

基于 CPi-calculus 的网格服务行为研究^{*}

周 静^{1,2} 曾国荪^{1,2}

(同济大学计算机科学与技术系 上海 201804)¹

(国家高性能计算机工程技术中心同济分中心 上海 201804)²

摘 要 当前基于网格服务组合的编排方案(choreography proposals)只给出服务描述层面上的定义,没有提供任何推导方法来保证基于此编排方案的服务的一致性。如何验证组合服务行为的正确性和有效性是当前亟需解决的一个研究课题。形式化方法提供了行之有效的建模、推导和验证机制。本文给出网格服务组合基调,基于此基调分析了网格组合服务的动态交互行为特性,提出基于 CPi-calculus(Conditional Pi-calculus)的网格组合服务的行为交互模型(IABM),并给出组合服务交互行为的形式化描述,最后进行交互行为的模拟验证。该网格服务行为交互模型为今后网格服务的组合优化工作提供了代数推导依据。

关键词 网格计算,服务组合,行为交互模型,CPi-calculus

A Study of Grid Service Behavior Using CPi-calculus

ZHOU Jing^{1,2} ZENG Guo-Sun^{1,2}

(Department of Computer Science and Technology, Tongji University, Shanghai 201804)¹

(Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 201804)²

Abstract Current grid service choreography proposals remain at the descriptive level, without providing any kind of reasoning mechanism for checking the compatibility of grid services. The research work on how to formalize and verify the behavior of composite grid services is therefore imperative. Formal approaches offer an effective and unambiguous way to model, represent and verify the grid service composition. A novel grid service composition signature is proposed in this paper. By analyzing the interactive behavior of the composite grid service based on the signature, we present interactive behavior model (IABM) for composite grid services, specify the interactive behavior using the CPi-calculus (Conditional Pi-calculus) and formally simulate the interactive behavior finally. The interactive behavior model for composite grid service is algebraic foundation to be used afterwards in automatic and dynamic composition.

Keywords Grid computing, Service composition, Interactive behavior model, CPi-calculus

1 引言

网格服务是一个开放的、动态交互并发的过程,它通过无缝组合广域分布的资源服务,使得研究工作者可以更好地合作解决科学问题^[1]。当前还没有有效的组合服务语言,提供行为描述同时给出行为推理机制和服务组合有效性的检查机制。形式化描述服务从而保证服务组合的有效性和一致性是当前研究的热点问题^[5]。现有的对服务组合形式化的研究工作^[9~13]没有给出一般化的代数模型、抽象的表征组合服务行为语义特征,无法指导服务的动态组合、匹配和消息的传递,不利于网格服务自动组合的实现。

本文试图将网格服务组合的层次结构和组合服务的动态交互特征,纳入统一的概念体系。由于服务交互行为受到内外环境的条件约束,我们对传统的 Pi-calculus 进行语义扩展,提出了基于条件的 Pi-calculus, CPi-calculus (Conditional Pi-calculus)。进而给出网格组合服务基调和组合服务的行为交互模型,使网格组合服务的结构和行为达到统一。最后使用 CPi-calculus 对组合服务的实现进行形式化推导,为网格服务的自动组合和优化提供了理论依据。

本文第 2 节给出 CPi-calculus 的操作语义。第 3 节详细介绍了我们提出的组合服务的基调和网格组合服务的行为交互模型,并给出基于 CPi-calculus 的形式化描述。第 4 节给出基于网格组合服务行为交互模型的推导验证的模拟。第 5 节介绍相关工作。最后是全文的总结和展望。

2 CPi-calculus (Conditional Pi-calculus)

进程代数因其简单、形式化的描述和推理特性,被认为是描述服务组合交互行为的最有效方法^[6]。Pi 演算^[14,15]是 Robin Milner 提出的以移动通信为研究重点的并发理论。传统上 Pi-calculus 用来研究分布式并发通信问题,模拟分析实际中的并发通信。由于网格服务组合不能脱离服务所处的环境,也不能忽视内部组成服务的行为变化,不能使服务失去组合的独立性和封装性,传统的 Pi-calculus 显然不足以表达服务组合的交互行为。为了明确地给出交互的进程所属的服务,以及控制条件对组合服务交互行为的影响,我们对 Pi-calculus 进行语义扩充,给出 CPi-calculus (Conditional Pi-calculus)的操作语义。

定义 1 $N = \{x, y, z, \dots\}$ 表示名字集合,用来表示服务

^{*}国家自然科学基金资助项目(60173026)、上海科委重大项目(03DZ15029)、上海高校网络技术 E-研究院资助(200301-1)。周 静 博士生,主要研究领域:并发计算、服务计算;曾国荪 博士,教授,博导,主要研究领域为异构计算、计算网格。

组合中的交互并发传递的消息。

服务之间通过发送消息进行交互,存在多种交互形式。有的交互在服务组合的构造过程中已经静态绑定,有的则是在服务调用过程中动态绑定。服务之间的交互大致可以分为同步消息传递和异步消息传递,交互可以一对一进行,也可以一对多、多对一以及多对多。

定义 2 令 S 表示服务进程 (Service Process), $C = \{P, Q, \dots\}$ 为所有交互的服务进程表达式的集合,交互进程表达式的递归定义如下:

- 1) 0——终止 SP 进程;
- 2) $x(y; N). P$ ——接受消息的 SP 进程;
- 3) $[\theta; Boolean] \bar{x}(y; N). P$ ——发送消息的 SP 进程;
- 4) $[\theta; Boolean] \tau. P$ ——内部 SP 进程;
- 5) $P+Q$ ——选择;
- 6) $P|Q$ ——复合;
- 7) $S = Def^P$ ——递归

SP 为交互服务进程的缩写。0 表示什么也不做的 SP 进程。 $x(y). P$ 表示从信道 x 接收消息 y , 然后执行 P 进程, 是无条件地接受消息。 $[\theta; Boolean] \bar{x}y. P$ 表示输出动作受局部控制条件 $[\theta; Boolean]$ 的控制, 条件满足时, 通信到 x , 输出消息 y , 然后执行 P 进程。 $[\theta; Boolean] \tau. P$ 中, τ 用于表示外部不可见的内部动作, $[\theta; Boolean] \tau. P$ 表示如果条件为真, 做完哑动作 τ 接着执行 P 。交互进程 $P+Q$ 表示要么执行 P , 要么执行 Q 。 $P|Q$ 表示进程 P 和 Q , 要么交错执行, 要么并发执行, 也就是异步和同步的概念。 $S = Def^P$ 定义了服务交互进程的递归表达式。

定义 3 Service Process (SP): $Id = \{i, j, \dots\}$ 表示服务标识集合, $C = \{P, Q, \dots\}$ 为所有交互的服务进程 SP 的集合, 我们给出服务进程的定义: $SP_i = P@i$ 。其中 $P@i$ 表示服务 i 的交互进程 P 。

定义 4 CPi-calculus 的操作语义: 假定 P, Q 为 SP, $x(y)$ 为输入操作, $\bar{x}(y)$ 为输出操作, a, a, b 为交互操作符, $\theta, \psi \in Boolean$ 为操作执行的条件, 则扩充后的 Pi-演算的操作语义规则“ $\xrightarrow{(\text{约束条件, 交互操作})}$ ”:

Sent rules:

$$\begin{aligned} & [\theta] \bar{x}(y). p@i \xrightarrow{(\theta, \bar{x}(y))} p@i | \bar{x}(y) @y_i \\ & [\theta] \bar{x}(y_1, y_2). p@i \xrightarrow{(\theta, \bar{x}(y_1))} p@i | \bar{x}(y_1) @y_{i1} | \bar{x}(y_2) @y_{i2} \\ & [\theta] \bar{x}(y_1, y_2). p@i \xrightarrow{(\theta, \bar{x}(y_1) | \bar{x}(y_2))} p@i | \bar{x}(y_1) @y_{i1} | \bar{x}(y_2) @y_{i2} \end{aligned}$$

Receive rules:

$$x(y). P@i \xrightarrow{(true, x(y|a))} P|a@i$$

Com rules:

$$\begin{aligned} & [\theta] a. P@i \xrightarrow{(\theta, a)} P@i \\ & \frac{P@i \xrightarrow{(\theta, a)} P'@i}{P@i + Q \xrightarrow{(\theta, a)} P'@i} \\ & \frac{P@i \xrightarrow{(\theta, a)} P'@i}{Q + P@i \xrightarrow{(\theta, a)} P'@i} \\ & \frac{P@i \xrightarrow{(\theta, a)} P'@i}{P@i | Q \xrightarrow{(\theta, a)} P'@i | Q} \end{aligned}$$

$$\frac{P@i \xrightarrow{(\theta, a)} P'@i}{Q | P@i \xrightarrow{(\theta, a)} Q | P'@i}$$

其中, 条件 θ, ψ 控制服务交互进程的输行动作。当条件为真的时候, $\bar{x}(y). P$ 执行一个异步消息发送动作 $\bar{x}(y)$, 并演变为两个并发进程 $P@i$ 和 $\bar{x}(y) @y_i$ 。 $P@i$ 对应于 $\bar{x}(y)$ 。 $P@i$ 的后继动作, $\bar{x}(y) @y_i$ 对应于一个异步消息发送和传递的过程, 表示异步消息 y 已到达可能的目的集合, 其中 \bar{x} 为消息 y 的目标分量。

3 网络组服务的行为交互模型 IABM (Interactive Behavior Model)

OGSA 构架, 可以基于简单的基本的服务, 构造出具有层次结构的高级别的服务。要较好地表达和验证网络组合服务的交互行为, 必须将其层次构架和行为表征统一化。因此本文给出网络组合服务的基调, 该基调给出了组合服务的基本属性和构造特征; 基于此基调, 我们给出网络组合服务的行为交互模型 (IABM), 使用 CPi-calculus 形式化表示网络组合服务的交互行为。该行为交互模型较好地刻画了组合服务组合的封装性、服务调用的事件驱动性、动态绑定等行为交互特征, 为服务组合正确性和有效性的形式化验证提供了理论基础。

3.1 网络服务基调

网络服务的组合进行形式化抽象能够准确地刻画网络服务的组合和交互并发过程。网络服务组合可以看成是由底层的服务向上组合构造而成, 上层服务是下层服务的组合服务, 下层服务称为上层服务的组成服务, 并且上层服务是下层服务的聚合和封装。作为网络服务组合的基本单元, 服务具有唯一标识, 服务的产生和消失是通过交互并发来实现的。服务的交互并发可以抽象为服务的交互行为推理序列, 称为推理映射 (Reduction Map), 即通过服务的交互行为来完成组合服务的具体功能。我们给出组合服务交互行为的定义:

定义 5 称五元组 $\Sigma_S = (I, A, IA, AB, \varphi, op)$ 为组合服务 S 的基调, 其中:

- $I = \{i, j, \dots\}$ 为服务 S 的组成服务标识集;
 - $A = \bigcup_{i \in I} A_i = \{a, b, \dots\}$ 为各组成服务操作集合的并;
 - $IA \subseteq I \times A$ 为服务执行操作的动作集合
- $$\forall i \in I, \forall a \in A, (i, a) \in IA \text{ 当且仅当 } a \in A_i \quad (IA 1)$$
- $AB \subseteq I \times I$ 为 I 上的一个抽象聚合关系;
 - $\varphi: IA \rightarrow IA^*$ 为 IA 到 IA^* 的一个推理映射, 满足:
- $$\forall (i, a) \in IA, \varphi(i, a) = (i_1, a_1); (i_2, a_2); \dots; (i_n, a_n) \quad (RM 1)$$

其中, i_k 表示组合服务 i 的组成服务, $(i_k, a_k) \in IA, i \leq k \leq n, n = \text{length}(\varphi(i, a))$ 表示动作序列的长度; $\varphi(i, a)$ 是组合服务 i 与其各个组成服务 $i_k (i \leq k \leq n)$ 之间的推理映射的集合。

• $op[[srv1; \rightarrow srv2]]SC$ 为组合服务中各组成服务之间的 E-CSPE 约束关系集合^[16]。

其中, S 是组合服务; $srv1$ 和 $srv2$ 是组合服务中前后相关的两个组成服务动作, 它们不需要紧密相连; C 是一个状态谓词, 它决定 $srv1$ 和 $srv2$ 之间的依赖关系是否存在。 op 有 3 种取值^[24]:

1) $a[[srv1; \rightarrow srv2]]SC$: 在服务 S 的调用过程中, 若条件 C 成立, 则 $srv2$ 发生于 $srv1$ 之后总有效。

2) $m[[srv1; \rightarrow srv2]]SC$: 在服务 S 的调用过程中, 若条

件 C 成立, 则 $srv1$ 发生以后 $srv2$ 必发生。

3) $\sim[[\text{srv1}; \rightarrow \text{srv2}]]SC$: 在服务 S 的调用过程中, 若条件 C 成立, 则 $srv2$ 发生于 $srv1$ 之后总无效。

任意给定服务 S 的一个可行调用 s 和针对服务 S 的一个约束 r , s 与 r 之间可定义 3 种关系:

- 1) $CompatibleClass(r, S)$: 所有覆盖 r 的可行服务调用集合;
- 2) $AgainstClass(r, S)$: 所有违反 r 的可行服务调用集合;
- 3) $NonsenseClass(r, S)$: 所有与 r 无关的可行服务调用集合。

令 $Feasible(S)$ 为组合服务 S 所有可行服务调用的集合, 显然有

$$CompatibleClass(r, S) \cup AgainstClass(r, S) \cup NonsenseClass(r, S) = Feasible(S)$$

服务 S 的一个约束 r 和调用 s 之间的关系, 可以作为形式化推导验证的依据。例如:

• 若 $\bigcap_{r \in op} NonsenseClass(r, S) = \{\}$, 则针对任一组合服务 S 的调用 s , op 中总存在与 s 相关(覆盖或违反)的约束 r 。认为约束集 op 相对于 S 的实现是充分的。

• 若 $\forall r \in op. AgainstClass(r, S) = \{\}$, 则针对 op 中的任一约束 r , 不存在违反 r 的可行事件序列。可以认为 S 的实现相对于约束集 op 是正确的。

定义 6 AB 为 I 上的一个抽象聚合关系, 满足自反性、对称性、传递性:

$$\forall i \in I, (i, i) \in AB \quad (AB \ 1)$$

$$\forall i, j \in I, \text{若 } (i, j) \in AB, \text{且 } (j, i) \in AB, \text{则 } i = j \quad (AB \ 2)$$

$$\forall i, j, k \in I, \text{若 } (i, j) \in AB, (j, k) \in AB, \text{则 } (i, k) \in AB \quad (AB \ 3)$$

$$\forall i, j, k \in I, \text{若 } (i, k) \in AB, (j, k) \in AB, \text{则 } (i, j) \in AB, \text{或 } (j, i) \in AB \quad (AB \ 4)$$

$$\forall i, j \in I, \exists k \in I, \text{使得 } (k, i) \in AB, \text{且 } (k, j) \in AB \quad (AB \ 5)$$

我们使用 Σ_s 刻画一个网络组合服务的组织结构, 它是由 Σ_s 中的二元关系 IA 和 AB 来确定的, 组成服务间的约束条件由约束 op 来确定。关系 IA 表明属于某个特定服务的特定操作集合, 而关系 AB 使用 $(AB1) \sim (AB5)$ 表明组合服务的抽象聚合集合, 也就是上层组合服务对下层组成服务集的抽象。同时引入服务调用的推理映射, 使得网络服务的结构框架和服务调用机制有机地结合起来。网络组合服务的基调是本文研究网络组合服务并发交互行为的一个统一的抽象概念框架, 将服务于正文。

定理 1 给定服务组合基调 $\Sigma_s = (I, A, IA, AB, \varphi, op)$, 则抽象聚合关系 AB 是一个偏序关系。

若 S 表示所有服务的集合, \leq 表示 S 上的一个偏序关系, 则用 $\text{sup}\{S, \leq\}$ 表示集合 S 在偏序 \leq 下的极大元素集合, 用 $\text{sub}\{S, \leq\}$ 表示集合 S 在偏序 \leq 下的极小元素集合。

定义 7 给定组合服务 S 的基调 Σ_s , 定义组成服务集合 I 在关系 AB 下的集合 $\text{sup}\{i \in I\}$ 和 $\text{sub}\{i \in I\}$ 分别称为根服务集合和叶子服务集合。 $\forall i \in I$

$i \in I\text{-sup}\{i \in I\}$, 称 $\uparrow i = \text{sub}\{j | j \neq i, (j, i) \in AB\}$ 为 i 的(上层)组合服务集合;

$i \in I\text{-sub}\{i \in I\}$, 称 $\downarrow i = \text{sup}\{j | j \neq i, (i, j) \in AB\}$ 为 i 的(下层)组成服务集合。

定理 2 给定组合服务基调 Σ_s , 则:

$$\forall i \in I\text{-sup}\{i \in I\}, \uparrow i \text{ 具有唯一元素};$$

$$\forall i, j \in I\text{-sub}\{i \in I\}, \text{且 } i \neq j, \text{则 } \downarrow i \cap \downarrow j = \emptyset.$$

定义 7 和定理 2 给出了(上层)组合服务和(下层)组合服务的定义和性质: 根服务作为一个组合服务基调的最大(最顶层)服务, 封装了所有的组成服务; 叶子服务作为一个组合服务基调的最小(最底层)服务, 被上层服务封装。

定义 8 给定 $\varphi(i, a)$ 表示 (i, a) 的推理序列, $(\varphi(i, a))_k$ 记为 $\varphi_k(i, a)$, 表示推理序列 $\varphi(i, a)$ 上的第 k 个元素, 即组合服务 i 与其第 k 个组成服务 i_k 之间的推理映射。我们将推理序列 $\varphi(i, a)$ 看作一个字符串 $Str = "s_1 s_2 \dots s_n"$, 推理序列 $\varphi(i, a)$, 即字符串 Str 的首字符 $\text{head}(Str) = s_1$ 。

定义 9 定义 $\partial_I: IA \rightarrow I, \partial_A: IA \rightarrow I$ 为 IA 在 I 和 A 上的映射, 则 $\forall (i, a) \in IA$:

$$\partial_I(i, a) = i, \partial_A(i, a) = a \in A;$$

$$\partial_I(\varphi_k(i, a)) \in \downarrow i, (\partial_I(\varphi_k(i, a)), \partial_A(\varphi_k(i, a))) \in IA$$

(RM 2)

3.2 网络服务行为模型

网络组合服务的交互行为是通过服务组合基调中的二元关系和约束关系来描述的。同时, 由于服务推理映射的引入, 使得服务组合的层次结构和组合内部的协同工作有机的结合起来。在这些工作的基础上, 我们给出网络服务组合的行为交互模型, 并使用 CPi -calculus 进行形式化描述。

由于网络服务是通过组合异构资源提供协同工作的计算环境, 因此组合服务 i 的行为参与要素包括: 1) 组成服务集合 $i_1, i_2, \dots, i_n, \forall i_k, 1 \leq k \leq n$; 2) 组合服务 i 所处的语义环境 $\tau(i)$; 3) 负责与外部交互的控制台 $CT(i)$ 。我们给出网络组合服务的交互模型, 如图 1 所示。

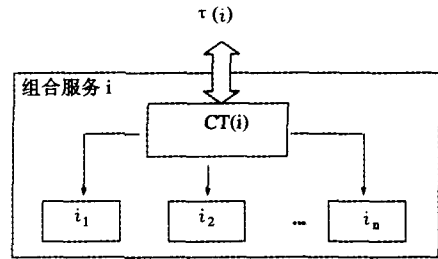


图 1 网络服务组合的行为交互模型

3.2.1 上下文环境(Context)

给定组合服务基调 $\Sigma_s = (I, A, IA, AB, \varphi, op)$, $\forall i \in I$, 集合 $T(i)$ 表示与服务 i 交互作用的外部语义环境, 定义为:

1) i 为非根服务标识, 则 $T(i) = \uparrow i$, 也就是封装服务 i 的组成服务;

2) i 为根服务标识, 则 $T(i) = \text{sup}\{i \in I\} - \{i\}$, 即除 i 以外的其它根服务。

3.2.2 控制台

给定服务组合基调 $\Sigma_s = (I, A, IA, AB, \varphi, op)$, $\forall i \in I$, 控制台 $CT(i)$ 负责组合服务 i 与外部环境的交互, 控制台定义为服务交互进程 $Control@i$, 并且分别考虑组合服务 i 为叶子服务和非叶子服务的情况。

1) 组合服务 i 为非叶子服务, 控制台进程为 $CT(i) = Control@i | \prod_{j \in \downarrow i} CT(j)$

2) 组合服务 i 为叶子服务, 控制台进程为 $CT(i) = Control@i$


```
def unset_auto(self);
    broker.unset_auto()
```

图3 网格服务的非形式化描述和定义

我们给出与金融交易相关的简单网格服务实例。假定上海股民投资者,希望购买美国市场的股票。通常需要将其投资资金从本地银行转到相关证券经纪公司的帐户上。由于是购买美国股票,必须使用美元进行交易,因此在资金转帐前,需要进行相应的货币转换。显然经纪公司方需要提供代理服务 BrokerAccount,而银行方也要提供相应的转帐和转换货币的服务 BankAccount。我们将以上服务进行组合,得到网格服务 CompositeAccount,如图3。

```
acc CompositeAccount // 对组合服务 Composite-
Account 的调用
acc.buy_and_exchange("IBM", 500)
acc.set_bank_account()
acc.transfer(5000, "from bank")
acc.set_auto_transfer()
acc.buy_and_exchange("SUNW", 50)
acc.file_to_parent()
```

图4 网格服务的简单调用实例

我们给出简单的网格服务调用实例,如图4。投资者买入500股IBM的股票和50股SUNW的股票,投资者只需要调用组合服务 CompositeAccount,即可完成操作。我们给出如下组合服务的行为进行形式化推导分析:

```
Control = x(acc). A | Control
//接收组合服务 acc 的调用//
A = [θ]B + [θ]B' //θ 用于检查消息类型、动态绑定
等//
//acc = τ. exchange⟨"MCY", "USD", 500⟩ @ BankAc-
count. [μ]buy⟨"IBM", 500⟩ @ Broker Account. ste-
_bank_account() @ Broker Account. set_auto-
_transfer @ Broker Account.
.exchange⟨"MCY", "USD", 50⟩ @ BankAccount.
[μ]buy⟨"SUNW", 50⟩ @ Broker Account.
file() @ Broker Account | file() @ BankAccount //
B = x̄⟨acc; y1, ρ1(φ1(acc, head(y1))), ρA(φA(acc, head
(y1))); y2, z1⟩. C
// 给出组合服务 acc 与组成服务的推理映射,此时分解
出组合服务序列中的第一个服务 y1 = τ,也就是对组合
服务 acc 所做初始化操作。broker = BrokerAccount(" *
* * * * *"); bank = BankAccount(" * * * * * * *
* * * *")
y1 = τ
y2 = exchange⟨"MCY", "USD", 500⟩ @ BankAccount.
[μ]buy⟨"IBM", 500⟩ @ Broker Account.
set_bank_account() @ Broker Account. set_auto-
_transfer @ Broker Account.
.exchange⟨"MCY", "USD", 50⟩ @ BankAccount.
[μ]buy⟨"SUNW", 50⟩ @ Broker Account.
file() @ Broker Account | file() @ BankAccount
μ 表示条件 bank_account⟨⟩ nil and auto_transfer_p =
= true //
```

```
B' = x̄(y). 0 //不执行不符合条件的消息
C = for(k=1; k ≤ length(φ(i, head(y2))); k++) D; H
//对从 y2 服务组合序列的循环分解,显然 k=7,组合服
务 acc 分解为 7 个组成服务
D = x(z). ([ψ]F + [ψ]F') //接受并检查分解动作的返
回结果
F = [z3 ≠ "decline"]G + [z3 = "decline"]G'
F' = x̄(z). 0
G = x̄⟨acc; z1, ρ1(φk(i, head(y2))), ρA(φk(i, head
(y2))); z2, z3⟩. 0 //φ(i, head(y2))推理序列的第
k 个元素
G' = x̄⟨i; y1, head(y1))⟩, y2, "decline"⟩. 0
H = x(z). P
P = [ζ]x̄⟨acc; y1, head(y1))⟩; y2, z3⟩. 0 + [ζ]x̄(z). 0
//确认推理序列最后一个动作的返回
```

$\rho_1(\varphi_1(\text{acc}, \text{head}(y_2))), \rho_A(\varphi_A(\text{acc}, \text{head}(y_2)))$ 表示组合服务 acc 与其组成服务间推理映射序列的第一个元素 (y_1, a_1)。 $\varphi(\text{acc}, \text{head}(y_2))$ 为组合服务 i 上的一个服务调用。进程 C 将动作 $\varphi(i, \text{head}(y_2))$ 分解成为组合服务和组成服务间的映射序列,并返回结果消息。其中 θ, ψ, ζ 为约束条件,检查异常事件和判定服务调用中组成服务序列(映射序列)的有效性。判定方法如下(op 约束集,例如本例中的 θ, ψ, ζ 构成约束集, S 为组合服务 i 调用消息集(调用组成服务), s 为组合服务的组成服务序列):

- $\forall r \in op, \forall s \in S, s \notin \text{AgainstClass}(r, S)$, 说明当前产生的服务组合序列没有错误;
- $\forall r \in op, \exists s \in S, s \in \text{CompatibleClass}(r, S)$, 说明当前产生的组合服务序列是有效的。

5 相关工作比较

网格服务组合技术是新兴的综合性课题,企业和研究人员已经提出若干针对组合服务的建模语言,如 WSFL^[2], WSCDL^[3], BPEL4WS(简称 BPEL)^[4]等。但是这些语言虽然给出了服务组合和交互的描述标准,但是由于 BPEL 是基于 XML 的代码描述,不具备形式化描述基础,因而不能保证服务组合的准确和有效性。当前形式化描述服务的方法主要包括有限自动机、进程代数和语义网。文[7,8]使用进程代数 LOTOS 描述服务,实现 BPEL 和 LOTOS 的自动转换;而基于有线状态自动机的方法 LTSA-WS,是将类似 BPEL 描述的组服务和工作流自动转换成有线状态进程,来检查安全性和有效性^[9,10]。文[17]使用 Mealy Finite State Machine (MFSMs)描述服务个体和组合服务,该方法的局限在于不能处理组合服务的本地属性和全局属性的关系,当产生消息排队或是服务并发执行时,会有不可预见的行为发生。在语义网的相关应用领域,较完备的体系构架是 DAML-S (Darpa Agent Markup Language Services)^[11-13],它定义了本体库和相关语言用来描述服务,从而能够实现自动化的服务发现、服务调用、服务组合和执行监控,但是没有相应的工具的支持。文[18]使用基于 CSP 的语言来描述交互操作的合法顺序,但只能模拟双方的交互行为,对于多个参与主体的交互,无法模拟。文[19]使用 Petri nets 描述服务,但是该模型不能处理组合服务中的事务和异常情况。文[20,22]使用 PA (process algebra) 形式化描述组合服务,但都只是使用 PA 完成与

BPEL 或 WSDI 的转换,对于组合服务的交互行为并没有给出形式化的描述和推理。文[23]尽管给出了基于 Pi-calculus 的组合服务的形式化描述,但是没有刻画服务的交互行为。

基于网格服务的不确定性,我们采用扩充的 Pi-calculus 来刻画网格组合服务的交互行为。与上述工作不同的是,本文首次提出网格组合服务的描述必须遵循抽象和具体相结合的原则,并将网格服务组合的层次结构和组合服务的动态交互特征纳入统一的概念体系,同时考虑了组合服务内外环境的条件约束,形式化描述了组合服务交互行为的模拟推理。

结论 形式化描述网格服务,是指导和验证服务的组合,产生机器可读的信息,实现网格服务组合优化的前提和基础。本文针对网格服务的动态行为交互特征和网格服务组合调用的有效性检查,提出基于 Pi-calculus 语义扩充的 CPi-calculus 及其操作控制语义,给出网格服务的行为交互模型和交互行为语义的形式化描述。与已有的形式化描述方法相比较,使用本方法能更好地推导验证组合服务的交互行为,有利于网格组合服务的结构描述向可执行规范转化;同时提供了行之有效的建模、推导和验证机制,有着重要的理论价值和实际意义。下一步工作将围绕如何验证组合服务的正确性和解决冗余问题而展开研究。

参考文献

- Foster I, Kesselman C, Nick J M, et al. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002
- Leymann F. Web Services Flow Language (WSFL 1. 0). IBM, May 2001
- World-Wide Web Consortium. Web services choreography description language. Candidate Version 1. 0. Geneva, Switzerland, November 2005b
- Andrews T, Curbera F, Dholakia H, et al. Business process execution language for web services. Version 1. 1. BEA, IBM, Microsoft, SAP, Siebel, May 2003
- Koehler J, Srivastava B. Web service composition: Current solutions and open problems. In: Proceedings of the 13th International Conference on Automated Planning & Scheduling, Trenton, Italy, 2003, 28~35
- Meredith L G, Bjorg S. Contracts and Types. Communications of the ACM, 2003, 46(10): 41~47
- Ferrara A. Web services; a process algebra approach. In: Proceedings of Second International Conference on Service-oriented Computing. New York, USA; ACM Press, November 2004. 242~51
- Chirichiello A, Salaun G. Encoding abstract descriptions into executable Web services; towards a formal development. In: Proceedings of Web Intelligence 2005. New York, USA; Institution of Electrical and Electronic Engineers Press, December 2005
- Foster H, Uchitel S, Kramer J, et al. Model-based verification of web service compositions. In: Proceedings of Automated Software Engineering 2003. Montreal, Canada, 2003
- Foster H, Uchitel S, Kramer J, et al. Compatibility verification for Web service choreography. In: Proceedings of Second International Conference on Web Services, San Diego, CA, USA, 2004
- Ankolenkar A, Burstein M, Hobbs J, et al. DAML-S: Web Service Description for the Semantic Web. In: Proceedings of the First International Semantic Web Conference (ISWC), Sardinia (Italy), June 2002
- Narayanann S, McIlraith S. Simulation, Verification and Automated Composition of Web Services. In: Proceedings of the 11th Int'l World Wide Web Conference, 2002
- Sirin E, Hendler J, Parsia B. Semi-automatic Composition of Web Services Using Semantic Descriptions. in Web Services: Modeling. In: Proceedings of the Architecture and Infrastructure Workshop in ICEIS 2003, Angers, France, April 2003
- Milner R. Communicating and Mobile Systems: The Pi Calculus. Cambridge, Cambridge University Press, 1999
- Milner R. Communication and Concurrency. Englewood Cliffs; Prentice Hall, 1989
- Gu Q, Chen D X, Yu M, et al. Validation test of distributed program based on events sequencing constraints. Journal of Software, 2000, 11(8): 1035~1040
- Berardi D, Calvanese D, De Giacomo G, et al. Automatic Composition of E-services That Export Their Behavior. In: Proceedings of ICSOC'03, Vol 2910 of LNCS, Italy, 2003, 43~58
- Venzke M. Automatic Validation of Web Services. In: Proceedings of the 8th CaberNet Radicals Workshop, October 2003
- Hamadi R, Benatallah B. A Petri-Net-Based Model for Web Service Composition. In: Proceedings of the 14th Australasian Database Conf Database Technologies. ACM Press, 2003, 191~200
- Gwen S, Bordeaux L, Schaerf M. Describing and reasoning on Web services using process algebra. In: Proceedings of the 2nd IEEE International Conference on Web Services, San Diego, California, USA, 2004, 43~50
- Broggi A, Canal C, Pimentel E. Formalizing Web Services Choreographies. In: ENTCS 105 (WS-FM'2004), Elsevier 2004, 73~94
- Cámara J, Canal C, Cubo J. Issues in the formalization of Web Service Orchestrations. In: Proceedings of the Second International Workshop on Coordination and Adaptation Techniques for Software Entities, 2005
- Lumpe M, Achermann F, Nierstrasz O. A Formal Language for Composition. In: Leavens G, Sitaraman M, eds. Foundations of Component-Based Systems, New York: Cambridge University Press, 2000
- (上接第 6 页)
- Noervaag K, Doukeridis C, Vazirgiannis M. Semantic Overlays for P2P Web Searching
- Crespo A, Garcia-Molina H. Semantic Overlay Networks for P2P Systems; [Technical report]. Stanford University, 2002
- Shen H T, Shu Y, Yu B. Efficient semantic-based content search in P2P network. IEEE Trans. on Knowledge and Data Engineering, 2004, 16 (7): 813~826
- Doukeridis, Nrvag K, Vazirgiannis M. DESENT: Decentralized and Distributed Semantic Overlay Generation in P2P Networks; [Technical report]. AUEB, 2005 <http://www.db-net.aueb.gr/index.php/publications/technical-reports/>
- Castro M, Druschel P, Ganesh A, Rowstron A, Wallach D. Secure routing for structured peer-to-peer overlay networks. In: Proc. of the 5th Usenix Symp. on Operating Systems Design and Implementation (OSDI 2002)
- Wallach D S. A survey of peer-to-peer security issues. In: Proceedings of the International Symposium on Software Security, Tokyo, Japan, November 2002
- Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content addressable network. In: Proceedings of ACM SIGCOMM'2001, August 2001
- Stoica I, Morris R, Karger D, Kaashoek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proceedings of ACM SIGCOMM'2001, August 2001
- Rowstron A, Druschel P. Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, 2001
- Zhao B Y, Huang L, Stribling J, Rhea S C, Joseph A D, Kubiatowicz J D. Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 41~53