

基于模式匹配的组合Web服务发现方法

刘晶晶¹, 吴毅坚², 赵文耘¹

¹复旦大学计算机科学与工程系 上海 200433 ²复旦大学软件学院 上海 200433

052021164@fudan.edu.cn, wuyijian@fudan.edu.cn, wyzhao@fudan.edu.cn

摘要 现有的基于关键字或语义分类信息的Web服务发现基本上都是针对单个Web服务行为的。本文提出了一种基于模式匹配的组合Web服务的发现方法, 通过对Web服务的交互模式匹配性和组合模式匹配性的检查, 可以得到满足用户要求的组合行为的服务, 从而降低了用户参与组合服务的难度, 实现Web服务大粒度的重用和高效的开发。

关键词 组合 web 服务, 服务发现, 交互模式, 组合模式, 模式匹配

Composite Web Service Discovery based on Pattern Matching

Jingjing Liu¹, Yijian Wu², Wenyun Zhao¹

¹ Department of Computer Science and Engineering, Fudan University, 200433, Shanghai

² School of Software, Fudan University, 200433, Shanghai, China

Abstract Current web service discovery methods mostly based on keywords or classified information of a single web service. This paper proposes a composite web service discovery method which is based on pattern matching. The satisfied service is discovered by checking both interaction pattern and composite pattern, which facilitates user involves service composition and realized the reuse of web service in larger granularity.

Keywords composite web service, service discovery, interaction pattern, composition pattern, pattern matching

1. 引言

现有的Web服务的发现一般是基于关键字和语义的分类信息。虽说本体有非常强大的描述和推理能力, 但现有的研究在语义Web服务发现中只用了它的分类树的功能, 对单个Web的发现比较有效, 却缺少对组合Web服务的组合行为方式的支持。

这篇文章从 Web 服务模式匹配的角度研究 Web 服务的发现和组合, 通过对业务流程行为的匹配实现业务级的服务发现。当用户要获得一个组合 Web 服务时, 他只需要知道它的部分信息和部分流程, 模式匹配算法会依次检查交互模式匹配性和组合模式匹配性, 把符合用户要求的匹配程度的服务返回给用户。用户可以直接选用已有的组合服务, 或者对已有的服务加以修改, 获得自己需要的服务。新的服务发布后也将存入服务注册信息库中, 作为先验知识供以后检索和重用。

2. Web 服务的模式

Web 服务的模式分为交互模式和组合模式两类。交互模式强调参与者的行为对 Web 服务状态的影响, 组合模式的关注的是对 Web 服务结构的组织或对业务流路径的建模。

定义2.1 Web服务的交互模式可以用状态机表示。

一个Web服务的状态机可以表示为 $AUTO = (M, T, q, F, \delta)$ 。其中, $M = M^{in} \cup M^{out}$, M^{in} 是输入消息的集合, M^{out} 是输出消息的集合, T 是服务的有限的状态集, $q \in T$ 是服务的初始状态, $F \in T$ 是服务的最终状态的集合。 δ 是状态变迁关系。

Web 服务的组合模式是我们进行 Web 服务组合时用到的基本的组合单位。类似 workflow 模式, 几种常用的 Web 服务模式有顺序, 循环, 并行, 选择。

定义 2.2 顺序模式: 表示两个服务的执行有先后的关系。一个服务完成后用消息通知下一个服务的执行。可以表示为 $Seq(A_1, \dots, A_n)$, 其中 A_1, \dots, A_n 是两个先后执行的Web服务。

定义 2.3 循环模式: 表示一个服务被多次调用。表示为 $Loop(A_1)$ 或 $Loop(A_1)^c$, c 是结束循环的条件, 如果不指定, 就无限循环下去。

定义 2.4 并行模式: 表示多个服务并发执行。分支可以完全覆盖或部分覆盖。表示为 $And(A_1, \dots, A_n)$ 或 $And(A_1, \dots, A_n)^{m, k}$ 。其中 A_1, \dots, A_n 是需要并行执行的服务, m 表示需要执行的分支的个数, k 表示在收到多少个消息后才能进入下一个服务。两者缺省值都为 n 。

定义 2.5 选择模式：表示分支中多个服务中只有一个会执行。表示为 $Xor(A_1, \dots, A_n)$ ，其中 A_1, \dots, A_n 表示供选择执行的服务。

定义 2.6 一个组合Web服务(CS)可以递归的定义为这样一个元组： $CS = (INFO, SSET, STRU, AUTO)$ ，其中INFO包含描述这个服务的基本信息，SSET表示组成这个服务的子服务的集合，对图 1(a)的服务来说 $SSET = \{A, B, C, D, E, F\}$ 。STRU可以描述服务内部的组织，可以用类似广义表的表示法。例如 图 1(a) 的服务 可以表示为： $Seq(Loop(A)^c, And(Seq(B, C, D), Xor(E, F)))$ 。AUTO是一个状态机如定义 2.1。

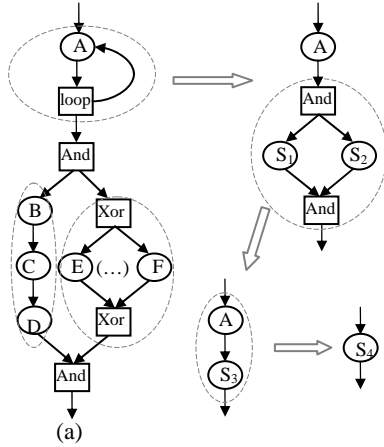


图 1. 组合 Web 服务应用模式的聚合

每个组合服务包含一个主模式，即其最外层的模式。STRU也可以递归的定义为 $(PAT, SSTRU)$ ，其中PAT是它的主模式，SSTRU是组成主模式的其他部分的结构的集合。如图 1(a)的服务的主模式为 Seq 。SSTRU = $\{CS_1, CS_2\}$ ，其中 CS_1 代表 $Loop(A)^c$ ， CS_2 代表 $And(Seq(B, C, D), Xor(E, F))$ 。

组合 Web 服务可以看作一个组合构件，可以从一个组合服务出发递归的把它的基本模式合并最终聚合成一个单独的服务。（图 1）

3. 模式匹配算法

3.1 交互模式匹配

交互模式匹配的任务是检查两个服务是否具有相同的交互行为（称为交互可替换）。例如两个服务 $Seq(A, B, C)$ 和 $Seq(A, D, C)$ ，其中B和D只与A、C有交互行为，则B和D是交互可替换的。

判断两个服务A和B是否是交互可替换的，

需要对与它们发生交互行为的每一个服务检查与另一个服务的可交互性。可以先通过搜索与A的输入输出消息有交集的服务的集合，再进一步判断它们与A、B的可交互性。两个服务的可交互性检查需要对它们之间每一对消息进行核对，检查是否存在不匹配的情况。

3.2 组合模式匹配

组合模式匹配是用服务组合的可能的模糊流程与已有的可用的Web服务进行匹配，如果匹配程度满足要求，已有的Web服务就成为一个建议。称用户的模糊流程为源模版，已有的Web服务为目标模版。如用户知道一个组合服务应该包括A、B、E，并且B、E是并行执行，A在它们之前执行，它的源模版可以表述为： $Seq(A, And(B, E))$ 。可以用它和目标模版 $Seq(Loop(A)^c, And(Seq(B, C, D), Xor(E, F)))$ （图 1(a)表示的组合Web服务）进行匹配。

组合模式匹配根据其匹配程度由强到弱分为完全匹配、包含匹配、替换匹配、替换包含匹配。

完全匹配是程度最高的匹配，源模版和目标模版完全相同时才满足完全匹配。这时已有的 Web 服务能力完全满足用户预定的要求。

包含匹配可以看作是检查源模版是否是目标模版的一个子图。如果源模版 $CS_1 = (INFO_1, SSET_1, STRU_1, AUTO_1)$ ，目标模版 $CS_2 = (INFO_2, SSET_2, STRU_2, AUTO_2)$ 满足包含匹配，当且仅当(1) $SSET_1 \subseteq SSET_2$ ，(2)对于源结构 $STRU_1 = (PAT_1, SSTRU_1)$ ，目标结构 $STRU_2 = (PAT_2, SSTRU_2)$ ，满足 $PAT_1 = PAT_2$ ，并且对任意一个 $SUBCS_1 \in SSTRU_1$ ，存在一个 $SUBCS_2 \in SSTRU_2$ ，使得 $SUBCS_1$ 是 $SUBCS_2$ 的一个子图。子图是这样一种关系：若一个单独的Web服务S是 CS_1 的一个子图满足 $S \in CS_1 \rightarrow SSET_1$ 。如 $Seq(A, And(B, E))$ 就是 $Seq(Loop(A)^c, And(Seq(B, C, D), Xor(E, F)))$ 的一个包含匹配。

替换匹配是交互能力相同的匹配，目标模版可以看作是源模版的某些服务被满足交互可替换的服务替换得到的。对于源模版 $CS_1 = (INFO_1, SSET_1, STRU_1, AUTO_1)$ ，目标模版 $CS_2 = (INFO_2, SSET_2, STRU_2, AUTO_2)$ ，它们满足替换匹配，当且仅当(1) CS_1 和 CS_2 作为一个服务是交互可替换的，(2)它们中任一个（或两者）不包含子模式或者

$CS_1 \rightarrow STRU \rightarrow PAT = CS_2 \rightarrow STRU \rightarrow PAT$ 且对任意一个 $SUBCS_1 \in CS_1 \rightarrow STRU \rightarrow SSTRU$, 存在一个 $SUBCS_2 \in CS_2 \rightarrow STRU \rightarrow SSTRU$, 使得 $SUBCS_1$ 和 $SUBCS_2$ 满足替换匹配。如源模版 $Seq(A, And(M, N), C)$, 目标模版 $Seq(A, D, C)$, 如果 $And(M, N)$ 和 D 是交互可替换的, 则满足替换匹配。

替换匹配可以和包含匹配相结合, 把满足替换匹配的服务当成目标模版(从可替换服务列表中选择替换), 用源模版和它进行包含匹配。这种匹配的程度比包含匹配更弱一些, 称为替换包含匹配。

如果源模版和目标模版不满足上述任何一种匹配, 则称它们不匹配。

判断源模版和目标模版满足哪种匹配, 需要根据上述每种匹配的定义, 依次判断完全匹配、包含匹配、替换匹配和替换包含匹配是否满足, 也可以只检查某种匹配。组合模式匹配算法对完全匹配的检查需要 $O(1)$ 的时间, 对包含匹配的检查为包含服务数大小的乘积。所有满足交互可替换的服务都会存到一个可交互匹配对应表中, 因此替换匹配的复杂度取决于这个表的大小和服务数。替换包含匹配的复杂度取决于替换匹配的结果集大小和服务数。

4. 应用模式匹配的服务发现

模式匹配的任务是把用户根据自己的知识编制的业务流程与已有的组合服务进行匹配, 找到关于已有的 Web 服务的建议, 用户可以直接采用已有的 Web 服务或加以修改构造自己需要的组合 Web 服务。

应用模式匹配的服务发现的过程如下:

(1) 接受用户输入的待匹配信息, 可能是包含具体 Web 服务的模糊模版(源模版), 则进行步骤(3), 也可能只是包含关于服务描述的关键字(或包含一部分服务的语义信息)的模糊模版, 则进行步骤(2)。

(2) 按照服务描述的关键字查找具体的服务(也可以利用语义研究的成果进行查找, 这个不是本文考虑的内容), 通过选择策略和用户自主选择决定包含具体 Web 服务的模糊模版(源模版)。

(3) 首先查找满足交互匹配的服务, 并通过关联可交互匹配对应表返回一个服务的集合 IS 。

(4) 初始化结果集 RS 。

(5) 对 IS 中的服务, 根据需要依次检查是否满足完全匹配, 替换匹配, 包含匹配和替换包含匹配, 也可以检查一部分匹配。如果满足, 则加入结果集 RS , 并从 IS 中删除。

(6) 返回集合 RS , 包括所有可能的建议。用户接受其中一个返回的服务或修改一个服务获得所需要的新的服务。并把新服务加入服务信息库中, 更新相应的可交互匹配对应表, 方便以后重用。

5. 结论

本文从模式匹配的角度提出了一套组合 Web 服务发现的方法。文章定义了服务的交互模式和组合模式, 对于组合模式, 还按照它们的匹配程度分为四个等级, 给出了如何利用模式匹配发现 Web 服务的解决方案。我们的方法有以下优点: (1) 用户可以从业务流程出发查找组合 Web 服务, 降低了用户参与服务组合的难度, 在业务级面向服务应用的构造上效率更高, 也有利于服务大粒度的重用。(2) 用户不必知道一个业务流程的全部细节就能够从先验数据中得到提示, 使得完善的组合 Web 服务可以不必是领域专家的作品。(3) 模式匹配算法可以与以研究独立 Web 服务为出发点的各种发现方法相结合, 如 Web 服务的语义发现。(4) 模式匹配算法不仅可以应用到服务发现上, 还可以应用到服务的验证和正确性分析上, 也可以给组合服务的动态绑定带来方便, 这也是我们以后的研究工作。

参考文献:

- [1] M. Voorhoeve. Compositional modeling and verification of workflow processed. In: W. Aalst, J. Desel, A. Oberweis eds. Business Process Management: Models, Techniques, and Empirical Studies. Berlin: Springer-verlag, 2000.
- [2] Jen-Yao Chung, Miller, S., Qianhui Althea Liang, Yang Ouyang. Service Pattern Discovery of Web Service Mining in Web Service Registry-Repository. In: Proceedings of IEEE International Conference on e-Business Engineering, 2006. pp.286-293.
- [3] 胡海涛, 李刚, 韩燕波. 一种面向业务用户的大粒度服务组合方法, 计算机学报, vol. 28, No.4, Apr. 2005.

姓名(中文)	刘晶晶	姓名(英文)	Jingjing Liu	性 别	女	出生年月	1983. 9
职称(职务)	学生	电 话	13611778420	Email	052021164@fudan. edu. cn		
手机	13611778420	传真		通 信 地 址	上海武川路 78 弄 65 号楼 701 室, 200433		
研 究	中 文	软件工程, web服务技术					
方 向	英 文	Software engineer, web services					
基 金	中 文	国家自然科学基金(60473061、60473062)、国家863计划(2005AA113120)资助					
资 助	英 文	National Natural Science Foundation of China under Contracts 60473061 and 60473062, National 863 Program under Contract 2005AA113120					