选取。

为了解决该问题,本文通过扩展 OWL-S,给出了一个描述关联服务 QoS 的模型,使得在选取或重选取过程中能够有效地刻画服务间关联关系对 QoS 的影响. 基于该模型,给出了一个支持服务间关联关系的重选取算法。

1 关联服务模型

1.1 关联服务的 OWL-S 描述

基于 OWL-S^[8],本文给出了一个表述不同类型 QoS 的模型,如图 1 所示. 通过扩展 Profile 描述了 Web 服务的两类 QoS. Profile, Parameter 与 Condition 是 OWL-S 中的初始类, QoS, DefaultQoS 与 CorrelatedQoS 是本文扩展的类,其中 DefaultQoS 表示默认的 QoS, CorrelatedQoS 表示关联 QoS. QoS 是 DefaultQoS 与 CorrelatedQoS 的超类,它具有 3 个数据类型的属性: hasCost, hasResponseTime 与 hasReliability 分别描述具体的价格、响应时间与可靠性. 同时添加了一个对象属性 depends, 把类 CorrelatedQoS 与 Condition 联系在一起,其表示的语义为 CorrelatedQoS 依赖条件 Condition.

设 q^i 表示 Web 服务的某一个 QoS 属性,当 i 取不同的值时, q^i 就表示不同的 QoS 属性,它可以表示为价格、响应时间、可靠性等. 那么具有 QoS 关联关系的服务定义如下。

定义 1 具有 QoS 关联关系的服务. 给定一组服务 $GWS = \{s_1, s_2, \dots, s_k\}$, 当这些服务一起使用时,存在某些服务 $\{s_a, s_{a+1}, \dots, s_{a+j}\} \subseteq GWS$, 它们的质量属性 q^i 受到 GWS 中其他服务的影响,则称 GWS 包含的服务之间具有 QoS 关联关系。

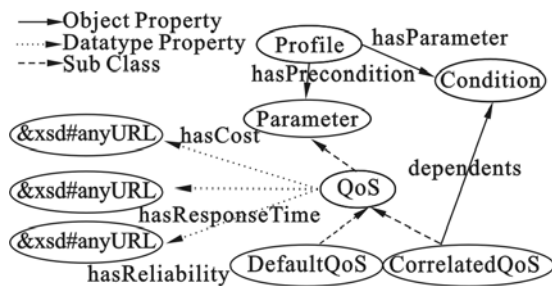


图 1 扩展 OWL-S Profile 得到关联 QoS

Fig. 1 Correlated QoS described by extension OWL-S Profile

1.2 关联服务的 QoS 模型

某一个任务 t 对应的备选服务 s 的质量属性 q^i 的值可以表示为

$$sv(s, q^i) = \{(v_1, c_1), (v_2, c_2), \dots, (v_m, c_m)\}.$$

式中, (v_u, c_u) 表示满足条件 c_u 的情况下, 服务 s 的质量属性 q^i 的值为 v_u . 本文利用表达式 $selected(t_u) \in S_u$ 表示选择备选服务集 S_u 中的服务绑定任务 t_u . 因此条件 c_u 可以形式化地表示为 $selected(t_u) \in S_u \wedge selected(t_{u+1}) \in S_{u+1} \wedge \dots \wedge selected(t_{u+k}) \in S_{u+k}$, 其中 $t_u, t_{u+1}, \dots, t_{u+k}$ 表示与 t 具有关联关系的任务, S_u, \dots, S_{u+k} 表示与服务 s 具有关联关系的备选服务的集合, 符号 \wedge 表示逻辑“与”操作. 由于服务 s 不能为质量属性 q^i 同时提供 2 个不同的值, 因此在选取过程中只能满足某一个条件。

进一步, 令等式 $a_u = 1$ 表示表达式 $selected(t_u) \in S_u$, $a_u = 0$ 表示表达式 $selected(t_u) \notin S_u$ (任务 t_u 没有绑定服务集 S_u 中的服务). 条件 c_u 的真值表如表 1 所示, 只有当变量 $a_u, a_{u+1}, \dots, a_{u+k}$ 的值都为 1 时, c_u 的值才能为“真”(逻辑“与”操作). 因此条件 c_u 还可以表示为 $a_u = 1 \wedge a_{u+1} = 1 \wedge \dots \wedge a_{u+k} = 1$, 此时服务 s 的质量属性 q^i 的值为 v_u .

表 1 条件 c_u 的真值表
Table 1 Truth table of condition c_u

条件 c_u	a_u	a_{u+1}	\dots	a_{u+k}
满足 c_u	1	1	\dots	1
不满足 c_u	other cases			

2 服务组合模型

在组合服务流程中包含循环控制结构, 可以通过循环展开等技术^[9]将循环控制结构展开为顺序控制结构. 经过处理后, 组合服务流程可以建模为有向图模型. 设工作流程包含 P 个执行路径, 每个执行路径不包含分支控制结构. 执行概率 $freq_p (p \in P)$ 是某一个分支路径的概率. T_p 表示执行路径 p 包含的任务集. 路径 p 的子路径可以表示为 sp , 它不包含并行控制结构. 组合服务的聚合函数可以表示为

$$\max \sum_{p=1}^P freq_p \cdot score_p(EP_p). \quad (1)$$

其中, EP_p 是执行路径 p 的执行计划, 它可以表示为一个特征向量 $[x_{ij}] (i \in T_p, j \text{ 表示绑定任务 } i \text{ 的一个服务})$. 目标值 $score(EP_p)$ 可以表示为

$$\text{score}(\text{EP}_p) = \sum_{n=1}^3 w_n \cdot v_n^p(\text{EP}_p). \quad (2)$$

式中: n 为质量属性 q^n 的标识; w_n 为 q^n 的权重; q^n 为进行归一化处理的结果. 对质量属性 q^n 的约束为

$$\sum_{i=1}^m \sum_{j=1}^{t_i} q_{ij}^n \cdot x_{ij} \leq Q_c^n, x_{ij} \in \{0, 1\}. \quad (3)$$

在组合服务 CS 中, 具有关联关系服务的 QoS 会受到其他服务的影响, 因此在提取质量属性 q^n 的真实值时, 需要检测当前的绑定关系是否满足服务条件. 一个服务 s 的属性 q^n 的值可以表示为 $\text{sv}(s, q^n) = \{(v_1, c_1), (v_2, c_2), \dots, (v_m, c_m)\}$, 且 $\text{selected}(t_i) \in S_i$ 表示绑定任务 t_i 的服务来自于服务集 S_i . 同时, 参数 x_{ij} 表达了服务 s_j 与任务 t_i 之间的绑定关系, 若 $x_{ij} = 1$, 则表示服务 s_j 绑定了任务 t_i . 如果 $s_j \in S_i$, 那么 $x_{ij} = 1$ 与等式 $a_i = 1$ 表达的意义相同. 设 $S_i = \{s_j, s_{j+1}, \dots, s_{j+u}\}$, 由于任务 t_i 在一次选取过程中只能绑定 S_i 中的一个服务, 所以 $\sum_{j \in \{j, \dots, j+u\}} x_{ij}$ 的值只能为 1 或者 0, 因而参数 $a_i = \sum_{j \in \{j, \dots, j+u\}} x_{ij}$.

如果条件 c_i 为 $a_i = 1 \wedge a_{i+1} = 1 \wedge \dots \wedge a_{i+d} = 1$, 意味着参数 $a_i, a_{i+1}, \dots, a_{i+d}$ 的值全部为 1 的情况下才能满足条件 c_i . 因此 c_i 就可以表示为 $\prod_{i \in \{i, \dots, i+d\}} a_i$, 则服务 s 的质量属性 q^n 的值可以表示为

$$\text{sv}(s, q^n) = v_1 \cdot c_1 + \dots + v_m \cdot c_m + v_0 \cdot (1 - \sum_{i=1}^m c_i). \quad (4)$$

式中, v_0 表示服务 s 在质量属性 q^n 上的默认发布值.

3 支持关联服务的重选取算法

当质量属性 q^n 的值相对于阈值 Δq^n 发生较大变化时, 会导致违反用户的约束^[2]. 在进行重选取前, 首先需要确定执行异常的组合服务的剩余任务流程. 文献[1]给出了一个确定剩余任务流程的方法, 利用该方法可以获得成功执行的服务, 剔除不可能再执行的分支, 并更新循环控制结构的迭代次数. 本文的重点是在考虑任务间关联关系的情况下, 如何对剩余任务流程进行重选取.

考虑任务间关联关系的情况下, 某些已经完成的任务可能会影响未执行任务与具体服务的绑定关系. 如图 2 所示, 任务 t_1, t_2 与 t_3 属于已经完成的任务, 并且 $\langle t_3, t_4, t_5 \rangle$ 是一组具体有关联关系的任务. 对任务 t_4 和 t_5 选取服务时, 首先需要

考虑的是与服务 s_{32} (绑定任务 t_3 的服务) 具有 QoS 关联关系的服务. 如果与 s_{32} 具有关联关系的服务实例无法满足用户的要求也需要重新对任务 t_3 进行重选取.

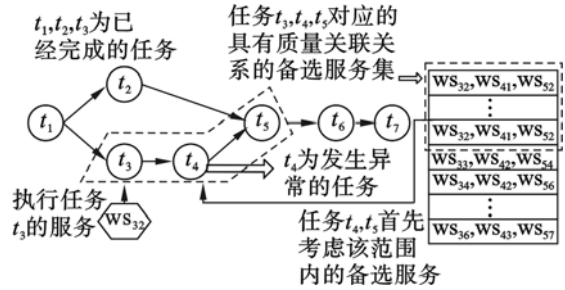


图2 尚未执行的任务与已经完成的任务间的关联关系
Fig. 2 Correlations between completed tasks and unexecuted tasks

组合服务剩余任务流程主要分为 3 类任务: ①独立于其他任务; ②具有关联关系的任务, 并且不受已经完成任务的影响; ③与已经完成的任务具有关联关系的任务. 支持关联关系的重选取算法的步骤如下.

1) 更新服务组合模型的参数: 已经完成的任务集为 $\text{completed_tasks} = \{t_1, t_2, \dots, t_u\}$, 绑定这些任务的服务为 $\text{executed_services} = \{s_1, s_2, \dots, s_u\}$, 再把表示 completed_tasks 与 executed_services 之间绑定关系的参数 x_1, x_2, \dots, x_u 的值都设置为 1. 将集合 executed_services 包含的服务在实际执行过程中的价格与响应时间代入服务组合模型, 并把这些服务在组合模型中对应的可靠性设置为 1.

2) 确定第②类任务的备选服务集: 设一组关联任务集为 $\text{correlated_tasks} = \{t_i, t_{i+1}, \dots, t_{i+d}\}$, 它对应的具有 QoS 关联关系的备选服务集为 $\text{cs_set} = \{\text{cs}_1, \text{cs}_2, \dots, \text{cs}_m\}$, 其中, 元素 $\text{cs}_j \in \text{cs_set}$ 是一组具有关联关系的备选服务. 任务集 $\text{cc_tasks} = \{t_i, t_{i+1}, \dots, t_{i+l}\} \subset \text{correlated_tasks} (l < d)$ 是一组已经完成的任务, 绑定 cc_tasks 的 Web 服务为 $\text{completed_cs} = \{s_i, s_{i+1}, \dots, s_{i+l}\}$. 然后对于任意一组关联服务 $\text{cs}_j \in \text{cs_set}$, 如果满足条件 $c: \text{completed_cs} \subset \text{cs}_j$, 则进行操作 $\text{cans}_k = \text{cs}_j - \text{completed_cs}$. 并将 cans_k 并入集合 canss , 直到不存在满足条件 c 的关联服务为止, 那么服务集合 canss 则是第②类任务首先考虑的备选服务集. 如果 canss 包含的服务无法满足要求, 则需要对 correlated_tasks 包含的所有任务进行重选取.

3) 将以上两步得到结果代入第 2 节的服务组合模型, 利用 0-1 规划算法得到重选取的结果, 0-1 规划算法的求解过程不再陈述.

4 实验分析

比较考虑了关联服务的重选取方法(简称为服务关联的重选取)与认为服务间互相独立的重选取方法^[1](服务独立的重选取)的效率与效果. 由于真实运行场景难以模拟,因此本文采用仿真数据生成的方法进行模拟.

4.1 仿真数据的生成

本实验运行的 PC 硬件配置为 Intel(R) Core i5 CPU 3.3 GHz, 4 GRAM. 软件环境采用了 Windows7 操作系统, JDK 1.7, Eclipse. 本文设定了 8 个任务, 组成一个流程, 并随机生成每个任务对应的具体服务的价格、响应时间和可靠性.

再从该任务流程中取 4 个子任务, 并为这些任务选择一定比例的具体服务作为支持 QoS 关联关系的备选服务. 生成关联服务 QoS 的方式: 设定服务之间的关联因子 $\mu \in \{0.2, 0.3, 0.4\}$, μ 代表每 2 个服务之间的 QoS 关联程度. 如果 μ 的值为 0.3, 它表示关联服务同时调用时, 它们的价格在默认发布值的基础上减少 30%, 可靠性提高 30% (且不能超过 100%), 响应时间也减少 30%, μ 的值可以通过随机的方式确定. 这两组任务对应的关联服务的比例是可以调整的, 初始设定为 50%.

对于考虑了关联关系的重选取方法则使用服务间的关联 QoS, 没有考虑任务间关联关系的方法则使用默认的 QoS. 本文假定第二个任务出现了异常, 然后对剩余的任务进行重选取.

4.2 组合服务重选取效率

首先设定每个任务对应的候选服务不变, 调整关联服务的比例, 比较 2 种方式的效率. 执行结果如图 3 所示, 在进行测试时重复选取 6 次, 计算重选取的平均执行时间.

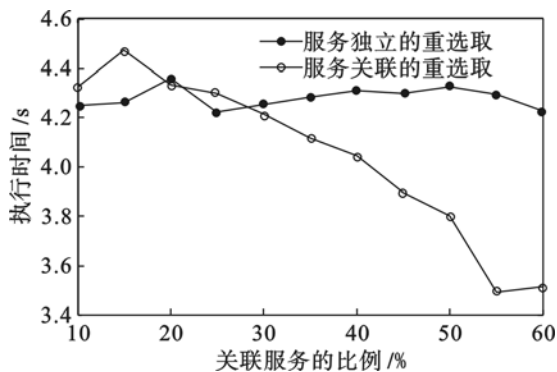


图 3 关联比例对执行时间的影响
Fig. 3 Effect of correlation proportion on execution time

由图 3 可以看出服务独立的重选取方法执行效率与关联服务的比例变化并没有关系. 这是因为该方法并没有考虑任务之间的关联关系, 所以它的执行效率没有较大的波动. 服务关联的重选取方法执行效率开始时呈上升的趋势, 随后随着关联服务比例上升开始下降. 这是因为关联服务的比例比较小的情况下, 关联服务的可选范围也比较少, 很多情况下难以满足 QoS 需求. 此时还要搜索整个服务空间中的备选服务, 这种情况下该方法的执行时间必然较长. 当关联服务的比例上升时, 服务关联的重选取方法大部分情况下可以通过搜索具有关联关系的服务满足 QoS 需求, 此时无需再搜索整个服务空间中的服务, 因此它的执行时间低于服务独立的重选取方法. 然而随着关联服务的比例逐步升高, 服务关联重选取方法的效率也在降低, 这是因为其搜索空间的范围在逐步增加.

图 4 将关联比例固定为 60%, 通过增加候选服务的方式比较 2 种算法的执行效率. 由图 4 可以看出, 2 种方法随着备选服务个数的增加, 选取的效率就越低. 因为服务关联的重选取方法是以关联服务作为相关任务的备选服务集, 所以减少备选服务的个数, 其效率优于服务独立的重选取方法.

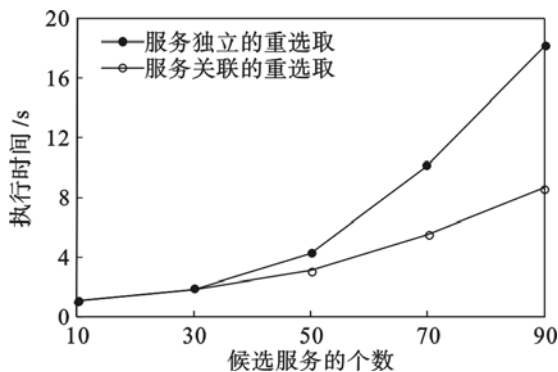


图 4 候选服务数对执行时间的影响
Fig. 4 Effect of candidate services number on execution time

4.3 组合服务重选取效果

图 5 为关联服务比例对 2 种算法调整结果质量的影响. 固定备选服务的个数为 70, 然后调整备选服务中关联服务的比例, 计算不同比例下组合服务的 QoS.

由图 5 可知, 由于备选服务固定不变, 因此服务独立的重选取方法获得的 QoS 也没有变化, 该方法不受关联服务比例的影响. 随着关联服务比例的提高, 关联服务的 QoS 范围变大, 因此服务

关联的重选取方法获得的 QoS 逐渐变大. 当关联服务比例比较低时, 服务关联的重选取方法利用关联服务得到的 QoS 优势并不明显. 只有当关联服务的比例逐步提高时, 服务关联的重选取方法获得的 QoS 也逐步提高.

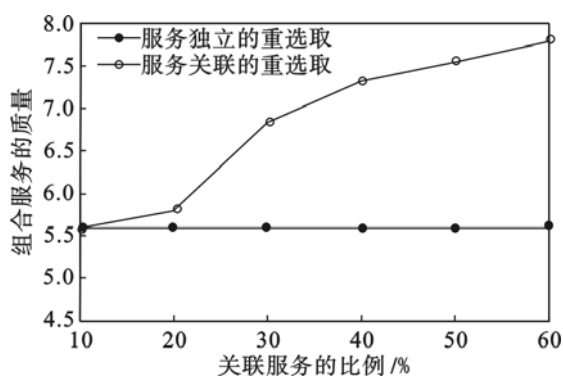


图5 关联服务比例对重选取质量的影响

Fig. 5 Effect of correlation proportion on reselection quality

图6将关联服务比例固定为60%, 逐步增加候选服务的个数, 对比了服务关联与服务独立的重选取方法获得的 QoS 与候选服务个数的关系. 由图6可知, 随着候选服务个数的增加, 服务独立的重选取方法的 QoS 逐步提高. 另一方面, 虽然关联服务的比例固定不变, 而随着候选服务数量的增加, 关联服务的数量也在增加, 因此考虑关联关系的方法重新选取出的组合服务的 QoS 也在提高. 在关联服务的数量逐渐增多的情况下, 关联服务的重选取方法选取出的组合服务的质量优于服务独立的重选取方法.

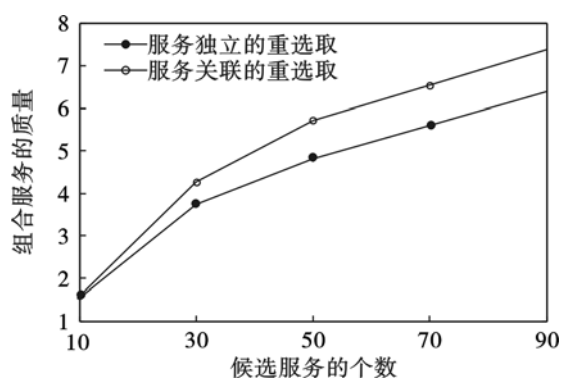


图6 候选服务个数对重选取质量的影响

Fig. 6 Effect of candidate services number on reselection quality

从上述实验可以看出, 由于关联任务的备选

服务集限定为具有 QoS 关联关系的服务, 减少了备选服务的个数, 因此在执行时间上优于假定任务独立的重选取方法. 同时, 考虑任务间关联关系的重选取方法, 以服务的关联 QoS 为选取基础可以获得更好的质量.

5 结 论

本文通过扩展 OWL-S 描述了服务间的关联 QoS, 并在此基础上, 给出了一个支持任务间关联关系的组合服务重选取算法. 与假定任务间独立的重选取算法相比, 本文的方法可以获得较高的质量. 在一定程度上减少了备选服务的范围, 提高了组合服务重选取的效率.

参考文献:

- [1] Canfora G, Penta D M, Esposito R, et al. A framework for QoS-aware binding and re-binding of composite Web service [J]. *Journal of System and Software*, 2008, 81 (10): 1754 - 1769.
- [2] Ardagna D, Pernici B. Adaptive service composition in flexible processes [J]. *IEEE Transactions on Software Engineering*, 2007, 33 (6): 369 - 384.
- [3] Cugola G, Pinto L S. QoS-aware adaptive service orchestrations [C]// IEEE 19th International Conference on Web Services. Honolulu, 2012: 440 - 447.
- [4] 张明卫, 张斌. 一种基于划分的组合服务选取方法 [J]. *计算机研究与发展*, 2012, 49 (5): 1005 - 1017.
(Zhang Ming-wei, Zhang Bin. A division based composite service selection approach [J]. *Journal of Computer Research and Development*, 2012, 49 (5): 1005 - 1017.)
- [5] Luo Y, Fan Y. Business correlation-aware modeling and services selection in business service ecosystem [J]. *International Journal of Computer Integrated Manufacturing*, 2013, 26 (8): 772 - 785.
- [6] Barakat L, Miles S. Efficient correlation-aware service selection [C]// IEEE 19th International Conference on Web Services. Honolulu, 2012: 1 - 8.
- [7] Feng Y, Ngan L D. Dynamic service composition with service-dependent QoS attributes [C]// IEEE 20th International Conference. Santa Clara, 2013: 10 - 17.
- [8] Martin D L, Burstein M H, McDermott D V, et al. Bringing semantics to Web services with OWL-S [J]. *World Wide Web*, 2007, 10 (3): 243 - 277.
- [9] Zeng L Z, Benatallah B, Ngu A, et al. QoS-aware middleware for Web services composition [J]. *IEEE Transactions on Software and Engineering*, 2004, 30 (5): 311 - 327.