

基于标注依赖图的服务组合方法

黄 炜^{1,2}, 朱巧明^{1,2}, 李培峰^{1,2}, 刘梅梅^{1,2}

(1 苏州大学 计算机科学与技术学院, 江苏 苏州 215006; 2 江苏省计算机信息处理技术重点实验室, 江苏 苏州 215006)

摘 要: 针对服务组合问题, 提出了一种既考虑服务输入/输出又考虑服务前提条件/执行效果的结构化组合方法. 根据服务请求查找服务组合路径, 得到服务依赖图, 然后通过依赖图中的依赖关系, 把依赖图转化为结构化流程模型, 除了标注结构化过程中的某些分支类型时, 需要一些额外的用户干预外, 整个过程是完全自动的.

关键词: 服务组合; 虚拟服务; 结构化组合; 标注依赖图; ADGSC

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2010)01-0094-05

Annotated Dependency Graph – Based Approach for Service Composition

HUANG Wei^{1,2}, ZHU Qiao-ming^{1,2}, LI Pei-feng^{1,2}, LIU Mei-mei^{1,2}

(1 School of Computer Science and Technology, Soochow University, Suzhou 215006, China;

2 Key Lab of Computer Information Processing Technology of Jiangsu Province, Suzhou 215006, China)

Abstract: This paper introduces a structured service composition method which considers both services' input/output and pre-condition/post-condition. It gets service dependency graph by means of searching service composition path based on service request. It transforms service dependency graph into a structured process model. It is fully automated, besides some additional user input needed to annotate dependencies with specific branching types.

Key words: service composition; virtual service; structured composition; annotated dependency graph; annotated dependency graph service composition (ADGSC)

1 引言

面向服务的计算正在成为开放、异构网络环境下构造松耦合、组合化分布式应用的一种主流技术. 服务是一个实体向另外的实体提供的能够满足其需求的功能. 在面向服务的体系结构中, 服务是基本的计算单元, 应用通过服务使用各种资源, 通过组合服务满足用户的业务需求. 目前, 具有代表性的服务规范主要有 Web Service、Grid Service^[1].

文中提出了一种从用户需求到产生结构化组合模型的自动组合方法. 该方法属于基于图模型的方法, 基于图模型的方法服务自动组合方法^[2]将服务之间的关系用有向图表达, 在途中寻找从输入到输

出过着从输出到输入的可达路径. 它由以下几步组成: 首先根据用户需求找到相应的匹配服务节点, 根据服务的输入/输出信息以及前置条件/后置条件构造服务间的依赖关系, 然后从前面的依赖关系中标注为 AND 和 XOR 两种类型, 最后把组合中服务构造为一个结构化模型.

2 基本概念

定义 1 物理服务: 一个物理服务定义为一个 5 元组: $S = (\text{Precondition}, \text{Input}, \text{Effect}, \text{Output}, Q)$.

定义 2 虚拟服务: 虚拟服务定义为一个五元组: $VS = (\text{Precondition}, \text{Input}, \text{Effect}, \text{Output}, \text{ServiceSet})$. 虚拟服务的语义信息来自各领域业务需求

收稿日期: 2008-12-30; 修回日期: 2009-02-25

基金项目: 国家自然科学基金项目(60673041); 江苏省高技术研究项目(BG2005020)

的抽象。

虚拟服务是对具体服务的抽象,它将一组具有相同功能,但服务约束不尽相同的服务抽象出来,并为用户提供统一的接口.用户面对的是虚拟服务,本文主要讨论的组合方法,后面的组合方法均是在虚拟服务层次上,不涉及物理服务的选择。

定义3 服务请求:服务请求定义为一个五元组: $SR = (Precondition, Input, Effect, Output, Constraints)$ 。

其中, $Precondition = \{p | p \text{ 是前提条件}\}$; $Input = \{i | i \text{ 是输入参数}\}$; $Effect = \{e | e \text{ 是执行服务后的显示效果}\}$; $Output = \{o | o \text{ 是输出参数}\}$; Q 是对服务的非功能属性描述,即服务的质量因素, $ServiceSet$ 为实现该虚拟服务的物理服务集合, $Constraints$ 为服务约束集. 分别使用 $S.Precondition$ 、 $S.Input$ 、 $S.Effect$ 和 $S.Output$ 表示服务 S 的前置条件、输入参数、执行效果和输出参数. $SR.precondition$ 、 $SR.Effect$ 、 $SR.Input$ 、 $SR.Output$ 和 $SR.Constraints$ 分别表示服务请求 SR 的前提条件、效果、输入、输出参数和服务约束。

定义4 可组合条件:若服务块 S_i 和 S_j 满足条件: $S_i.Effect \supseteq S_j.Precondition$ and $S_i.Output \supseteq S_j.Input$, 则称服务块 S_i 和 S_j 是可组合的. 服务块指单个服务或者已组合好的服务集合。

定义5 入口服务:若服务 S 和服务请求 SR 满足条件: $SR.Precondition \supseteq S.Precondition$ and $SR.Input \supseteq S.Input$, 则称服务 S 为 SR 的入口服务。

定义6 出口服务:若服务 S 和服务请求 SR 满足条件: $SR.Effect \subseteq S.Effect$ and $SR.Output \subseteq S.Output$, 则称服务 S 为 SR 的出口服务。

定义7 服务路径: 若存在一个服务序列 $S_1, \dots, S_i, S_{i+1}, \dots, S_n$, 若该序列满足以下条件:

- ① S_1 为入口服务;
- ② S_i 与 WS_{i+1} 是可组合的 ($1 \leq i < n$);
- ③ S_n 为出口服务。

则称该服务序列为满足服务请求的服务路径。

3 服务依赖图

3.1 抽象的服务依赖图

根据服务的前置条件/执行效果、输入参数/输出参数,定义服务间的依赖关系.若 S_i 和 S_j 是可组合的,则称服务 S_j 依赖于 S_i 。

服务依赖图:服务依赖图定义2元组: $G = (S, E)$; $S = \{s_1, s_2, \dots, s_3\}$; $E = \{(s_i, s_j) \in S \times S | \text{服}$

务 S_j 依赖于 $S_i\}$. 对依赖图做以下假定:

- (1) 图中不存在平行边;
- (2) 图中节点不存在自环;
- (3) 图中不存在封闭的有向环。

对依赖图引入一些辅助函数,对于一个服务 s , $pre(s)$ 表示 s 依赖的服务, $post(s)$ 表示依赖于 s 的服务:

$$pre(s) = \{x | (x, y) \in E \wedge y = s\}$$

$$post(s) = \{y | (x, y) \in E \wedge x = s\}.$$

下面以几个服务组成的买卖流程为例,用箭头表示服务间的依赖关系,图1为买卖过程的服务依赖图。

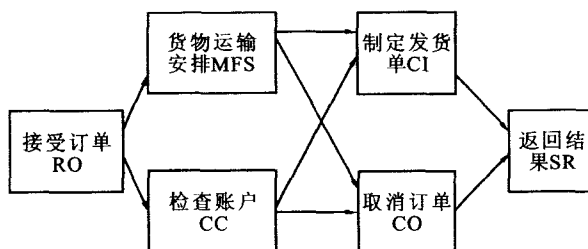


图1 服务依赖图

3.2 构建服务依赖图

用户给出服务请求 SR , 首先查找满足条件的出口服务和入口服务, 再根据服务间的依赖关系, 记录从出口服务到入口服务路径和服务节点, 这些路径上的边和服务节点就构成了抽象的服务依赖图, 图2为构建服务依赖图算法的流程图。

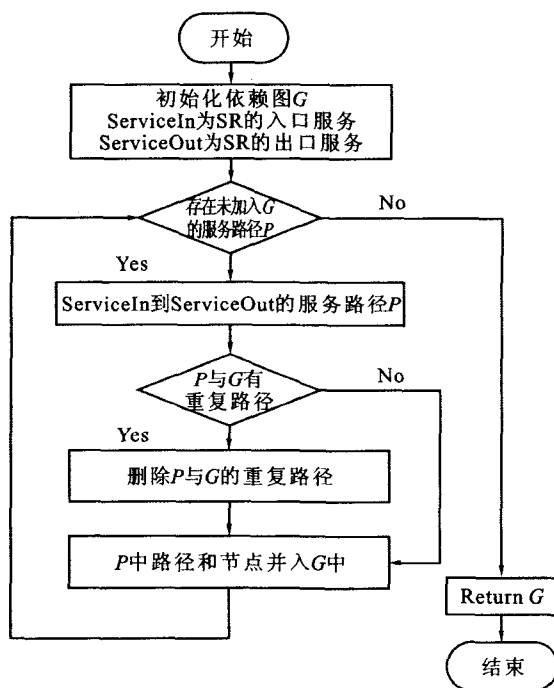


图2 构建服务依赖图算法流程图

带标注信息的服务依赖图需要具体化抽象依赖图中每个服务的输入和输出的依赖关系分支类型,根据 3.3 节的方法为抽象依赖图添加标注信息.而这些类型需要领域专家来具体化这些信息,这是因为信息依赖的本身通常并不是充分的依赖的类型.例如现在有 2 个服务接收的都是货物订单,如果 2 个服务都是安排货物运输,那他们之间的关系是 XOR,如果一个服务安排货物运输,另一个是计算总共的费用,那么 2 个服务都是需要的,他们之间的关系是 AND.

3.3 带标注信息的服务依赖图

带标注信息的服务依赖图^[3](如图 3)定义为 $G(S, E, \text{join}, \text{fork})$, (S, E) 为上面的依赖图, join 和 fork 标记服务输入和输出依赖关系的分支类型: AND 和 XOR.

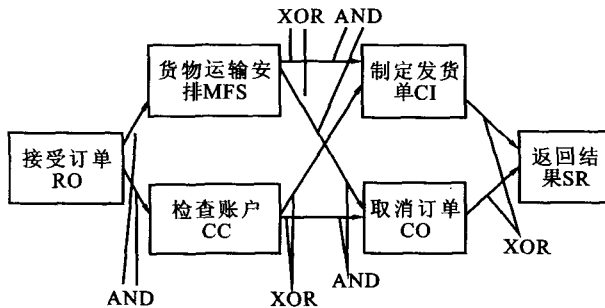


图 3 带标注信息的服务依赖图

AND 关系: 如果要想实现服务 S_1 的功能, 同时也要实现服务 S_2 的功能, 则称 S_1 和 S_2 具有 AND 关系, 记为 $\text{AND}(S_1, S_2)$.

XOR 关系: 只要实现服务 S_1 的功能或服务 S_2 的功能, 则称 S_1 和 S_2 具有 XOR 关系, 记为 $\text{XOR}(S_1, S_2)$.

当服务 s 依赖于多个 (> 1) 服务时标记为 $\text{join}(s)$, 当多个 (> 1) 服务依赖于服务 s 时标记为 $\text{fork}(s)$.

$\text{join}, \text{fork}: S \rightarrow \{\text{AND}, \text{XOR}\}$.

4 ADGSC 组方法

4.1 组合语言

我们这里选择一种层次化的视图, 叶子节点都是服务, 非叶子节点都是结构化块. 文中定义两种类型结构化块: (1) 组合结构化块 COMP, 并且 COMP 块被标注成 AND 或者 XOR; (2) 顺序结构化块 SEQ. 在图形语法^[3]中, 结构化块的开始和结束分别由 split 节点和 join 节点标记, 所以在每个 COMP 块

的开始和结束都增加虚拟的节点 (split/join).

结构化块的孩子节点由组合结构化块 COMP 的块集合和顺序结构化块 SEQ 的块序列组成. 如流程 $\text{COMP}\{\text{SEQ}[x, y], \text{SEQ}[Z]\}$, 表示 X 在 Y 之前执行, 并且 XY 和 Z 并行或者互斥的执行. 下面给出流程的形式化定义: 给定的服务集 S , 在 S 上的结构化组合的形式化归纳定义如下: (1) 每个服务 $s \in S$ 都是结构化组合; (2) 如果 $X_1, X_2, \dots, X_n$ 是结构化组合, 则他们只能是 $\text{SEQ}[X_1, X_2, \dots, X_n]$ 和 $\text{COMP}\{X_1, X_2, \dots, X_n\}$.

在结构化块上添加一些辅助函数. 给定的块 b , $\text{children}(b)$ 表示 b 的孩子节点, $\text{parent}(b)$ 表示 b 的父节点, $\text{services}(b)$ 表示 b 的直接或间接的孩子, 如 $\text{Services}(\text{COMP}\{\text{SEQ}[X, Y], \text{SEQ}[Z]\}) = \{X, Y, Z\}$.

4.2 ADGSC 算法

对算法中的一些公式和定义解释如下:

初始化阶段, 从 $\text{Initial}(S, E)$ 开始构建一个组合, 函数 $\text{Initial}(S, E)$ 返回不依赖其他服务的服务:

$$\text{Initial}(S, E) = \{s_2 \in S \mid \text{不存在 } s_1 \in S: (s_1, s_2) \in E\} \quad (1)$$

函数 $\text{constructBlock}(X)$ 是将给定的服务集 X 转换成一个组合块.

$\text{constructBlock} =$

$$\begin{cases} x, & \text{if } X = \{x\} \\ \text{COMP}\{\text{SEQ}[x] \mid x \in X\}, & \text{otherwise} \end{cases} \quad (2)$$

结构化组合的输出是由 S 中的服务迭代产生, 在每一次的迭代过程中, 一些服务被加入到 processed 中, 并被归入到 Toprocess 中. 当一个服务的所有依赖的服务都在 processed 中, 则这个服务将是下一步被归入 processed 中. 函数 $\text{next}()$ 返回下一步要被加入到 process 中的服务, 这些未被归入 processed 的服务的前置服务都已经在 processed 中.

$$\text{next}(\text{processed}, S) = \{s \in S \mid \text{pre}(s) \subseteq \text{processed}\} \setminus \text{processed} \quad (3)$$

在算法中还引入了影响 (Influence) 的概念. 如果 2 个服务都依赖于同一个服务, 称它们直接影响 (Directly Influence). 在图 4 中 B 和 C 都依赖于 A , 因此 B 和 C 是 Directly Influence. 注意的是每个服务都 Directly Influence 自己.

有 2 个服务 s_1 和 s_2 , 如果 s_1 和 s_2 互相直接影响 (Directly Influence) 或者有一个服务 s 同时直接影响 s_1 和 s_2 , 则称 s_1 和 s_2 相互影响. 影响关系是直接影响关系的传递闭包. 如在图 4 中, D 和 E 直接

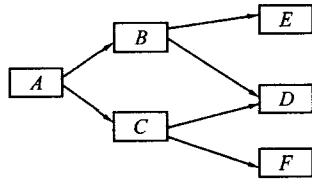


图 4 依赖图实例

影响,并且 D 和 F 也是直接影响,得到 E 和 F 是相互影响。

算法的流程图如图 5 所示,算法的输入为依赖图,返回值为满足依赖关系的结构化组合。

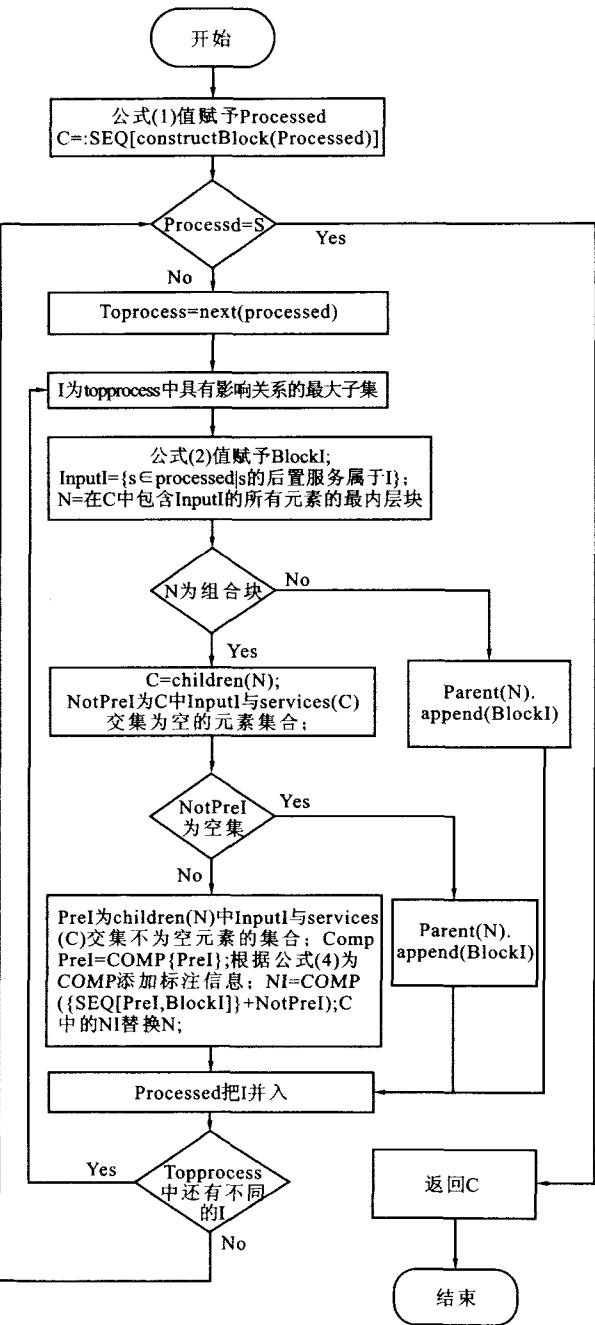


图 5 组合算法的流程图

4.3 构建标注信息

如在 3.3 节中提到,如果一个服务有多个后继服务时或者多个前驱服务时,这种依赖关系需要添加标注信息.带标注信息的服务依赖图中标注类型,能够直接对应到算法中的结构化组合的 COMP 节点.如果有一个 COMP 块 b ,一个服务 s 有多个后继服务依赖, b 的孩子服务依赖于服务 s , 那么 COMP 块标注为 $fork(s)$. 如果一个服务 s 有多个前驱服务依赖, 块 b 的每个孩子服务都是服务 s 依赖的, 那么 COMP 块标注为 $join(s)$ ^[4].

$type: COMP \rightarrow \{AND, XOR\}$

$type(b) =$

$$\begin{cases} fork(s), & \text{if for every child } c \text{ of } b, services(c) \cap post(s) \neq \emptyset \\ join(s), & \text{if for every child } c \text{ of } b, services(c) \cap pre(s) \neq \emptyset \end{cases}$$
 (4)

5 实验分析

通过 ADGSC 算法执行的 CPU 开销和成功率来验证 ADGSC 算法解决实际问题的有效性和可行性. 实验考虑了当虚拟服务数在 20、40、60、80、100 的情况下 ADGSC 算法的 CPU 运行时间开销. 在实验中有有效虚拟服务数为 16, 起始状态有 4 个干扰服务, 通过增加虚拟服务数量改变实验环境. 实验结果如图 6 所示. 实验结果表明 ADGSC 算法的有效性是明显的. 因本体的不完善和语义推理能力不足, 使得组合有一定的失败率, 但是并不影响算法的可行性.

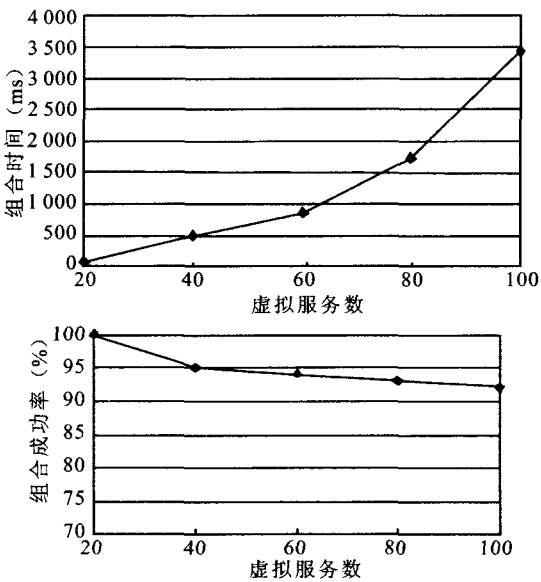


图 6 有效服务数为 16 时组合时间、组合成功率和虚拟服务数的关系

6 结束语

文中提出一种从用户服务请求到产生结构化服务组合的方法.虽然方法需要一些人工输入,但是基本上是完全自动的.算法的关键部分是给定的服务集和它们之间的依赖关系,然后完全自动的构建结构化组合来满足给定的依赖.该方法的特点是构建结构化组合,并且利用了基本工作流程模型.与文献[5]相比,不仅考虑单个服务间的依赖关系,还考虑了多个服务输出联合起来为一个服务提供输入的情况.与文献[2]相比,不仅考虑了输入/输出参数之间的依赖,更考虑了服务的行为约束条件.整个方法中并没有涉及到循环结构的处理,下一步工作将对算法进行扩展以处理循环结构^[6-7].

参考文献:

- [1] Foster I, Kesselman C, Nick J, et al. An open grid services architecture for distributed systems integration[C]// Physiology of the Grid. Chicago: University of Chicago, 2002.
- [2] Lang Qha, Su Syw. AND/OR graph and search algorithm for discovering composite Web services[J]. Journal of Web Services Research. 2005, 2(4): 46 - 64.

- [3] Rik Eshuis, Paul Grefen, Sven Till. Structured service composition[C]//BPM 2006. The Netherlands, LNCS 4102, 2006: 97 - 112.
- [4] Seyyed Vahid Hashemian, Farhad Mavaddat. A Graph - based approach to web services composition[C]// Proc of the Symp on Application and the Internet. Canada: Waterloo University, 2005: 183 - 189.
- [5] 付燕宁, 刘磊, 金成植. 基于服务链的 Web 服务组合方法[J]. 通信学报, 2007, 28(7): 92 - 97.
- [6] 李刚, 马修军, 韩燕波, 等. 动态网络环境下的透明服务组合[J]. 计算机学报, 2007, 30(4): 579 - 587.
- [7] 许兰, 朱巧明, 贡正仙, 等. 一种优化服务组合执行的设计方案[J]. 微电子学与计算机, 2007, 24(12): 144 - 146.

作者简介:

黄 炜 男, (1985 -), 硕士研究生. 研究方向为网格计算、中文信息处理.

朱巧明 男, (1963 -), 教授. 研究方向为中文信息处理、Web 信息处理、嵌入式系统.

李培峰 男, (1971 -), 副教授. 研究方向为中文信息处理和多文种信息处理、分布式信息系统、网格计算.

刘梅梅 女, (1981 -), 硕士研究生. 研究方向为中文信息处理、网格计算.

(上接第 93 页)

- [4] Jerome A Solinas. National security agency: an improved algorithm for arithmetica family of elliptic curves[R]. USA: Ft. Meade, MD20755.
- [5] King B. An improved implementation of elliptic curve cryptography over GF(2m) when using projective point arithmetic[M]. Berlin Heidelberg: Springer, 2001: 134 - 150.
- [6] Menezes A, Hankerson D, Hernandez J. Software implementation of elliptic curve cryptography over binary field[J]. Journal of Electronic Science and Technology of Chi-

na, 2003, 1(1): 1 - 24.

- [7] Luis Enrique Erro. GF(2m) arithmetic modules for elliptic curve cryptography electronics[J]. Computer Science Department, 2006(9): 689 - 696.

作者简介:

陈光化 男, 副教授. 研究方向为数字集成电路设计.

张晓莉 女, 硕士研究生. 研究方向为数字集成电路设计.