

面向组合的 Web 服务间数据关联定位方法研究

马炳先 崔纪鹏 张正明

(山东省网络环境智能计算技术重点实验室 济南 250022) (济南大学信息科学与工程学院 济南 250022)

摘要 Petri 网因其在描述和分析并发、分布式系统方面的优势,越来越多地被用于 Web 服务组合中。在基于 Petri 网描述的 Web 服务组合中,如何确定各个子服务参数之间的关联关系,并以此为基础得到 Web 服务组合的 Petri 网,是对 Web 服务组合性质进行分析的前提与基础。首先给出 Web 服务的 Petri 网描述和基于 Petri 网的 Web 服务注册系统,定义了 Web 服务参数间的数据关联及定位算法,并通过例子说明数据关联在自动生成 Web 组合服务 Petri 网中的应用。

关键词 Petri 网, Web 服务, 组合, 数据关联

中图分类号 文献标识码 A **DOI** 10.11896/j.issn.1002-137X.2014.08.029

Research on Data Association Location Method for Web Services Composition

MA Bing-xian CUI Ji-peng ZHANG Zheng-ming

(Shandong Provincial Key Laboratory of Network Based Intelligent Computing, Jinan 250022, China)

(School of Information Science and Engineering, University of Jinan, Jinan 250022, China)

Abstract Petri net plays an increasingly important role in Web service composition due to its advantages at the description of concurrency and distribution system. To study Web service composition using Petri net, it is crucial to assure the association between parameters of single Web services. This paper first introduced the description of Web services using Petri net, as well as a Petri-net-based register system. Further more, the conception of data association was defined to determine the association between Web service arguments. Using data association, fusible places can be found and Petri net for Web service composition can be established by automatic sharing synthesis of Petri nets. Finally, an applying example was given about the usage of data association.

Keywords Petri net, Web service, Composition, Data association

1 引言

Web 服务以其松耦合、平台无关和开放性等特点有效解决了平台异构性和互操作等问题,允许异构软件和应用系统之间的互联互通,已经成为一种新型的分布式计算模型和互联网应用模式。显然,单一 Web 服务一般不会开发得很复杂^[1],所提供的功能相对比较简单,易于调用。如果要实现某一较为复杂的功能,则需要通过多个 Web 服务间的交互协作,即 Web 服务组合。Web 服务组合将多个简单的 Web 服务按照业务流程逻辑组合起来,从而提供更强大、更完整的业务功能^[2],因此成为当前工业界和学术界研究的一个热点问题^[3-5]。

近年来 Petri 网因其在描述和分析并发、分布式系统方面的优势而广泛应用于 Web 服务组合问题的研究中。例如,文献[6]针对语义 Web 服务的自动组合问题,提出了基于 Web 服务的参数和行为约束上的自动组合,即通过将注册服务转化为一组 Horn 子句,用 Petri 网对 Horn 子句集建模的方法,

利用 T-不变量判断是否存在满足用户需求的组合服务;文献[7]对有色 Petri 网模型在 Web 服务组合中的性质进行了研究,给出了服务组合运算的性质以及执行算法;文献[8]主要讨论了具有多个动态描述逻辑动作的 Petri 网描述和分析方法;文献[9]利用 Petri 网对 Web 服务组合的故障问题进行了分析,给出了组合过程中故障处理的框架。

显然,若要实现 Petri 网技术在 Web 服务组合领域中的实际应用,不能停留在目前仅仅依靠手动绘制 Web 服务组合 Petri 网来进行相应的问题分析,需要进一步解决 Web 服务组合 Petri 网的自动生成问题,从而将 Petri 网真正嵌入到实际系统中。当前已有工作多是在假定已得到 Web 服务组合的 Petri 网的基础上开展相关的性质分析及性能评价,而较少关注 Web 服务组合 Petri 网的自动生成方法。基于此,我们在前期的研究工作中对 Web 服务组合 Petri 网的自动生成方法及其实现进行了相关研究。在文献[10]中针对 Web 服务与 Petri 网的建模问题,提出了在发布各个 Web 服务的时候即创建其相应 Petri 网的方法,对 Petri 网与 Web 服务的衔接

到稿日期:2013-06-19 返修日期:2013-07-18 本文受国家自然科学基金(60903099),山东省优秀中青年科学家奖励基金(BS2009DX012),山东省高等学校科技计划项目(J09LG14)资助。

马炳先(1977—),男,博士,副教授,硕士生导师,主要研究领域为 Petri 网、服务计算;崔纪鹏(1985—),男,硕士生,主要研究领域为服务计算;张正明(1986—),男,硕士生,主要研究领域为服务计算。

问题从系统实现的角度进行了初步研究;同时提出了 Web 服务组合的 Petri 网自动生成方法^[11],在假定已得到各个子服务之间输入/输出关系的基础上,利用 Petri 网的共享合成技术将各子服务的 Petri 网模型进行集成,进一步得到 Web 服务组合的 Petri 网模型。但文献[11]中并未深入探讨如何确定各个子服务间输入/输出的对应关系的确定方法,针对这一问题,本文提出了 Web 服务间数据关联的概念及相关算法,其能够精确描述、定位各子服务的 Petri 网中库所元素间的数据关联关系,确定可融合库所元素,为进一步实现各子服务的 Petri 网间的共享合成操作及 Web 服务组合 Petri 网的自动生成提供依据及基础。

本文第 2 节简要介绍 Web 服务的 Petri 网描述及其注册系统;第 3 节具体讨论 Web 服务参数间的数据关联概念定位算法;第 4 节通过一个例子说明本文的方法;最后对本文工作进行总结并指出进一步工作。

2 Web 服务的 Petri 网描述及注册系统

2.1 Web 服务组合的 Petri 网

Web 服务组合中各子 Web 服务通过它们的输入/输出进行交互及协作来共同满足较为复杂的用户需求。为了有效利用 Petri 网相关方法对 Web 服务组合的服务规划、服务组合故障分析问题进行分析,我们给出 Web 服务组合的 Petri 网的定义如下^[11]:

定义 1 一个 Web 服务组合的 Petri 网模型是一个 5 元组 $\Sigma = (S, T; F, M, L)$, 其中:

- 1) S 为库所元素集合,一个库所可对应于 Web 服务组合中某个子服务的一个输入、输出、前置条件或执行效应;
- 2) T 为变迁元素集合,一个变迁元素代表 Web 服务组合中一个原子服务或服务编制类组合 Web 服务;
- 3) $F \subseteq (S \times T) \cup (T \times S)$ 为流关系集合;
- 4) $M: S \rightarrow \{0, 1, 2, \dots\}$ 是库所元素对应的标识函数;
- 5) $L: S \rightarrow D \cup \{\tau\}$ 为库所元素的语义标记函数,设定 Web 服务组合中各子 Web 服务定义中涉及到的本体为同一个领域本体, D 是该领域本体中类及个体的集合, τ 为空语义标记。

2.2 Web 服务的 PNML+OWL 描述

PNML(Petri Net Markup Language)^[12]是基于 XML 的 Petri 网的交换格式。通过 PNTD(Petri Net Type Definition)来定义 Petri 网的类型,PNML 支持任何一种类型的 Petri 网。PNML 主要用来在不同的工具间实现 Petri 网模型的共享问题。OWL(Web Ontology Language)是一种本体描述语言,它是以描述逻辑为理论基础构建的语言系统,继承了 RDF(Resource Description Framework)基本事实的陈述方式以及 RDFS(RDF Schema)类和属性的分层结构,并在此基础上进行扩展,加入了许多新词汇,克服了 RDF/RDFS 对概念、属性之间关系描述能力弱的问题。

PNML+OWL 是一个具有语义标注的 Petri 网描述文件,它将 PNML 文件与 OWL 语言结合在一起,一个原子服务的 PNML+OWL 描述包括如下 3 部分内容^[11]:

- (1)头文件部分:包括说明服务对应的 XML 版本信息、领域本体、命名空间;
- (2)PNML 标记部分:原子服务的 Petri 网结构描述,其

中库所、变迁及弧(Petri 网中存在的流关系)元素各自拥有相应的 PNML 标签;

- (3)OWL 标记部分:原子服务的 Petri 网中各库所元素的语义标签,即该库所元素对应的服务结构元素在领域本体中的定义,可为类或个体。

Web 服务的 PNML+OWL 描述是通过服务注册系统来实现的。服务提供者注册的每一个服务都会在系统中对应于唯一一个 PNML+OWL 文档,作为对该服务的唯一确定描述。

2.3 Web 服务注册系统

Web 服务应用的整个过程中包含 3 个部分:服务提供者(Service Provider)、服务注册中心(Service Registry)以及服务请求者(Service Requestor)。服务注册中心作为服务提供者和服务请求者之间的桥梁,是 Web 服务中重要的一部分,直接关系到 Web 服务的发现、匹配及调用。为了实现基于 Petri 网的 Web 服务组合,我们设计开发了 Web 服务注册系统(见图 1),主要包括 Web 服务注册端和 Web 服务存储端。

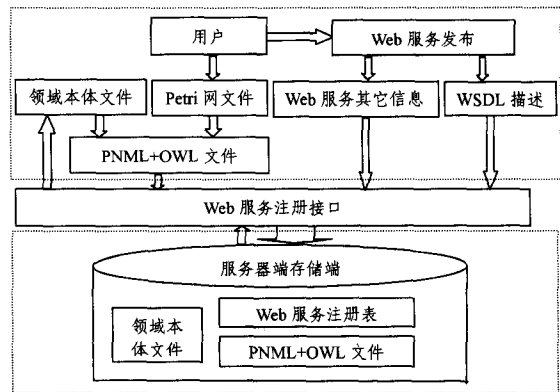


图 1 Web 服务注册系统示意图

Web 服务注册端对服务提供者发布的服务进行描述,对服务参数添加语义,构建、生成相应的 PNML+OWL 文件^[13]。

Web 服务存储端包含 3 部分:Web 服务的领域本体文件、Web 服务的注册表和 PNML+OWL 文件。服务的领域本体不仅对所有服务的参数名称提供了参考,而且还为各个参数赋予了系统能够推理的语义;服务的注册表存储着除内部结构以外的所有服务信息,为 Web 服务的发现、选择、调用等提供信息;Web 服务组合、规划、检测等所需要的信息则存储在 PNML+OWL 文件中。

Web 服务注册系统要求服务提供商在进行服务注册时,必须按要求提供以下信息:

- (1)服务所述领域信息:用关键字 Category 描述,该信息是系统预先定义的,服务注册者只能选择使用现有的 Category,不能够自定义新的 Category;
- (2)服务的功能信息:用关键字 Function 描述,该信息也是系统预定义的,服务注册者只能选择使用现有的 Function,不能够自定义新的 Function;
- (3)服务的参数信息:包括服务调用过程中的参数类型和参数的语义信息。

基于在服务注册过程中对服务的领域 Category、服务功能 Function 和服务参数的类型信息和语义信息的标准化约束,注册成功的 Web 服务成为按功能聚合的统一服务资源视

图,可以按照服务领域、服务功能等信息进行检索,而且不再有任何模糊性和不确定性。

在基于 Petri 网的 Web 服务组合中,根据文献[11]的 Web 服务组合 Petri 网自动生成方法得到服务组合的 Petri 网模型是其中关键的一步,而自动生成 Web 服务组合 Petri 网模型,需要首先依据构成服务组合的各个子服务输入、输出参数之间的语义关系,确定各可融合库所对,通过相应 Petri 网之间的共享合成操作来实现,为此我们给出 Web 服务输入、输出参数之间数据关联的概念。

3 数据关联

3.1 数据关联模型

首先从 Web 服务输入、输出参数的角度给出 Web 服务的描述模型。

定义 2 一个 Web 服务可定义为一个 5 元组 $WS = \{N, U, f, B, P\}$, 其中:

- (1) N 是 Web 服务的输入变量集合,包含 Web 服务中所有的输入参数;
- (2) U 是 Web 服务的输出变量集合,包含 Web 服务中的所有输出参数;
- (3) f 是 Web 服务中的传递变量集合,包含 Web 服务中其他的参数;
- (4) B 是 Web 服务中的行为动作集合,是一个抽象集合,包含 Web 服务中所有的操作行为;
- (5) P 是 Web 服务的端口集合,含有访问该服务的端口信息。

在 Web 服务注册时,系统为服务的各个参数赋予了明确的语义信息,以便可以通过这些语义信息来确定各服务参数之间的语义关系。在具体实现中,为服务的每个参数名称分配一个参数类型 ID,类型 ID 与领域本体中的概念一一对应,一个参数类型 ID 中可包括多个参数名称,如此既可以通过参数名称查到其对应的参数类型 ID,也可以通过参数类型 ID 找到其对应的参数名称。用 $L(p)$ 表示参数 p 的类型 ID 在领域本体中对应的概念,做如下约定:

- (1) 如果 $L(p_1)$ 和 $L(p_2)$ 是相同个体或为等价类,那么称参数 p_1 和参数 p_2 为相同关系,记作 $p_1 = p_2$ 。
- (2) 如果 $L(p_1)$ 和 $L(p_2)$ 是本体中的类,且其中一个是另一个的子类,那么称参数 p_1 和 p_2 为相似关系,记作 $p_1 \approx p_2$ 。
- (3) 如果 $L(p_1)$ 和 $L(p_2)$ 一个是本体中的类,另一个是该类的一个实例,那么同样称参数 p_1 和 p_2 为相似关系,记作 $p_1 \approx p_2$ 。

基于定义 2 和上述约定,下面给出数据关联的定义:

定义 3 对于两个不同的 Web 服务, $WS_1 = \{N_1, U_1, f_1, B_1, P_1\}$ 和 $WS_2 = \{N_2, U_2, f_2, B_2, P_2\}$, 如果 $\exists p_1 \in (N_1 \cup U_1), \exists p_2 \in (N_2 \cup U_2)$, 使得 $p_1 = p_2$ (或 $p_1 \approx p_2$) 成立,那么称 p_1 和 p_2 为数据关联。

当 $p_1 = p_2$ 时称为相等关系,当 $p_1 \approx p_2$ 时称为相似关系。

由定义 3 可知,按照数据的输入输出的类型,数据关联可分为 4 种情况:

- (1) 在服务 WS_1 和服务 WS_2 中,如果 $\exists p_1 \in N_1, \exists p_2 \in U_2$, 使得 $p_1 = p_2$ (或 $p_1 \approx p_2$) 成立,那么称 p_1 和 p_2 为输入/输出关联;
- (2) 在服务 WS_1 和服务 WS_2 中,如果 $\exists p_1 \in U_1, \exists p_2 \in$

N_2 , 使得 $p_1 = p_2$ (或 $p_1 \approx p_2$) 成立,那么称 p_1 和 p_2 为输出/输入关联;

(3) 在服务 WS_1 和服务 WS_2 中,如果 $\exists p_1 \in N_1, \exists p_2 \in N_2$, 使得 $p_1 = p_2$ (或 $p_1 \approx p_2$) 成立,那么称 p_1 和 p_2 为输入/输入关联;

(4) 在服务 WS_1 和服务 WS_2 中,如果 $\exists p_1 \in U_1, \exists p_2 \in U_2$, 使得 $p_1 = p_2$ (或 $p_1 \approx p_2$) 成立,那么称 p_1 和 p_2 为输出/输出关联。

对于多个 Web 服务的情况,基于定义 3 的描述,可能存在这样的情形:

定义 4 对于 $k(k \geq 2)$ 个服务 $WS_i = \{N_i, U_i, f_i, B_i, P_i\}$, $i = 1, 2, \dots, k$, 若存在以下情形:

(1) $\exists p_i \in U_i, \exists p'_{i+1} \in N_{i+1}$, 并且满足 $p_i = p'_{i+1}$ 或 $p_i \approx p'_{i+1} (i = 1, 2, \dots, k-1)$;

(2) 对于 $p'_1 \in N_1, \exists p_k \in U_k$, 并且满足 $p'_1 = p_k$ 或 $p'_1 \approx p_k$;

则称这 k 个服务构成回路服务。

在实际的 Web 服务组合过程中,当多个 Web 服务构成回路服务的时候,每一个服务都需要以回路中前一个服务的执行结果作为输入(或部分输入),这样多个服务就会因得不到相应的输入参数而陷入等待。这种情况在基于 Petri 网的服务组合系统中是不允许存在的,因此如果遇到回路服务,将意味着此次服务组合失败;需要返回回路服务的信息,并寻找可替换服务策略。

参数类型之间隐含了数据关联关系,为了方便使用这些隐含的信息,下面定义数据关联表的概念。

定义 5 假设服务注册时形成的服务参数类型 ID 集合为 $P \subset N^+$, 集合 P 上的二元关系 R 表示两个 Web 服务参数类型之间具有数据关联关系;数据关联表是记录数据关联关系信息的二维表,它是由数据关联首项 ID 和数据关联次项 ID 构成。

对于整数 $i \in P$, 如果 $\exists j \in P$, 使得 iRj (其中 $i \neq j$) 成立, 那么整数 i 存入数据关联首项 ID, 所有满足上述条件的整数 j 的集合存入对应于首项的数据关联次项 ID。

表 1 数据关联表例子

数据关联首项 ID	数据关联次项 ID
2	1; 4; 3
4	2;
3	2;
1	2;

表 1 给出了数据关联表的一个例子,可以看出,参数类型 ID 为 2 的参数与参数类型 ID 为 1、4 和 3 的参数之间存在数据关联。

生成数据关联表的过程如下:

第 1 步 在创建 Web 服务的领域本体时,为其中的每个参数名称分配一个参数类型 ID;

第 2 步 根据领域本体的概念相似度推理,确定参数类型间的数据关联。

3.2 数据关联定位算法

建立数据关联表之后,就可以基于数据关联表的信息,定位 Web 服务组合中各服务输入、输出参数之间的数据关联。实现数据关联定位算法如下:

初始化计数器 $k=1$;

- Step 1 生成数据关联表;
- Step 2 输入构成 Web 服务组合的子服务 ID 集合 P;
- Step 3 通过 PNML+OWL 解析模块,解析子服务 ID 得到子服务的参数类型 ID;
- Step 4 将解析后得到的参数类型 ID 发送到数据关联匹配模块;
- Step 5 把第 k 个子服务对应的参数类型 ID 与数据关联表的首项 ID 逐一匹配,填写数据关联表的次项;
- Step 6 根据回路服务的定义,判断是否形成回路服务,若是则返回形成回路服务的 Web 服务信息,算法结束,此次服务组合不成功,转而寻找回路服务的可替换服务;
- 若 $k=|P|$,表明所有子服务参数已匹配完毕,执行 Step 7;
- 否则 $k=k+1$,执行 Step 5;
- Step 7 返回数据关联匹配结果。

该算法中:(1)PNML+OWL 解析模块以子服务 ID 作为输入,通过解析返回该子服务对应的参数类型信息,这些参数类型信息将用于数据关联匹配;(2)数据关联匹配模块完成数据关联匹配,它以 PNML+OWL 解析模块返回的服务参数类型 ID 作为输入,根据算法开始运行时生成的数据关联表,逐项匹配数据关联表的首项,直至发现回路服务或所有子服务参数匹配完成为止。

4 例子

假设在某一次 Web 服务组合中,组合服务是由 7 个不同功能的子服务所构成;这些子服务在注册后拥有相同的 Category 属性,但 Function 属性各不相同,表示属于同一个领域、同时提供不同功能的服务,它们相互协作能够实现用户要求的复杂服务功能。

首先,在系统运行时,根据服务注册参数信息和领域本体库生成数据关联表。其中数据关联首项 ID 和数据关联次项 ID 都是在服务注册时系统为服务的每个参数名称所分配的整型数字,该数字与领域本体中的参数名称是一一对应的。所生成的数据关联表部分结构如表 2 所列。

表 2 领域本体的数据关联表(部分)

数据关联首项 ID	数据关联次项 ID
1	2;4
3	7;
6	8;
10	13;
11	15;

Service_ID 是一个构成服务组合的子服务 ID 序列,包含了这些子服务在注册服务集中的 ID 号。在此次 Web 服务组合中,服务发现模块返回代表 7 个子服务 ID 的字符串如下:

Service_ID="1;3;4;5;6;7;8";

该服务组合由服务 ID 为 1,3,4,5,6,7,8 的子服务组合而成。

将 Service_ID 送到 PNML+OWL 解析模块可得到各子服务参数类型 ID,PNML+OWL 解析模块以字符串的形式返回各个子服务输入输出所对应的参数类型 ID 如下:

service_parameter="t1:1;2;3;/t2:4;5;6;/t3:7;8;9;/t4:10;11;12;/t5:9;13;/t6:9;14;/t7:13;15;/t8:11;16"

其中 t_i 表示第 i 个服务,冒号后面的数字代表该服务输入对应的参数类型 ID,例如 $t_1:1;2;3$ 表示服务 t_1 的 3 个参数对应的参数类型 ID 分别为 1,2,3。

将所得到的 service_parameter 送入服务数据关联匹配模

块,该模块基于已生成的领域本体数据关联表,对各子服务参数进行数据关联匹配。每个新输入的子服务在完成自身所有参数匹配后,系统都要进行一次回路服务检测,判断是否形成回路服务,如果形成回路服务则结束匹配算法,返回相关信息。若不存在形成回路服务的情况,最后得到的数据关联匹配结果如图 2 所示。

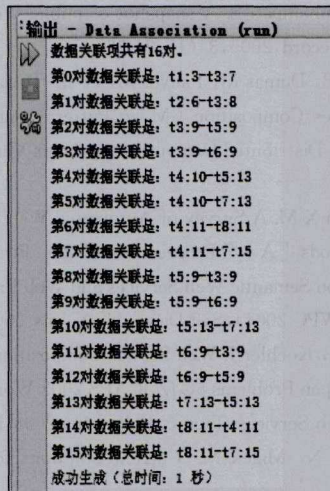


图 2 数据关联结果

图 2 中 $t_1:3-t_3:7$ 表示服务 t_1 中参数类型 ID 为 3 的参数与服务 3 中的参数类型 ID 为 7 的参数之间的数据关联,说明 t_1 中参数类型 ID 为 3 的参数对应的库所元素 p_3 与服务 3 中的参数类型 ID 为 7 的参数对应的库所元素 p_7 是可融合库所对,进一步将 p_3 与 p_7 通过共享合成操作进行融合,得到服务 t_1 与 t_3 组合后的 Petri 网,如图 3 所示,其中,库所 P_{fusion} 表示将 p_3 与 p_7 融合后的库所。

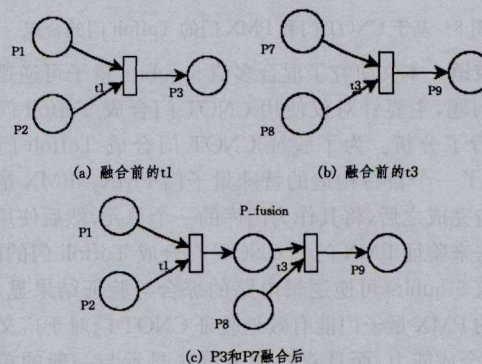


图 3 库所融合示意图

结束语 本文首先给出了 Web 服务的 Petri 网描述,并介绍了基于 Petri 网描述的 Web 服务注册系统,注册后的 Web 服务将会在领域描述、功能描述、参数类型描述等方面得到约束,每一个注册的 Web 服务将会唯一对应于系统生成的 PNML+OWL 文档。在此基础上,针对基于 Petri 网的 Web 服务组合问题,定义了 Web 服务参数间数据关联的概念及服务参数间数据关联的定位算法,并用例子简要说明了数据关联在 Web 服务组合的 Petri 网自动生成中的应用。本文方法的意义在于考虑 Web 服务组合的 Petri 网的生成技术,对于并发性质的应用在于基于服务组合的 Petri 网系统进行服务规划和组合服务动态调度等后续工作。

正在进行及进一步的工作包括:(1)目前数据关联表的生

成还需要部分人为的参与,如果 Web 服务领域本体较大,会给数据关联表的维护带来较大的困难,如何实现数据关联表的自动生成需要进一步的研究;(2)基于 Web 服务组合的 Petri 网,研究 Web 服务组合的相关结构性及服务规划等。

参考文献

[1] Fan J, Kambhampati S. A snapshot of public Web services[J]. SIGMOD Record, 2005, 34(1): 24-32

[2] Benatallah B, Dumas M, Fauvet M, et al. Towards Patterns of Web Services Composition [M] // Patterns and Skeletons for Parallel and Distributed Computing. Springer Verlag, UK, 2003; 265-296

[3] Rao J H, Su X M. A Survey of Automated Web Service Composition Methods [A] // Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004). San Diego, USA, July 2004; 43-54

[4] Srivastava B, Koehler J. Web Service Composition-Current Solutions and Open Problems [C] // ICAPS 2003 Workshop on Planning for Web Services. Trento, Italy, June 2003; 51-59

[5] Milano Vic N, Malek M. Current Solutions for Web Service

Composition [J]. IEEE Internet Computing, 2004, 8(6): 51-59

[6] 汤宪飞, 蒋昌俊, 丁志军, 等. 基于 Petri 网的语义 Web 服务自动组合方法[J]. 软件学报, 2007, 8(12): 2991-3000

[7] 郭玉彬, 杜玉越, 奚建清. Web 服务组合的有色网模型及运算性质[J]. 计算机学报, 2006, 29(7): 1067-1075

[8] 马炳先, 徐颖蕾. 动态描述逻辑动作间关系的 Petri 网分析方法研究[J]. 自动化学报, 2007, 33(11): 1144-1149

[9] 范贵生, 虞慧群, 陈丽琼, 等. 基于 Petri 网的服务组合故障诊断与处理[J]. 软件学报, 2010, 21(2): 231-247

[10] 张正明, 马炳先, 相东明. 基于 Petri 网的 Web 服务的创建与描述[J]. 系统仿真学报, 2011, 23(s1): 19-25

[11] 马炳先, 相东明, 张正明. Web 服务组合的 Petri 网自动生成方法[J]. Journal of Chinese Computer System, 2013, 34(2)

[12] Billington J, Christensen S, van Hee K, et al. The Petri Net Markup Language: Concepts, technology, and tools [C] // van der Aalst W, Best E, eds. 24th International Conference Application and Theory of Petri Nets 2003, LNCS 2679. Berlin: Springer, 2003, 6: 483-505

[13] 张正明, 马炳先, 相东明. 基于 Petri 网的 Web 服务注册方法的研究与实现[J]. 电信科学, 2012, 11(28): 86-91

(上接第 117 页)

子门使用,再利用一般的量子合成算法进行合成。这里我们使用了双向搜索算法,这一点在前面已经进行了介绍。最后得到 Toffoli 门的 CNOT 合成电路,如图 8 所示,其中 PMX 指在前面构造的 PMX 量子逻辑门电路。

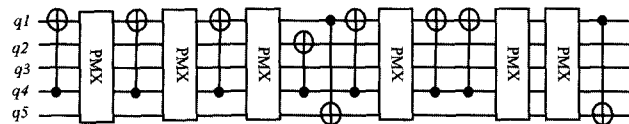


图 8 基于 CNOT 门和 PMX 门的 Toffoli 门的合成

结束语 本文研究了混合多值 5-qubits 量子可逆逻辑电路综合问题,主要针对仅使用 CNOT 门合成 Toffoli 门的可行性进行了分析。为了验证 CNOT 门合成 Toffoli 门的能力,提出了一个自行构造的特殊量子门 PMX。PMX 量子逻辑门构造完成之后,将其作为门库的一个单元,然后使用双向搜索算法来验证 PMX 门和 CNOT 门合成 Toffoli 门的能力。混合多值 5-qubits 可逆逻辑电路的综合与验证结果显示,本文构造的 PMX 量子门能有效地验证 CNOT 门对于广义 Toffoli 门的合成能力,而且双向搜索算法显示与一般的单向搜索算法相比,在效率上至少提高了一倍。

参考文献

[1] Fredkin E, Toffoli T. Conservative logic [J]. International Journal of Theoretical Physics, 1982, 21: 219-253

[2] Nielsen M A, Chuang I L. Quantum computation and quantum information [M]. Cambridge, UK: Cambridge University Press, 2000

[3] Landauer R. Irreversibility and heat generation of the computing process[J]. IBM Journal of Research and Development, 1961, 5(3): 183-191

[4] Reck M, Zeilinger A, Bernstein H J, et al. Experimental realization of any discrete unitary operator [J]. Phys. Rev. Lett., 1994, 73(1): 58431

[5] DiVincenzo D P. Two-bit gates are universal for quantum com-

putation [J]. Phys. Rev. A, 1995, 51(2): 1015-1022

[6] Deutsch D, Barenco A, Ekert A. Universality in quantum computation [J]. Mathematical and Physical Sciences, 1995, 449(1937): 669-677

[7] Yang Guo-wu, Song Xiao-yu, Perkowski M, et al. Realization ternary quantum switching networks without ancilla bits[J]. Journal of Physics A: Mathematical and General, 2005, 38(44): 9689-9698

[8] Yang Guo-wu, Xie Fei, Song Xiao-yu, et al. Universality of 2-Qudit Ternary Reversible Gates[J]. Journal of Physics A: Mathematical and General, 2006, 39(24): 7763-7773

[9] Iwama K, Kambayashi Y, Yamashita S. Transformation rules for designing CNOT-based quantum circuits [J]. Proceedings of Design Automation Conference. New Orleans, LA, USA, 2002, 28(4): 419-425

[10] Shende V V, Prasad A K, Markov I L, et al. Synthesis of reversible logic circuits [J]. IEEE Trans on Circuits and Systems-I, 2003, 22(6): 723-729

[11] Miller D M, Maslov D, Gueck G W. Spectral and two-place decomposition techniques in reversible logic [C] // Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems. Tulsa, AR, USA, 2002; 493-496

[12] Song X Y, Yang G W, Perkowski M, et al. Algebraic characteristics of reversible gates [J]. Theory of Computing Systems, 2004, 37(2): 311-319

[13] Maslov D, Gueck G W, Miller D M. Toffoli network synthesis with templates [J]. IEEE Trans on Circuits and Systems-I, 2005, 24(6): 807-817

[14] Yang Guo-wu, Song Xiao-yu, William N, et al. Group theory based synthesis of binary reversible circuits [J]. Lecture Notes in Computer Science, 2006, 3959: 365-374

[15] 李志强, 陈汉武, 徐宝文. 基于 Hash 表的量子可逆逻辑电路综合的快速算法[J]. 计算机研究与发展, 2008, 45(12): 2162-2171

[16] Yang Guo-wu, Song Xiao-yu, Hung W N N, et al. Bi-directional Synthesis of 4-bit reversible Circuits [J]. The Computer Journal, 2008, 51(2): 207-215