基于 Petri 网的 Web 服务组合建模

陈丁剑　吴　健　马满福　胡正国
（西北工业大学计算机学院　西安 710072）

摘　要　针对 Web 服务组合流程进行建模，可以实现可靠的服务组合。文中首先提出了一种基于 Petri 网理论的 Web 服务组合建模方法，对服务组合进行形式化建模，然后通过分析 Web 服务网的可达性和活性对 Web 服务组合进行验证，最后举例说明此方法的应用。

关键词　Web 服务，Web 服务组合，Petri 网

Web Service Composition Modeling Based on Petri Net

CHEN Ding-Jian　WU Jian　MA Man-Fu　HU Zheng-Guo
(College of Computer, Northwestern Polytechnical University, Xi'an 710072)

Abstract　It is very important that there is need for modeling techniques and tools for reliable Web service composition, so a Petri net-based model for Web service composition is proposed. The model defines the Web service net, and then analyzes its reachability and liveness to validate the Web service composition. Finally, an example proves this model.

Keywords　Web service, Web service composition, Petri net

近年来，Web 服务作为一种新兴的 Web 应用模式，是一个崭新的分布式计算模型，是 Web 上数据和信息集成的有效机制，发展非常迅速。Web 服务的目的是要解决异构平台上的数据和应用的整合与共享问题。由于各种 Web 服务可能是运行在各种异构系统中，以不同的方式创建、用不同程序语言实现、由不同供应商提供的，那么服务的请求需要根据特定的应用背景和需求进行合理的服务组合；另外，为了可重用性，基本服务(basic service)不可能很复杂。因此需要按照一定的粒度进行 Web 服务的组合。单个 Web 服务可能只提供唯一的调用函数来完成一个单一的功能，而将多个 Web 服务进行有机组合就可以完成一系列的复杂任务，例如 SOA (Service Oriented Architecture)就是利用组合 Web 服务进行应用整合的构架。到目前为止，一些 Web 服务组合语言，譬如 WSCI、WSFL 和 BPEL4WS，对如何将多个基本服务组合成一个复杂服务做了描述，但都仍然是语法级别的，它们无法验证服务组合是否正确，无法验证组合服务是否能在有限步内结束。因此需要对 Web 服务组合流程进行形式化建模，以实现可靠的组合服务。本文提出一种基于 Petri 网理论的 Web 服务组合建模方法，对服务组合进行形式化建模和验证，分析其可达性和活性，并举例说明此方法的应用。

1　Web 服务组合的 Petri 网模型

Petri 网最初是由德国人 C. A. Petri 在 20 世纪 60 年代提出来的，此后得到了很大的发展，并被应用到了众多的领域。一个 Petri 网是一个有向连通图，其节点分别称为库所和变迁。每个库所代表一种资源，而库所中的托肯数表示资源的数量。当变迁 t 的所有输入库所中都至少含有一个托肯时(标识 M)，t 就可以发生，记为 $M[t\rangle$。变迁发生的结果是从每个输入库所中移除一个托肯，而给每个输出库所移入一个托肯，此时到达标识 M'，记为 $M[t\rangle M'$[1]。

1.1　Web 服务的 Petri 网模型

一个 Web 服务可以直接映射为一个 Petri 网。Web 服务中的操作作为变迁，Web 服务的状态作为库所，操作和状态之间的因果关系则作为库所和变迁之间的流关系。同时我们假设每个 Web 服务的 Petri 网都有一个输入库所 i 代表其输入信息，和一个输出库所 o 代表其输出信息。Web 服务的状态不外乎“未实例化”、“就绪”、“执行”、“暂停”和“完成”这五种[8]。当 Web 服务处于“就绪”态时就意味着输入库所 i 中有托肯，与 i 相连的某个变迁具有发生权，此时 Petri 网标识记为 M_i；处于“完成”态时就表示输出库所 o 中有托肯，且不再有任何变迁可发生，此时 Petri 网标识记为 M_o。因此，可以对一个 Web 服务作如下定义[2]：

定义 1　四元组 $S=(SName, SWsal, CS, SN)$称为一个 Web 服务，当且仅当它满足下面几个条件：

(1) $SName$ 为服务名称，是 Web 服务的唯一标识；

(2) $SWsdl$ 为 Web 服务描述，其中描述了服务功能，服务调用地址；

(3) CS 为构成 Web 服务的组件服务(Component Services)集合，如果 $CS=\{SName\}$，则 S 是一个基本服务，否则 S 是一个组合服务，即 S 是由各个基本服务迭代组合而成；

(4) $SN=(P, T; F, i, o, l, m_i, M_o)$为 Web 服务的 Petri 网(Service Net)，其中：

- P 为有限库所集，其中每个库所代表一种状态，其容量是无限的；
- T 为有限变迁集，其中每个变迁代表 Web 服务中的一个操作；
- $F\subseteq(P\times T)\cup(T\times P)$为流关系，其中每条有向弧代表一个从操作到状态或从状态到操作之间的因果关系；

陈丁剑　博士生，主要研究方向：Web 服务和语义 Web；吴　健　教授，主要研究领域：软件工程、组件技术；马满福　讲师，博士生，主要研究方向：计算机系统结构、网格计算；胡正国　教授，博士生导师，主要研究方向：高性能计算、软件理论。

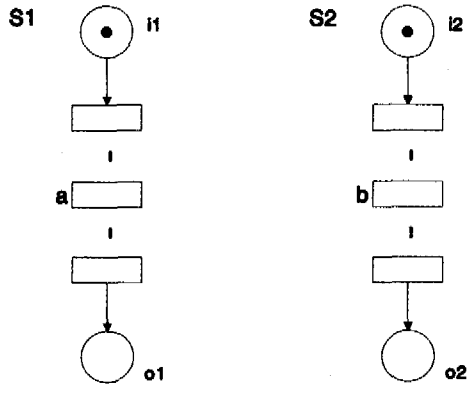

图 1　Web 服务 S_1 和 S_2 的 Petri 网描述

• i 为输入库所，o 为输出库所，即 $\cdot i=\varnothing, o^{\cdot}=\varnothing$；

• $l: T\rightarrow A\cup\{\tau\}$，其中 A 为 Web 服务中的操作集，而 τ 为空操作。

• M_i 为 SN 的起始标识，M_o 为 SN 的终止标识，且有 $M_i(i)=M_o(o)=1, M_i(o)=M_o(i)=0$。

显然，当 SN 处于标识 M_i 时，S 开始执行，直到 SN 处于标识 M_o 时，S 结束。空操作 τ 用黑色方框表示。图 1 为两个独立 Web 服务 S_1 和 S_2 的 Petri 网描述。

1.2　组合 Web 服务

组合 Web 服务可以由多个基本服务或组合服务嵌套组合而成，但最基本的组合结构包括如下五种：顺序、选择、循环、并行和调用。

(1) 顺序

顺序结构允许两个 Web 服务 S_1 和 S_2 组合以后一个接一个地顺序执行，即 S_1 必须在 S_2 开始之前完成。组合服务 S 满足：$i=i_1, o=o_2$。当一个服务需要用到另外一个服务的输出时就要使用顺序结构来组合。例如网上购书时，要先找到想买的书，然后才能定购，即先执行 *Locate Service*，然后再执行 *Order Service*。顺序结构的 Petri 网描述如图 2(a)。

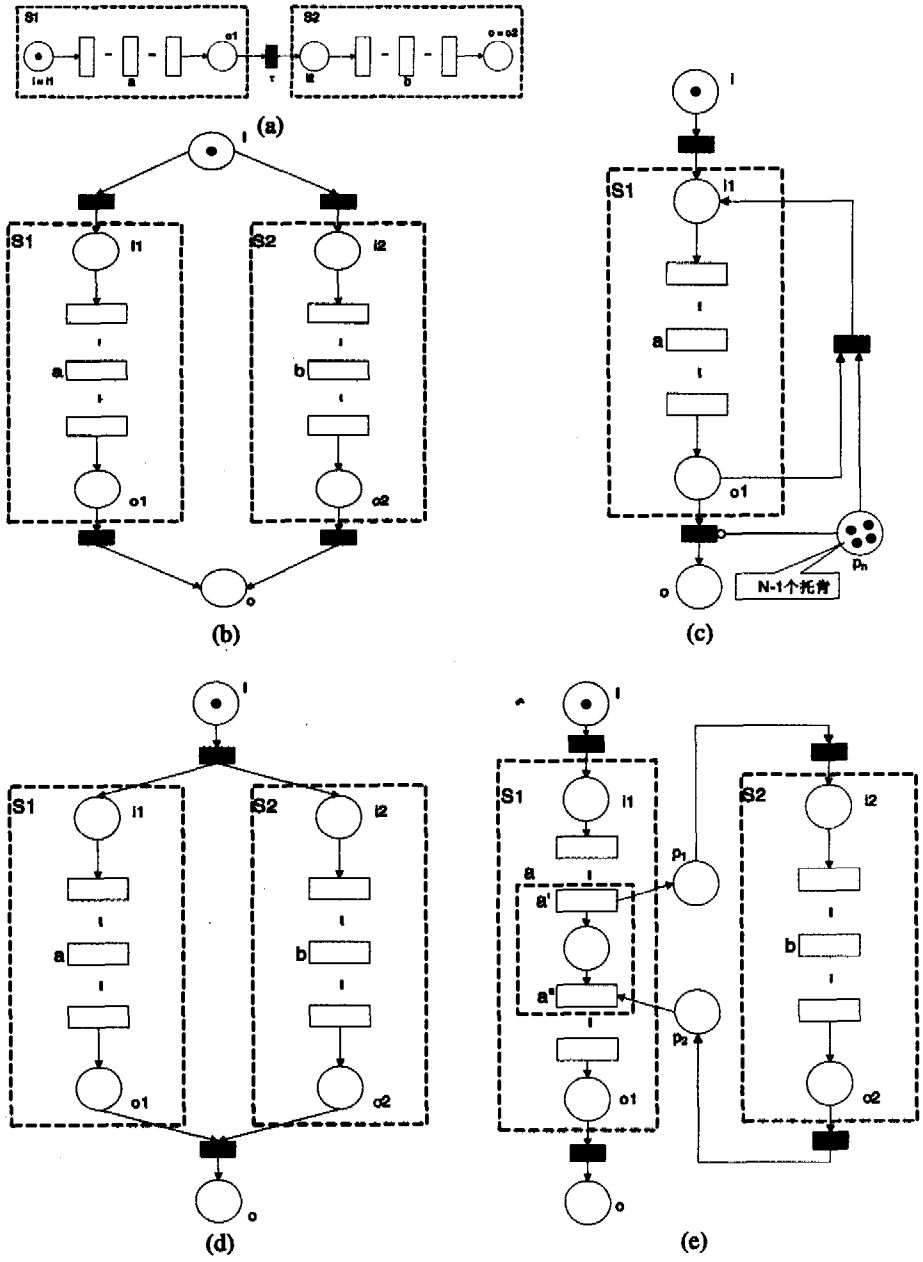

图 2　五种基本组合结构：(a)顺序(b)选择(c)循环(d)并行(e)调用

(2) 选择

当要实现只执行 S_1 和 S_2 中的一个的时候需要使用选择

结构来组合。通过选择结构可以做到或者执行 S_1，或者执行 S_2，且必须执行一个服务。例如用户为所购买的书籍进行网上支付时，他可以选择使用招商银行或者工商银行来支付，但必须使用一种支付方式。即执行 *PayByCMB Service* 或者执行 *PayByICBC Service*。选择结构的 Petri 网描述如图 2(b)。

(3) 循环

有时候需要多次重复执行同一个 Web 服务来完成既定任务，此时可以通过循环结构实现。如图 2(c)所示，把库所 P_n 作为服务出口变迁的约束，并且 P_n 中有 $N-1$ 个托肯，表示循环执行 S_1 共 N 次。

(4) 并行

并行结构允许 Web 服务 S_1 和 S_2 可以不分先后地完成，但是一定要两者都执行完成后整个服务组合才算执行完成。例如，在线组装电脑，用户可以同时并行地购买各种配件(CPU、内存等等)，但是必须等到所有配件都购买齐全后才能将一台完整的电脑组装好。这种情况下，就要用到并行结构来完成服务组合。并行结构的 Petri 网描述如图 2(d)。

(5) 调用

假设一个已有 Web 服务 S_1 中的某个操作原本在本地实现，但是由于环境和条件的变迁，本地不能或者不足以再实现此操作，而需要通过调用另外一个 Web 服务 S_2 来完成任务，此时就形成了调用组合结构。调用结构的 Petri 网描述如图 2(e)。

有了上述五种基本结构，就可以组成各种复杂的组合。很显然，通过这五种组合结构得到的服务组合同样有且仅有一个输入库所 i 和一个输出库所 o，因而复杂的组合 Web 服务可以用定义 1 中的 Petri 网模型来描述。给 Web 服务建模以后，就可以在 Petri 网定义的范围内对其性质和行为进行分析和仿真。

2 Web 服务验证

组合服务中的各个子服务可能由不同的供应者所创建和发布，但是它们之间又必须进行交互，是一种紧耦合的关系，因此服务组合以后可能产生与预期不一样的效果，影响客户的使用，所以必须在组合服务发布之前验证其正确性。而之所以用 Petri 网对 Web 服务建模，就是因为通过研究 Petri 网的活性和可达性可以形式化验证服务组合正确性，以及验证组合服务是否能在有限步内结束。因此首先要定义 *SN* 的变迁规则。

设 s_1 和 s_2 为两个基本 Web 服务，由这两个基本服务组合成的组合服务为 S。基本服务内部的变迁不予考虑，因为我们假定基本服务是正确的，我们只需考虑基本服务整体的变迁规则，即将基本服务作为一个变迁来看待。根据定义 1，每个基本服务 s 都有且仅有一个输入库所和一个输出库所，所以只要输入库所中具有托肯(标识 M)，则整个基本服务就可以正常执行，并最终在输出库所中放入托肯(标识 M')，记为 $M[s\rangle M'$。针对五种基本组合结构，只要在 i 中放入托肯，则整个组合服务就可以执行，发生变迁。因而由五种基本组合结构迭代组合出来的组合服务同样只要在其最开始的输入库所 i 中放入托肯，就可以驱动整个服务的执行，记为 $M[s\rangle M'$。这就是 *SN* 的变迁规则。但是要注意的是，这里所说的放入托肯只是在 Petri 网领域中的说法，此时的托肯是无具体意义和不加区分的，但在实际应用中，要区分各个 *SN* 的输入条件和输出结果，对各种组合结构进行匹配。例如顺序结构中，S_1 的输出结果必须与 S_2 的输入条件相匹配才是正确的组合。这种匹配是通过语法甚至语义层次的匹配验证来保证的。

有了变迁规则，下面就可以定义 S 的活性。

定义 2 对基本服务 $s\in S$，若对任一可达标识 $M\in[M_0\rangle$，均有从 M 可达的标识 $M'\in[M_0\rangle$，使得 $M'[Ms\rangle$，就说基本服务 s 是活的。若所有 $s\in S$ 都是活的，就说 Web 服务 S 是活的。

定义 2 的含义很明显，就是组合服务的活性要求组合服务中所有的基本服务都必须是活的。而要确定基本服务是否活的，就必须知道 *SN* 的可达标识集 $[M_0\rangle$。$[M_0\rangle$ 可以通过构造可达树 $T(SN)$ 来获得。更进一步地，构造可达图 $G(SN)$ 则可以更直接明了地验证 S 的活性，进而验证 S 能否在有限步内结束。具体的可达树和可达图的构造算法和相关定理这里不再赘述，请读者自行参考相关文[1,6]。

3 应用示例

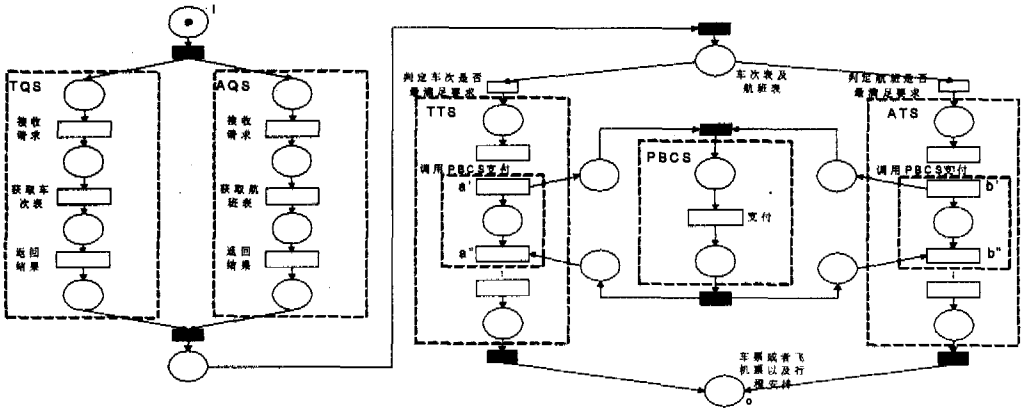

图 3 行程规划服务 Petri 网描述

某西安公司的职员想通过乘火车或者乘飞机到北京开会，他使用网上行程规划服务(Schedule Service)来安排自己的行程。行程规划服务具体过程如下：首先同时询问火车站和航空公司是否有当天从西安到北京的列车或者航班，然后根据客户所要求的时间、价格以及服务质量等方面来综合评估，从中选择一个最能满足要求的车次或者航班；接着向火车站或者航空公司请求订票服务，并通过招商行信用卡进行网

(下转第 135 页)

Step 3.2：对包含在 $doc\_set(s_i)$ 中的每一个文档 d_{ij}，为其构造一个消息 m_{ij}；

Step 3.3：在每一个消息 m_{ij} 与每一个从外界接收 d_{ij} 的服务端口 p_{ij} 之间构造一个映射 σ；

Step 3.4：在消息与构件接口之间建立映射 ξ。如果一个接口 inf 接收某一单独的消息 m_{ij}，那么使用 MM 的映射方式；如果某一消息的一部分信息即可满足接口 inf 的需求，那么使用 MD 映射方式；如果接口 inf 需要多条消息的复合，那么使用 MS 的映射方式。图 7 种给出了 MM、MS 和 MD 的几个示例；

Step 3.5：使用特定的程序语言来实现 s_i，例如 VS.Net、J2EE 等；

Step 3.6：使用 UDDI 技术封装与描述 s_i，并为 $msg\_set(s_i)$ 中的每一个消息书写 SOAP 格式。

结论 本文分析了目前在面向服务的计算领域存在的一个关键问题，即不存在成熟的服务识别与设计方法。为解决该问题，讨论了业务层面的服务与企业内/企业间业务过程之间的映射，以及技术层面的服务与基于软构件的软件架构之间的映射。基于这两类映射，提出了一种服务的规范化模式 SNF，建立了好的服务应遵循的标准，以及规范化服务的识别与设计方法。该方法将企业管理模式，企业内信息系统与服务的识别、设计紧密结合在一起，为企业发现有价值的 Web 服务提供了有效的手段，并支持企业内信息系统与企业间电子商务系统的有效集成。

参 考 文 献

1 Papazoglou M P, Georgakopoulos D. Service-Oriented Computing. Communications of the ACM, 2003, 46(10): 25～28
2 Yang J. Web Service Componentization. Communications of the ACM, 2003, 46(10):35～40
3 Papazoglou M P, Yang J. Design Methodology for Web Services and Business Processes. In: Proceedings of the Third International Workshop on Technologies for E-Services, Lecture Notes in Computer Science, London, UK, 2002, 2444: 54～64
4 Curbera F, Khalaf R, Mukhi N, et al. The Next Step In Web Services. Communications of the ACM, 2003, 46(10): 35～40
5 W3C. Web Services Description Language (WSDL) 1.1. Mar. 15, 2001. http://www.w3.org/TR/wsdl
6 UDDI specification, version 3.0. http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3
7 W3C, SOAP specification, version 1.2. Jun. 24, 2003. http://www.w3.org/TR/soap12/
8 Curbera F, Goland Y, Klein J, et al. Business Process Execution Language for Web Services. Jul 31, 2002. http://www-106.ibm.com/developerworks/ webservices/library/ws-bpel
9 Lee R, Kim H K, Yang H S. An Architecture Model for Dynamically Converting Components into Web Services. In: Proceedings of 11th Asia-Pacific Software Engineering Conference (APSEC'04), Busan, Korea, 2004. 648～654
10 Hanson J. Coarse-grained interfaces enable service composition in SOA. http://builder.com.com/5100-6386_14-5064520.html
11 Anexinet Corp. Enabling Service-Oriented Architecture: Aligning the enterprise for agility and competitive advantage: [Technical White Paper]. 2004
12 Sanders R T, Bræk R, von Bochmann G, et al. Service Discovery and Component Reuse with Semantic Interfaces. http://beethoven.site.uottawa.ca/dsrg/PublicDocuments/Publications/Sand05a.pdf
13 Cape Clear Software Inc. Principles of Service-Oriented Architecture (SOA) Design: [Technical Report]. CPV-DOC-303, 2004
14 Keith L, Mili A. A goal-driven approach to enterprise component identification and specification. Communications of the ACM, 2002, 45(10): 45～52
15 Jung J Y, Hur W, Kang S H, et al. Business Process Choreography for B2B Collaboration. IEEE Internet Computing, 2004, 8(1): 37～45
16 Lambros P, Schmidt M T, Zentner C. Combine business process management technology and business services to implement complex Web services: [White Paper]. IBM Corp, 2001

(上接第 130 页)

上支付；最后将车票或机票以及行程安排返回给客户。从上述过程可以看到，Schedule Service 是由 Train Query Service (TQS)、Airline Query Service (AQS)、Train Ticket Service (TTS)、Airline Ticket Service (ATS) 和 PayByCMB Service (PBCS) 组合而成。用 Petri 网对 Schedule Service 建模如图 3 所示，其中用到了平行、选择、顺序和调用组合结构。针对图 3 的 Petri 网，我们可以构造其可达树或者可达图以研究其活性，从而验证 Schedule Service 的正确性。

4 相关工作

文[2]提出了一种 Web 服务的 Petri 网模型，对 Web 服务网做了定义，并在此基础上提出了 Web 服务代数，但是与我们的模型相比较，文[2]中的模型过于复杂，在实际应用中不便使用，并且没有定义 Web 服务网的变迁规则以及活性。文[3,4]对现有 Web 服务和 Web 服务组合技术做了总结，它们指出除 Petri 网外，还可以使用 π 演算和有限状态机等手段对服务组合进行形式化建模，并且各种方法各有优劣。一些文献使用 Petri 网为 WSCI、WSFL 和 BPEL4WS 等服务组合语言建模[7]，具有一定的参考价值。除此以外，还有文献描述了如何使用 Petri 网对工作流进行建模和分析。

结束语 Web 服务组合是 Web 服务应用的重要手段，需要对 Web 服务组合流程进行建模，以实现可靠的组合服务。文中首先根据 Petri 网理论定义了 Web 服务及其服务网 (Service Net)，并给出了五种基本组合结构的 Petri 网模型；然后定义了服务网的变迁规则，从而进一步分析其可达性和活性，以验证组合服务的正确性，以及组合服务是否能在有限步内结束；最后举例说明此方法的应用。Petri 网提供了一种有效的手段去模拟、分析和验证 Web 服务组合，但如何使用 Petri 网对语义 Web 服务以及自动化的 Web 服务组合进行建模则是我们进一步的研究课题。

参 考 文 献

1 袁崇义. Petri 网原理. 电子工业出版社, 1993
2 Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition. ADC, 2003, 2003, 17
3 Milanovic N, Malek M. Current Solutions for Web Service Composition. IEEE Internet Computing, NOVEMBER · DECEMBER 2004
4 Papazoglou M P, Dubray J. A Survey of Web Service Technologies: [Technical Report]. June 2004
5 Zhang J, Chang C K, Chung J Y, Kim S W. WS-Net: A Petri-net Based Specification Model for Web Services. ICWS'04, 2004
6 蒋昌俊. Petri 网的行为理论及其应用. 高等教育出版社, 2003
7 孙健，陶晓峰. 基于 Petri 网的 Web 服务 BPEL4WS 建模与分析. 计算机工程, 2004, 30(22)
8 Cardoso J, Sheth A. Introduction to Semantic Web Services and Web Process Composition. SWSWPC 2004, LNCS 3387, 2005