

DOI: 10.7652/xjtub201308001

依赖知识的服务组合算法

鲍军鹏, 宋楠, 陶斌

(西安交通大学计算机科学与技术系, 710049, 西安)

摘要: 针对面向服务的体系结构中 Web 服务和服务组合技术在下一代互联网和云计算领域中的应用问题,提出了一种依据知识对原子服务进行组合的 3 步服务组合算法,分为相邻节点连接匹配、断枝节点清理和原子服务组合 3 个步骤。相邻节点连接匹配就是完成相邻的服务节点之间连接匹配,将所有允许的连接都存入弧的连接链表中。通过断枝节点清理,删除所有没有连接的候选原子服务,因为在候选队列中会存在许多断枝服务节点。通过原子服务组合,从清理过断枝节点之后的剩余候选服务中整理出服务组合实例。实验表明:算法可以实现服务的自动化装配,不需要用户参与服务调用过程,通过原子服务间的可连接性以及知识模型生成服务组合实例,从而实现原子服务数量和检测范围的有效控制。

关键词: 面向服务的体系结构;服务组合;Web 服务

中图分类号: TP311 **文献标志码:** A **文章编号:** 0253-987X(2013)08-0001-06

A Knowledge-Based Service Composition Algorithm

BAO Junpeng, SONG Nan, TAO Bin

(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A knowledge-based service composition algorithm is proposed to apply the techniques of service-oriented architecture including Web service and service composition in the area of the next generation Internet and the cloud computing. The algorithm consists of three key steps, i. e. connection matching between adjacent nodes, deleting broken nodes, and building service composition instance. All possible links between candidate primary service nodes are found and stored in an arc list at the first step. Then, the candidate primary service nodes that can not make a legal path are deleted in the second step. A service composition instance is built from the rest candidates in the step 3. The algorithm automatically assembles services without user's intervention during the service procedure, and builds the service composition instance in terms of a knowledge model and the connectivity between primary services. Test results show that the number of primary services and detection ranges can be effectively controlled.

Keywords: service oriented architecture; service composition; Web service

在 IBM、谷歌、微软、百度等大型 IT 企业和众多研究学者的推动下,云计算概念已经形成了重要发展趋势,而面向服务的体系结构(Service Oriented Architecture, SOA)和软件即服务(Software as

a Service, SaaS)^[1]是云计算中的核心技术之一。SOA 和 SaaS 的理念是把软件看做像水、电、广播电视一样持续性的服务,软件生产者将转变为服务提供者,向用户提供可满足其需求的服务。

收稿日期: 2012-10-29。 作者简介: 鲍军鹏(1974—),男,副教授。 基金项目: 国家自然科学基金资助项目(60903123);百度大规模机器学习与数据挖掘研究计划资助项目。

网络出版时间: 2013-04-28

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20130428.0908.001.html>

Web 服务^[2]是 SOA 和 SaaS 中服务概念的功能载体和具体表现形式。网络中已有海量稳定易用的服务,但是仅仅依靠单个服务来满足用户需求往往会捉襟见肘,将多个服务以一定方式组合起来,则能够发挥更大的潜力。服务组合的另一个优点在于能使服务开发者更专注于自身核心功能的设计与实现,剔除其他辅助“胶水”功能,更有利于服务开发。因此,将多个服务组合起来满足用户应用需求已经成为一种全新的、迫切的软件生产方式。

本文将软件即服务的概念应用到自组织的服务体系结构中,提出了一种基于知识的自组织服务组合体系结构原型——数据挖掘服务体系结构(Data Mining Service Architecture, DMS Arc)^[3],用于提供数据挖掘服务。DMS Arc 通过对传统服务部署框架进行改变,增加了知识专家的角色和服务的本体描述信息,能够根据知识对原子服务进行组合,为服务请求者完成服务组合。本文则详细介绍该体系结构中的服务组合算法,通过使用网络中的海量原子服务,快速响应用户请求,并且及时、有效地实现服务组合,为用户提供最佳的服务组合结果。

DMS Arc 的假设前提是:不考虑服务搜索中服务查找的难度,服务在统一的服务注册库中进行服务注册管理;根据知识模型与用户的描述,获得实际的服务需求;在服务匹配时通过查询获得服务注册库中的服务描述,与需求进行匹配,匹配完成后获得符合用户需求的服务描述,通过这个描述获取各个服务,最后进行服务之间的组合连接。

1 相关工作

根据背景知识的不同,服务组合技术可分为 3 种基本策略:①基于流程建模;②基于语义网;③基于部署的其他方法。表 1 列出了常见的服务组合方式和代表性方法。表中的各种方法并不是完全彼此互斥,不同方法可以混合使用,例如语义 Web 模型可与工作流模型同时使用等。

基于工作流模型的典型代表是 Web 服务的业务流程执行语言 BPEL4WS^[4] (Business Process Execution Language for Web Services),它是一种全面的业务流程自动化框架,已经成为一项规范标准。BPEL4WS 的主要功能是将一组现有原子服务通过服务交互、操作数据、流程生命周期或其他活动(如终止、等待、捕获等)连接起来,从而定义一个组合 Web 服务。

基于状态演算的服务组合方法主要是依靠建立

服务组合描述与服务状态模型的一一映射关系,然后通过形式化的模型和工具分析服务组合系统。状态演算模型的主要代表是 Petri 网模型^[5-6],它对于“安全、死锁、可达”的定义在处理服务并发、冲突与同步时具有很大优势。Tang 等人将服务操作对应 Petri 网的变迁、服务状态对应库所,用库所与变迁之间的箭头表征状态间的因果关系^[5]。其优点在于状态演算侧重于对服务组合的形式化建模,有直观的图形化建模,缺点在于本质上仍然是基于工作流模型的服务方法,适用于处理静态系统描述,难以处理服务组合的动态特性。

表 1 服务组合方法分类

背景角度	驱动类型	组合模型	代表方法
流程建模	过程驱动	工作流模型	BPEL4WS ^[4]
		状态演算模型	Petri 网模型 ^[5-6]
		进程代数模型	π -演算模型 ^[7]
	模型驱动	知识复用模型	ViPen ^[8] , MDA ^[9]
语义网	语义驱动	语义 Web 模型	CoSMoS ^[10] , ASA-M ^[11]
部署	服务部署	P2P 架构模型	SELF-SERV ^[12]
		网格计算模型	WSRF ^[13] , Rudder ^[14]

进程代数模型是指用于描述和分析并发、异步、非确定和分布式等系统行为的进程代数,它是一种对动态实体进行建模的语言,具有严密的形式化语义,主要代表为 π -演算模型^[7]。 π -演算是一种以进程间移动通信为研究重点的并发理论,在 π -演算中实体统称为名字,其基本计算实体为名字和进程,进程之间的通信通过传递名字来完成。

模型驱动的 Web 服务组合方法^[8-9]的最大特点是将服务组合的设计与实现分离,使得设计与实现可使用不同的标准。该策略使用 UML 来描述服务组合,能够构建出独立于平台的模型,所以适用范围广,但是其缺点是需要集成规划工具和服务组合服务器等子系统,系统实现过程比较复杂。

与过程驱动不同,语义驱动的服务组合^[10-11]策略主要强调服务自描述,其目标是期望能够使 Web 服务描述和服务请求成为被计算机所理解的语义,最终能够使服务组合自动完成,因此在语义驱动中最重要的是服务描述。一般来说,语义服务描述需要覆盖数据输入、数据输出、条件输入、输出影响和服务质量 5 部分语义。

还有一些服务依据不同的网络结构和部署情

况,形成了面向特定环境下的组合方式,例如针对 P2P 架构^[12]和针对网格环境的服务组合^[13-14]等。

2 服务组合过程

2.1 基本环节

DMS Arc 是基于知识的模型驱动式服务组合,实现了一种 3 步服务组合算法。DMS Arc 的核心是服务组合中心,包括服务注册中心(Service Registration Center, SRC)和知识注册中心(Knowledge Registration Center, KRC)。SRC 是服务发现的支持者,它管理已经注册的 Web 服务,对其进行管理,并向服务组合中心(Service Composition Center, SCC)提供查找原子服务的接口。KRC 是知识的管理者,它接收专家撰写的服务组合知识,管理所有的可用知识,提供知识的增删改查等基本功能,并且提供知识推理,以响应更为复杂的组合请求。SCC 依据知识自动地把原子服务装配成可满足用户需求的服务执行实体,并完成服务调用过程,最后将服务的组合执行结果返回给用户,用户通过选择知识来体现对服务的需求。知识刻画了抽象内容,描述了服务可以干什么、达到什么目的,以及完成服务的基本过程(即 DAG),但是知识并没有指定达到目的需要哪些具体服务。因此,本文通过服务组合算法把具体的服务代码和抽象的知识经过匹配、组合以后,形成满足知识要求(即体现用户需求)的服务组合实例。

服务组合算法是 SCC 的核心部分。服务组合中的一个关键问题是对原子服务进行自动组合,避免人工干预,并且有效控制原子服务的数量。本文算法根据知识模型进行服务组合,利用知识中的有向无环图(Directed Acycline Graph, DAG)模型有效控制服务数量,并且能够基于知识进行自动装配完成服务组合。本文将服务组合过程分为 3 个基本

环节:查询匹配、DAG 组合和服务规约(见图 1)。

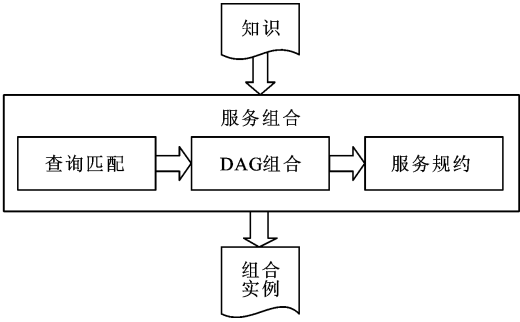


图 1 服务组合的基本环节

2.2 查询匹配

查询匹配就是根据知识中原子需求的描述去寻找能够满足要求的服务。由于网络上的服务种类与数量繁多,能够满足用户需求的服务可能会有多个,如何从海量的信息中过滤出相关有效信息就是服务查询迫切需要解决的问题。服务查询最主要的目的是从众多服务中选择出最适合的一个或者多个服务,作为之后进行服务组合的原子服务,其关键过程就是服务匹配问题。

为了便于系统实现,我们维护了一个 SRC。这个注册中心管理的主要内容就是所有的原子服务信息,所以我们在这个服务库中进行服务查询,不再考虑从一般网络获取服务信息的情况。

服务匹配器经过对知识中的每一个服务需求的描述与服务注册库中服务的描述进行匹配,为知识中的每一个服务找到最佳的服务描述匹配结果。在没有经过服务匹配器时,知识中的内容是抽象的。当服务匹配完成之后,知识中的需求描述变为了服务描述,从抽象内容变为具体内容,同时知识中的每个原子需求获得一个服务描述匹配列表,匹配列表中存放一个或多个匹配结果,即候选原子服务队列。服务查询匹配过程的示例如图 2 所示。

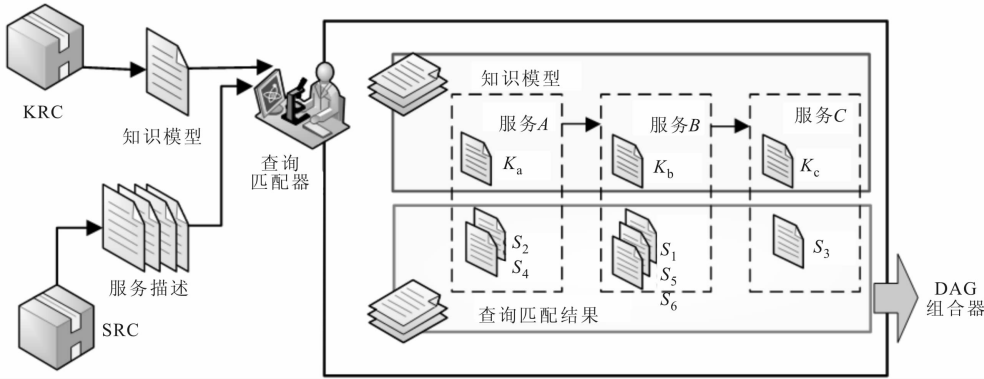


图 2 服务查询匹配过程

2.3 DAG 组合

查询匹配与 DAG 组合共同完成原子服务间的初步组合连接。DAG 组合的主要工作是在候选服务队列中继续进行服务筛选,查询匹配是服务节点的初步装配,装配信息是候选服务队列中的具体服务信息,此时知识已经初步完成从抽象到实际的转换。

在第 2 环节进行原子服务间的 DAG 组合连接装配时,主要依据服务之间消息传递的一致性。如若根据语义网中输入输出参数的消息一致性,还可以将更多的消息形式纳入其中。例如: A 的候选原子服务输出消息(前件消息)是否满足 B 的候选源服务的输入消息(后件消息)要求;只有当前件消息满足后件消息格式时,服务才能继续在候选队列中继续存在。

当对所有相邻候选队列进行消息传递的一致性判定之后,便可以获得原子服务之间的连通情况。然后,根据连通关系对候选队列进行清理,删除所有失去连接的原子服务,最终得到一个(或多个)组合实例。DAG 组合的操作源是知识 DAG,目的是获得服务组合实例。这个过程的一个关键问题是如何从众多候选服务中选择最合适的服务,构成有效服务组合实例,本文在后面将专门讨论这个问题。

2.4 服务规约

服务规约完成服务组合的行为和约束信息及服务质量匹配,主要涉及服务的全局约束条件、服务质量、响应时间以及服务收费计算等内容。

服务规约器从 DAG 组合器接收服务组合实例,并计算所有服务组合实例的服务质量估价值,判断其是否满足知识中的要求。如果满足,则进一步计算出组合实例的服务匹配度,最后使用服务匹配度最高的实例作为服务组合结果,或者根据服务匹配度给出服务组合推荐队列。

假设一个服务组合实例包含了 n 个子服务 $S_1, S_2, \dots, S_n$,其中全局服务质量 Q 需要满足如下条件

$$\bigcup S_i(Q) \Rightarrow K(Q) \quad (1)$$

即所有子服务的服务质量并集能够覆盖知识对服务质量的要求,此时这一组合实例才满足知识中的所有要求。服务组合实例的匹配度 M 可由所有子服务的匹配度 M' 依据下式得到

$$M = \frac{1}{n} \sum_{i=1}^n M' \quad (2)$$

其中 M' 可在第一环节查询匹配时得到。

3 DAG 组合算法

DAG 组合的过程分为 3 步:相邻节点连接匹配、断枝节点清理与原子服务组合。

3.1 相邻节点连接匹配

相邻节点连接匹配就是完成相邻的服务节点之间连接匹配,匹配源为相邻两个节点的候选服务列表,将所有允许的连接都存入弧的连接链表中。对所有弧处理完之后,若出现弧的连接链表为空,则表示此服务组合缺乏可以连接的原子服务,无法完成,下一步则要删除这些没有连接的原子服务。

假设在知识模型中服务 K_a 与 K_b 相邻,且 K_a 的输出作为 K_b 的输入。 K_a 的候选服务队列中有 S_a 、 K_b 的候选服务队列中有 S_b ,当 S_a 与 S_b 满足如下条件时,则认为 S_a 与 S_b 可以连接匹配

$$|S_a(P_i)| \wedge S_a(O_i) \supseteq S_b(I_n) \wedge |S_b(P_r)| \quad (3)$$

式中: $|S_a(P_i)|$ 表示如果 S_a 与 S_b 连接, S_a 的后置条件得到满足; $|S_b(P_r)|$ 表示如果 S_a 与 S_b 连接, S_b 的前置条件得到满足; $S_a(O_i) \supseteq S_b(I_n)$ 表示 S_a 的输出包含 S_b 的输入,即 S_b 的所有输入都能够被满足。

3.2 断枝节点清理

断枝节点清理就是删除所有没有连接的候选原子服务。经过相邻节点连接匹配之后,在候选队列中会存在许多断枝服务节点,这些断枝节点有些是输入输出都没有连接,有些是输入输出中只有一个方向有连接等。

例如,图 3 显示了某知识中服务 DAG,其中 A 、 B 、 C 、 D 和 E 分别表示不同的服务,服务 D 需要 2 个先导服务 B 和 C 才能满足,即 BD 和 CD 都满足的情况下服务 D 才能执行。图 4 显示了对应该 DAG 的候选服务节点图,其中每个服务各有 3 个候选原子服务,并且经过服务查询匹配与相邻服务连接之后,形成了相互连接图,但是其中只有阴影节点可以构成符合图 3 要求的有效服务组合实例,其他候选服务节点都是断枝节点。图 4 中: b_1 没有任何节点与之连接,可以删除; b_2 虽然在输入方向上与 a_1 连接,但是输出方向没有连接,所以也可以删除;

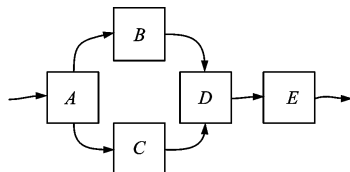


图 3 一个服务有向无环图

当 b_2 被删除之后, a_1 也失去了所有连接,也可以被删除; d_2 在输入与输出方向上都有连接,但是没有从 B 候选节点 ($b_1 \sim b_3$) 到 d_2 的连接,即 d_2 不满足 BD 弧,所以也应该被删除;当 d_2 被删除之后, c_2 、 e_2 也都被删除; a_3 、 d_3 与 d_2 的情况类似,所以都会被删除,同样也会导致 c_3 、 e_3 被删除。

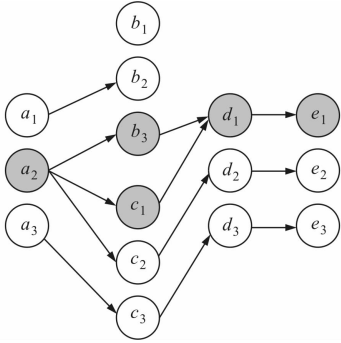


图 4 候选服务节点图

不能形成有效连接的候选服务节点即为断枝节点。由于断枝节点缺少必要连接,它不可能被用来组合成服务实例,所以必须从服务 DAG 中删除。如前所述,当一个断枝节点被删除之后,可能会导致其他节点也变成断枝节点,所以断枝节点清理过程需要反复遍历服务节点的候选服务队列。

3.3 原子服务组合

原子服务组合就是从清理过断枝节点之后剩余的候选服务中整理出服务组合实例,这些是有效的服务组合实例,有效的组合实例可能存在多个。

完成断枝节点清理后,仍然不能保证每一个候选服务节点都能够组成一个组合实例。再以图 3 中的服务 DAG 为例,图 5 显示了另外一种候选情况。其中,所有候选节点都不是断枝节点,因为它们在每个方向上都至少有一个连接,但是只有图中的阴影节点能够组成服务组合实例,其他节点无法形成符合图 3 的 DAG 服务组合实例。

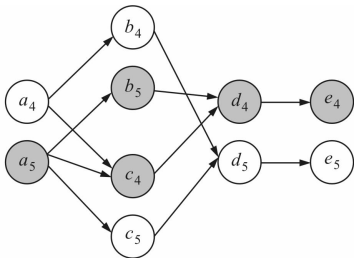


图 5 无断枝节点情况下的有效服务组合和无效组合

为了进一步阐述组合算法的具体实现过程,下面先给出几个定义。

无断枝图:对于一个 DAG 模型,图中的节点为 DAG 服务节点的候选原子服务,边为原子服务之间的连接。如果这个图中没有断枝节点,那么这个图是 DAG 模型的无断枝图。

组合子图:对于 DAG 模型无断枝图的一个子图,若子图的节点数与 DAG 服务节点数相同,且子图中每个节点都属于不同服务节点、DAG 模型的每个弧都能在子图找到一个对应连接,那么这个子图是该 DAG 模型的一个组合子图。组合子图就是一个服务组合实例。

组合树:将 DAG 模型视为一个无向图,则这个无向图的任意一棵生成树称之为该 DAG 模型的一棵组合树。

增长弧:对于 DAG 模型中的一条弧,如果在其组合树中存在一条边与之对应,则该弧称之为增长弧。

判定弧:对于 DAG 模型中的一条弧,如果在组合树中不存在一条边与之对应,则该弧称之为判定弧。

原子服务组合的目的就是要在一张经过断枝节点清理后获得的无断枝图中,找到所有对应于 DAG 模型的组合子图。算法将寻找组合子图的过程分为两个阶段:组合树生成阶段与组合树填充阶段,即先生成一棵树,然后再填充判定弧构成图。

(1)组合树生成,本文使用无向图生成树的方法来寻找组合子图。把一个 DAG 模型看作无向图,任选一个节点作为根节点,使用生成树算法生成了它对应无向图的组合树。树中的边对应增长弧,树中没有的 DAG 弧即为判定弧。

组合树的基本操作单位为树节点。一个树节点包含 6 个属性: P 指向此节点的父节点; C 链表存放此节点的所有孩子节点; N 指向 DAG 模型中对应的顶点(即服务节点); A 指向与父节点的连接在 DAG 中对应的弧; L 链表存放候选服务节点; I 表示当前候选服务节点在候选服务链表 L 中的位置。 L 链表在树生成阶段为空,在组合树填充阶段才会使用。

DAG 模型中的服务节点由 3 个域组成: D 域存放服务节点相关信息, f_1 指向以该节点为弧头的第一个弧节点, f_2 指向以该节点作为弧尾的第一个弧节点。弧节点由 5 个域组成:尾域 t 和头域 h 分别指向弧尾和弧头这两个顶点在 DAG 图中的位置, l_1 指向与弧头相同的下一条弧, l_2 指向与弧尾相同的下一条弧, i 中存放此弧原子服务的连接信息。

组合树生成算法采用类似深度搜索遍历的方式进行,具体过程如下。

输入:知识模型 DAG

输出:判定弧队列 q ,组合树根节点 r ,树节点遍历顺序 t

Step1 取 DAG 中的起始服务节点为当前节点 n ,进行初始化,并建立服务节点遍历顺序 s ;

Step2 递归处理当前节点 n 的输出弧直至空;若弧头节点已在 s 中,则将该弧加入 q ,否则将弧头节点加入 s ,并将其作为当前父节点的一个子节点;

Step3 递归处理当前节点 n 的输入弧直至空,方法同上;

Step4 依次取出当前父节点的所有子节点,重复执行 Step2 至 Step3;

Step5 返回根节点 r ,判定弧队列 q 与树节点遍历顺序 t 。

该算法生成了一个判定弧队列 q ,存放所有判定弧,同时生成了一个树节点遍历顺序 t ,将在下一阶段使用。 t 的目的在于不必再对树进行多次遍历,节省程序运行开销,有助于统计所有的组合子图。

(2)组合树填充。组合树填充的思想是:在组合树生成之后,填充满这棵树的所有节点并且完成所有判定弧对应树节点的连接;如果所有判定弧都可以连接,那么将判定弧加入到组合树上,就组成了一个组合子图。

生成组合子图的过程分为两个步骤:组合树的生长和组合树的判定。

组合树的生长步骤是对树节点中的候选服务链表进行填充。因为无断枝图中所有节点都不是断枝节点,那么当树中一个父节点被非断枝节点(即候选原子服务)填充后,由于父节点的孩子节点还没有填充,所以一定能够通过增长弧找到对应该节点的连接节点,为每一个孩子节点进行填充。循环这一过程直到树中所有节点都被填充。

组合树的判定步骤是判断判定弧对应的节点是否互相连接,只有所有判定弧都满足连接后才能生成组合子图,即服务组合实例。判定的过程是:从判定弧队列 q 头部开始依次对判定弧进行判定,当弧两端节点中的候选服务无法连接时,更换弧尾节点中的候选服务,并且清空组合树遍历顺序中处于该弧尾节点之后的树节点,重新进行组合树增长,直到队列 q 中所有弧都得到判定,此时树中所有原子服务就能组合成一个组合实例。

4 结 语

本文提出了一种基于知识的服务组合方法,利用服务的本体描述信息进行相邻原子服务连接判断,根据知识中的有向无环图进行服务组合,能够有效地控制使用原子服务的数量和检测范围。该算法首先在服务注册中心查找所需原子服务,然后通过原子服务之间的可连接性以及知识模型,生成服务组合子图,接着使用原子服务的绑定信息对服务依次调用,同时对服务间传递的输入输出消息进行装配,获得服务组合的装配结果。

我们以 XML 文本相似性计算和 XML 文本聚类为应用背景测试了本文所提算法,取得了成功。以后我们将结合更多的具体应用背景,在实践中发挥服务组合灵活高效的软件生产模式,进一步根据云计算的特点实现高性能智能化的软件服务体系。

参考文献:

- [1] PAPAOGLOU M, HEUVEL W J. Service oriented architectures: approaches, technologies and research issues [J]. The VLDB Journal, 2007,16(3):389-415.
- [2] STAAB S, AALST W, BENJAMINS V R, et al. Web services:been there,done that? [J]. IEEE Intelligent Systems, 2003,18(1):72-85.
- [3] BAO J P, TAO B, SU J, et al. Data mining service architecture:an exploration on self-organizing software [C]//Proceedings of the 7th ICSOFT. Rome, Italy: SciTePress, 2012:463-467.
- [4] KHALAF R, MUKHI N, WEERAWARANA S. Service-oriented composition in BPEL4WS [C]//Proceedings of the WWW2003. New York, USA: ACM, 2003:680-687.
- [5] TANG X, JIANG C, ZHOU M. Automatic Web service composition based on Horn clauses and Petri nets [J]. Expert Systems with Applications, 2011, 38 (10): 13024-13031.
- [6] HUANGFU X, SHU Z, CHEN H H, et al. Research on dynamic service composition based on object Petri net for the networked information system [C]//Proceedings of the Fifth International Joint Conference on INC,IMS and IDC. Piscataway, NJ, USA: IEEE, 2009:1075-1080.
- [7] 廖军,谭浩,刘锦德. 基于 Pi-演算的 Web 服务组合的描述和验证 [J]. 计算机学报, 2005,28(4):635-643.

LIAO Jun, TAN Hao, LIU Jinde. Describing and ver-

(下转第 67 页)