

Web 服务的 Pi 演算描述^{*})

郭小群 郝克刚

(西北大学计算机科学系 西安 710069)

摘要 如何从已有的 Web 服务构造新的增值性的正确的 Web 服务,已成为研究 Web 服务的群体关注的一个焦点。因此,为了得到可靠的、高质量的服务,必须找到一种建模方法和工具,在 Web 服务投入使用之前,对其进行分析和验证。本文主要从过程代数的角度出发,用 Pi 演算来描述 Web 服务以及 Web 服务的组合。

关键词 Pi 演算, Web 服务, Web 服务组合

Pi Calcula-Based Model for Web Services

GUO Xiao-Qun HAO Ke-Gang

(Department of Computer Science, Northwest University, Xi'an 710069)

Abstract The problem that researchers of Web services field focuses on is how to integrate existed Web services to create new add-valued and correct Web service. Therefore, there is a need for modeling techniques and tools for reliable Web service. In this paper, based on thinking that Web services are processes, we proposed using Pi to model Web service and Web service composition.

Keywords Pi calculas, Web service, Web service composition

1 引言

目前的 Internet 网已经变成一巨大的商业平台,许多组织和个人通过这个平台相互通信,进行商业活动或增值性服务。由于 Web 服务为这些不同组织或个人的不同应用的互操作提供了可行性,所以 Web 服务受到越来越多的关注。Web 服务主要基于下面这些技术标准: UDDI(统一描述、发现和集成)、WSDL(Web 服务描述语言)、SOAP(简单对象访问协议)。目前 Web 服务面临的一个新的挑战是:这些技术标准并不能实现复杂的服务组合,只能提供有限的组合^[1]。为了解决这一问题,人们主要从这两个方面研究 Web 服务组合^[2]:一个是工业化的方法,主要用 WSDL 和 BPEL4WS 描述服务及服务组合,这种方法是基于流程的,其理论基础是 Petri 网和进程代数;另一个是语义 Web 方法,主要用 RDF/DAML-S 和 Golog/Planning 描述服务及服务组合,这种方法是基于人工智能的,其理论基础是描述逻辑和时序逻辑。这两种理论的共同点是对流程进行形式化描述,以此实现流程再建模阶段的分析和验证。本文从基于流程的观点出发,以 Pi 演算作为 Web 服务的建模语言,表达 Web 服务的语义并进一步描述 Web 服务组合。我们之所以选择 Pi 演算作为 Web 服务的描述语言,主要在于 Pi 演算和 Web 服务中的关键概念非常类似。

2 Pi 演算简介

Pi 演算^[3,4]起源于 1991 年,是由图灵奖得主 Robin Milner 参照物理学大统一理论提出的。它是一种描述和分析并发系统的计算模型,用动态演化结构表示过程间的间歇性相互作用,名称是基本的概念、值、变量和管道通过名称引用。图 1 是 Pi 演算的一种图形表示,其中 P, Q 表示过程, $\bar{b}$ 表示端

口(一个过程可有若干端口), b 表示动作, $\rightarrow$ 表示管道。在图中,过程 P 通过端口 $\bar{b}$ 沿管道发送消息,过程 Q 通过管道接受消息。下面是 Pi 演算的代数定义。



图 1 Pi 演算的图形表示

定义 1 设 N 是一个表示名称的可数无限集合,用 $a, b, x, y, \dots$ 表示 N 中的元素; A 是一个代理(或过程)的集合,用 P, Q, R 表示 A 中的元素,则 Pi 演算的语法可定义如下:

$P ::= 0 \mid \bar{x}y. P \mid x(y). P \mid \tau. P \mid (y)P \mid [x=y]P \mid P \mid Q \mid P + Q \mid A(x_1, x_2, \dots, x_n) \mid !P$

这里, $x(y). P$ 和 $(y)P$ 中的 y 被称为约束名称, P 是它们的辖域。在 P 中出现的非约束名称称为自由名称, P 中的自由名称的集合表示为 $fn(P)$, 约束名称的集合表示为 $bn(P)$, P 中名称的集合用 $n(P)$ 表示。操作符的含义如表 1 所示。

表 1 操作符的含义

操作符	符号	含义
前缀	$y. P$	顺序执行
动作	$\bar{x}y, x(y)$	通信
求和	$P + Q$	选择
复制	$!P$	循环
组合	$P \mid Q$	并发
约束	$(y)P$	封装

定义 2 Pi 演算的操作语义可用标记转移系统(LTS)表示。LTS 是一个 3 元组 $(P, A, \rightarrow)$, 这里 P 是过程(或代理)的

^{*}) 本文得到西安市科技攻关基金项目(GG200220)及西北大学科研基金项目(02NW11)的支持。郭小群 博士生, 主要研究领域为软件工程; 郝克刚 博士生导师, 主要研究领域为软件工程。

集合, A 是动作的集合, $\rightarrow \subseteq F \times A \times P$ 。根据 LTS, P_i 演算的操作语义如表 2, 其中 $\{y/z\}$ 表示用 y 替换 z 。

3 Web 服务

3.1 P_i 演算和 Web 服务的比较

Web 服务至今尚未有统一的定义。广义上, 可以把它理解为通过 Web 进行数据及功能共享的方式; 狭义上, 可以把它理解为一种通过 Web 及标准接口进行调用的软件, 类似 CORBA 或 EJB。W3C 最近把 Web 服务定义为一种软件应用^[5], 其应用领域包括电子商务、电信、网格计算、电子政务等。Web 服务的主要概念见表 3^[6], 其中类型描述的是被传递的信息的数据类型。信息是被传递的数据的抽象定义, 它由若干逻辑部分组成, 每一个逻辑部分和一种数据类型相关联。操作是服务支持的动作的抽象描述。端口类型是一组抽象操作, 每一个操作会涉及到一个输入信息和一个输出信息。绑定是指一个特定的端口类型定义的信息和操作所使用的具体的协议和数据格式的说明。服务是一系列端口的集合。

表 2 P_i 演算的操作语义

结构单元	$P' \equiv P, P \xrightarrow{a} Q, Q \equiv Q'$ $P' \xrightarrow{a} Q'$	前缀	$\frac{}{a.P \xrightarrow{a} P} \quad a ::= \tau \bar{x}y x(y)$
求和	$\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'}$	并发	$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q'}{P Q \xrightarrow{a} P' Q} \quad \text{bn}(a) \cap \text{fn}(a) = \emptyset$
匹配	$\frac{P \xrightarrow{a} P'}{[x=y]P \xrightarrow{a} P'}$	约束	$\frac{P \xrightarrow{a} P'}{(y)P \xrightarrow{a} (y)P'} \quad y \notin n(a)$
通信	$\frac{P \xrightarrow{\bar{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P Q \xrightarrow{\tau} P' Q' \{y/z\}}$	复制	$\frac{}{! a.P \xrightarrow{a} P ! a.P}$

表 3 P_i 演算和 Web 服务关键概念的对照

Web 服务	P_i 演算	Web 服务	P_i 演算
类型	类型	绑定	交互作用
消息	消息	端口	端口
操作	动作	服务	过程
端口类型	类型		

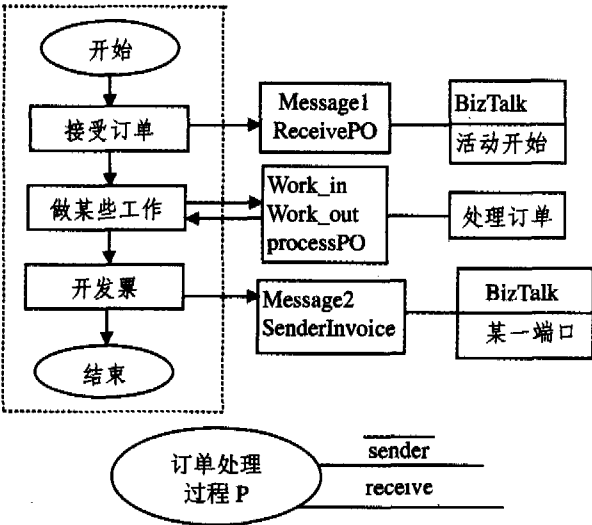


图 2 订单处理流程

图 2 是一个订单处理的表示, 上面的图用 Web 服务表示, 下面的图用 P_i 演算表示。在上面的图中, 虚线框内部分

表示一个服务, ReceivePO 表示端口, $\rightarrow$ 表示操作, 描述活动开始的工具 BizTalk 表示绑定, Message1 表示消息。

3.2 Web 服务组合的描述

Web 服务组合指的是对已存在的单个 Web 服务进行集成, 以创造新的商业流程。单个的 Web 服务既可以是组合的, 也可是原子的。根据前面的表格, Web 服务中的服务和 P_i 演算的过程相对应, 则 Web 服务组合可用 P_i 演算的语法定义如下:

$$S ::= 0 | S_1.S_2 | S_1 + S_2 | S_1.S_2.S_1 | ! S | (\bar{x}y.S_1 | x(y).S_2) + (\bar{x}y.S_2 | x(y).S_1) | (S_1 | S_2).S_3 | \sum_{i=1}^n x_i(y). \bar{x}_i y.S_i | S_1(S_2/a)$$

- 0 表示空服务。
- S 表示一个服务常量, 可被看作原子服务或基本服务。
- $S_1.S_2$ 表示组合服务, 这个组合服务先执行服务 S_1 再执行服务 S_2 。
- $S_1 + S_2$ 表示组合服务, 这个组合服务执行服务 S_1 或服务 S_2 , 只执行一个。
- $S_1.S_2.S_1$ 表示服务 S_1, S_2 以任意顺序执行。
- $(\bar{x}y.S_1 | x(y).S_2) + (\bar{x}y.S_2 | x(y).S_1)$ 表示多次执行服务 S 。
- $(\bar{x}y.S_1 | x(y).S_2) + (\bar{x}y.S_2 | x(y).S_1)$ 表示服务 S_1, S_2 独立执行并有可能进行通信。
- $(S_1 | S_2).S_3$ 表示再激活服务 S_3 之前, 服务 S_1, S_2 中有一个被执行, 这里 S_1, S_2 并发执行但不通信。
- $\sum_{i=1}^n x_i(y). \bar{x}_i y.S_i$ 表示从 n 个服务中动态地选择一个服务执行。它的行为如下: 首先发出服务请求, 每个服务从自己的端口接受请求, 然后给出响应, 发出请求者根据响应的消息做出最佳选择。
- $S_1(S_2/a)$ 表示组合服务的行为和服务 S_1 相同, 除了 S_1 中的标记 a 被一个非空服务 S_2 代替。

结束语 Web 服务投入使用时, 如果包含错误, 则用户会感到失望, 甚至抱怨, 所以在 Web 服务投入使用之前必须对其进行分析。Web 服务分析主要包括死锁和活锁的检查。使用 P_i 演算描述 Web 服务可以有效地检查死锁和活锁问题, 也很容易验证两个 Web 服务行为的等价性。 P_i 演算作为一种过程代数, 具有许多代数性质。这些性质同样适用于 Web 服务组合。通过对 Web 服务操作的度量, 例如需要的成本、花费的时间等, 这些代数性质可用来转换和优化 Web 服务组合。但是目前的问题是缺乏一种自动化工具来支持用 P_i 演算对 Web 服务进行分析和验证。

参考文献

1. Hamadi R, Benatallah B. A Petri Net-based Model for Web Services Composition. Proc. of the Fourteenth Australasian Database Conference on Database Technologies, 2003
2. Biplav S, Jana K. Web Services Composition-Current Solution and Open Problems. Online <http://www.zurich.ibm.com/pdf/ebizz/icaps-ws.pdf>
3. Milner R, Parrow J, Walker D J. A calculus of Mobile Process, Part I. Report ECS-LFCS-89-85, Laboratory for Foundation of Computer Science, Computer Science Department, Edinburgh University, 1989
4. Milner R, Parrow J, Walker D J. A calculus of Mobile Process, Part II. Report ECS-LFCS-89-85, Laboratory for Foundation of Computer Science, Computer Science Department, Edinburgh University, 1989
5. 葛声, 马殿富, 胡春明, 等. 基于 Web 服务的网络软件运行平台研究与实现. 北京航空航天大学学报, 2003, 10
6. π -Calculus for Web Services. online <http://www.gcc2002.ict.ac.cn/reports/27B/5B5.ppt>