

组合服务自动合成技术^{*})

钱海波 钱柱中 陆桑璐 陈道蓄

(南京大学计算机软件新技术国家重点实验室 南京 210093)

(南京大学计算机科学与技术系 南京 210093)

摘要 Web 服务凭借其良好的平台无关性得到了广泛的关注和运用,但单个 Web 服务功能单一,不能满足当今日益复杂多变的服务请求。为了能够快速、低成本地满足复杂的用户需求,Web 服务组合应运而生。本文首先介绍了各类 Web 服务的描述模型,然后基于不同的描述方法,结合近年来 Web 服务组合研究的成果,介绍了 Web 服务自动合成技术,最后对 Web 服务组合的未来发展和研究方向做出了展望。

关键词 Web 服务, Web 服务组合, 语义网络, 智能规划, 自动机, Petri 网

Automatic Synthesis of Composite Service

QIAN Hai-Bo QIAN Zhu-Zhong LU Sang-Lu CHEN Dao-Xu

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract Due to its independency of both platform and language, Web service is widely used these days. As Web services are being created and updated on the fly, finding a way to organize them effectively to satisfy the various requests from users has become urgent and important. This paper introduces modeling techniques of Web services from both industry and academic domain. Based on these modeling techniques, several composition approaches are introduced with recent research work. In the end, an outlook to future research work is presented.

Keywords Web service, Web service composition, Semantic Web, AI planning, Automata machine, Petri net

1 引言

近年来,随着信息技术和互联网的飞速发展,社会各部门信息化程度越来越高。如何在这种信息化大背景下,更加有效地利用信息化的社会各部门已经成为一个重要的课题,电子商务和电子政务应用的兴起就是一个极好的例证。Web 服务模型正是为了解决这些问题而提出的一种面向服务的分布式计算模型^[1~3]。

Web 服务架构通常由三个实体组成,即服务提供者(provider)、服务请求者(requester)和服务目录(registry)。Web 服务的发现和交互由这三个部分协同完成:服务提供者在服务目录中发布(publish)服务,服务请求者根据服务目录查找(find)需要的服务,最后服务请求者根据查找返回的信息使用(bind)Web 服务。同时,Web 服务制定和利用了一系列标准和协议,使其独立于平台和语言,为其在异构环境下的整合提供便利。如用 UDDI(Universal Description, Discovery and Integration)定义服务的发布和查找、用 WSDL(Web Service Description Language)来定义服务的描述、用 SOAP(Simple Object Access Protocol)协议来规定消息交换的格式等^[3]。

对于这样的一个体系架构,人们希望能够高效地发现符合自身需求的服务,尽可能地使用那些可用的服务,而避免花较高代价开发新的服务。然而,那些可用的单个服务往往不足以完成服务请求者的请求,因此需要协同这些服务模块,使

之能够完成复杂的功能需求。为此,工业界提出了一系列的组合服务标准和引擎,用于支持组合 Web 服务。然而,随着 Web 服务应用的日益普及,用效率低下的手工方式对 Web 服务进行整合已经不能满足人们的需要,其主要原因在于^[2]: Web 服务数量繁多;Web 服务动态性强,随时都有 Web 服务被创建和更新;不同组织发布的 Web 服务概念模型也往往有差异。因此,如何能够根据用户需求,将分布于网络上的单个 Web 服务自动、有效地组织起来,即 Web 服务的自动合成,正成为一个迫切需要解决的问题。

本文以 Web 服务的描述为基础,总结和讨论了当前组合 Web 服务合成方面的研究进展和取得的成果。文章结构如下:第 2 节介绍 Web 服务的主要描述模型,在此基础上,第 3 节结合近年来国内外的研究成果,介绍 Web 服务合成的几种方法,最后总结全文,并展望未来 Web 服务组合的研究方向。

2 Web 服务的描述模型

2.1 工业界的描述标准

到目前为止,工业界提出了诸多与 Web 服务相关的标准和语言,用于描述 Web 服务、定义 Web 服务之间的协同和交互等。

WSDL^[4]是 Web 服务最为著名的标准之一,也在事实上成为了一个现行通用的 Web 服务描述语言。WSDL 由类型段、消息段、端口类型段、绑定段和服务段这 5 个部分组成,指

^{*} 本文受江苏省自然科学基金(BK2005411)资助。钱海波 硕士研究生,主要研究方向为 Web 服务组合、分布式计算。钱柱中 博士研究生,主要研究方向为面向服务的计算、分布式计算。陆桑璐 教授,博士生导师,主要研究领域为分布式计算、并行处理。陈道蓄 教授,博士生导师,主要研究领域为分布式计算、并行处理。

定了 Web 服务的输入和输出参数、服务可供使用的方法和服 务在网络上的位置,以及一些有关如何对服务进行访问的说明。但是 WSDL 只描述了 Web 服务的静态接口。为了能够定义 Web 服务的交互行为, BEA 和 SUN 等公司向 W3C 提交了 WSCI(Web Service Choreography Interface)。WSCI 能够描述交互信息的正确顺序,并允许定义同一个 Web 服务基于不同上下文环境中可能会出现的不同行为。

此后, BEA、IBM 和微软又发布了另一份规范,称为 BPEL4WS(Business Process Execution Language for Web Service)^[5]。它是在 IBM 提出的 WSFL(Web Service Flow Language)和微软提出的 XLANG 基础上开发的。通过 BPEL4WS 可以创建不同的活动,然后将这些活动连接起来,从而创建出复杂的流程。BPEL4WS 支持抽象流程,其中的活动通过 PortType 的抽象定义来表示,执行时则通过 Endpoint 将这些抽象的活动绑定到具体的 Web 服务的行为。如今, BPEL4WS 已经成为了主流的组合 Web 服务描述方式,已经有多种商业和开源的引擎(如 BPWS4J、ActiveBPEL 等)能够支持 BPEL4WS。

2.2 学术界的描述模型

由于 WSDL 只描述接口,不支持语义,研究者提出使用语义网络为 Web 服务提供语义支持。语义网络是将信息以便于机器处理的方式连接起来而组成的网络。语义网络通过松耦合的 本体(ontology)将不同领域的信息关联起来,而 Web 服务组合也由松耦合的 Web 服务构成。这种结构上的相似性意味着可以通过使用本体描述 Web 服务,借助语义网络的整合能力,达到 Web 服务自动组合的目的。目前已有多种语言通过这种方式来描述 Web 服务(如 OWL-S、SWIS、WSMO、SESMA 等)。下面以比较常用的,也是 W3C 推荐的 OWL-S^[6]为例介绍一下这种描述方式。

OWL-S 是以 OWL 网络本体语言为基础,结合 Web 服务的具体应用发展而来,其前身是 DAML-S。OWL-S 的目的主要是为了实现 Web 服务的自动发现、Web 服务的自动调用、Web 服务的自动组合和交互以及 Web 服务执行的自动监控。OWL-S 使用三类本体从不同角度描述 Web 服务: Service Profile 描述服务做什么,提供搜索服务主体所需要的信息和服务能力的描述,如服务的简单描述、IOPE(Input, Output, Precondition, Effect)等; Service Model 描述服务具体如何工作,如复杂服务的过程模型; Service Grounding 则描述访问服务的具体细节,如通信协议、消息格式等。

这类描述模型侧重于将 Web 服务以原子服务的形式表达,大部分研究工作的着眼点在于如何利用描述提供的各方面信息(功能、服务质量、服务代价等)进行匹配,寻找合适的 Web 服务进行组合,一般很少关注各个服务组件本身的内部行为。

随着 Web 服务技术的不断发展,人们越来越多地关注服务内部行为之间的关系,于是研究者又提出了 Web 服务的行为表述模型。与语义网络描述模型不同,用行为表述模型进行服务组合时,组合是基于服务的内部行为进行的,服务不再仅仅是一个类似“黑盒”的原子服务,因而这种模型可以支持那些需要多次操作才能完成的服务。这类描述模型一般借助形式化的工具(如自动机、Petri 网等)描述服务。下面分别进行说明。

如果将内部行为考虑在内, Web 服务的执行实际上是一组操作按照一定流程按序执行的结果。一方面,对于不同的

用户请求,可能会有不同的执行流程;另一方面,有些 Web 服务可以在执行过程中与用户进行交互,这也可能导致流程的改变。如果在服务的每个动作执行完后引入一个与已执行流程相关的状态,而将可能的用户输入以及服务执行结果作为转移函数的参数,就可以形成一个与该服务对应的自动机,自动机的每一个状态都表示服务所处的某一个特定状态,这样服务组合就转化为对自动机进行组合的问题。

Petri 网也可以用来描述 Web 服务^[24,25],可以用 Petri 网中的资源代表消息,变迁代表 Web 服务的行为(操作)。Petri 网模型能够表述组合 Web 服务的流程逻辑,支持服务模块的并发执行。另外,基于 Petri 网的特性,这样的模型表现出了 Web 服务构架中的消息驱动行为的基本特征,因此能够同时描述组合 Web 服务的信息流和控制流。

此外,通过会话(conversation),即 Web 服务组件之间的消息传递,也可以对服务及其行为进行描述。使用会话描述服务时,通常会有一个全局的观察者,它能够从全局观察到各个服务组件之间的消息传递。服务组件产生消息,预示着其内部状态的某种改变,这是会话能够反映服务行为的原因。通过研究观察者观察到的全局消息模式与各个服务组件的局部消息模式之间的关系,可以指导组合的进行。文^[22]对此进行了比较深入的研究。

3 Web 服务自动组合

通常 Web 服务组合要解决的问题如下:某个用户请求不能用现有的任何一个 Web 服务来满足,但是经过将某些现有的 Web 服务通过某种方式组合起来以后却可以满足这个请求。那么,如何发现这种满足请求的组合流程呢?方法可以分为静态(手工)组合和动态组合两种。静态组合是指在设计阶段手工进行服务流程的编制,按照流程手工挑选合适的服务,并将服务连接、部署起来,构成一个服务组合。这种方法只适用于那些服务流程、服务组件都很少变化的情况,其扩展性比较差。本文主要介绍动态组合方法,即能够自动完成组合服务流程的制定或者服务的选择的一类方法。

如图 1 所示,通常一个动态服务组合系统可以分为如下几个部分(虚框):服务组合引擎、知识库或计划模版、执行引擎、服务代理。

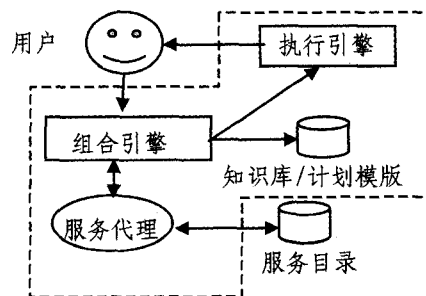


图 1 一种动态服务组合系统

系统工作时,用户请求被发送到服务组合引擎,组合引擎利用知识库推理或者直接根据计划模版得到一个能够满足用户请求的组合流程,接着组合引擎通过服务代理从服务目录选择合适的服务进行组合,随后组合引擎将组合以某种形式(如 BPEL4WS)交给执行引擎绑定执行,最后将得到的结果返回给用户。整个组合执行过程对用户来说是透明的,因此可以将服务组合系统看作一个中介代理系统(mediator)。下

面介绍一些具体的组合技术。

3.1 基于原子服务的自动组合方法

3.1.1 工作流的方法

工作流是一系列相互衔接、自动进行的业务活动或任务。一个工作流包括一组任务以及它们之间的相互顺序关系,还包括流程及任务的启动和终止条件,以及对每个任务的描述。将 Web 组合服务中的各个服务组件看成工作流中的任务,服务组合就转化为一个构建工作流的过程。用工作流模型构建起来的组合服务可以利用工作流的研究成果(如 Petri 网模型),方便地进行数据一致性、死锁、约束满足等方面的检查。目前基于工作流的 Web 服务组合研究一般都以偏静态的方法为主,即基于事先构建好的抽象过程模型对服务进行动态选择和绑定,但抽象过程模型本身往往具有一定的动态性。

文[8]构建了一个基于工作流的自动服务组合平台 EFlow。EFlow 通过一个过程计划书来描述组合服务(类似工作流),计划书由服务节点、判断节点和事件节点组成的图表示。服务节点可以表示一个基本服务或者组合服务,判断节点提供流程控制的支持,事件节点使服务过程支持事件的接受和发送。EFlow 的服务组合过程分为两步:一是根据用户需求实例化过程计划书,二是根据实例化了的计划书选择服务进行绑定。EFlow 主要考虑的问题有两个方面:一是服务的自适应,以适应服务不断变化的动态环境;二是如何支持对计划书的动态修改,以适应某些特殊的用户需求。EFlow 使用服务选择规则来描述服务节点,从而达到在执行时动态选择服务的目的。同时通过支持运行时的 Ad-hoc 修改(针对个别用户的定制)和计划书实例化阶段的批量修改(针对满足特定条件的请求),来支持计划书的动态修改。类似的系统还有 E-ADOME^[9]等。

这类系统的优点在于工作流的引入简化了组合服务的构建,而且通过一些办法仍可以达到一定动态性,但仍需要事先定义一个相对固定的流程,而且也不能支持那些需要多次调用才能完成的服务。而文[10]中的 PPM(Polymorphic Process Model)模型除了使用类似 EFlow 的机制对基于引用过程的服务进行动态选择和绑定外,对于那些基于多个外部服务的组合(可能需要多次交互),PPM 使用状态机来描述其状态及状态转换,然后使用基于这些状态机的推理来进行动态组合,从一定程度上支持了那些多次调用才能完成的服务。

3.1.2 智能规划的方法

智能规划(AI Planning)在解决 Web 服务组合问题中应用很广泛,与工作流不同,这种方法不需要事先给定服务流程。传统的智能规划问题一般可以描述如下:给定状态集 S 和动作集 A ,定义 S 和 A 上的状态转移关系 $\Gamma \subseteq S \times A \times S$ (知识库),给定一个初始状态 $s_0 \in S$ 以及要达到的目标状态 $S_G \subseteq S$,智能规划的目标就是要找到一个 A 中的一个动作序列,使得 s_0 经过这个动作序列的变换以后到达目标状态 S_G 。OWL-S 描述模型中 Service Profile 所描述的前提条件和效果可以看作是 Web 服务执行前后的两个状态,而 Web 服务本身则是一个动作。因此,智能规划往往使用类似 OWL-S 的语义网络模型作为服务组合的描述模型。

文[11]扩展了 Golog 语言,将 Web 服务组合用 Golog 语言表示成高层抽象的过程,然后利用情景演算(Situation Calculus)来实现 Web 服务的自动发现、运行、组合以及交互。所谓情景演算,指的是将世界的每一个状态作为一个情景(situation),情景的改变由动作(action)引起,其目的在于寻找初始

情景到目标情景的动作序列。其中的情景和动作均由一阶语言表示,此外还有一系列公理作为知识库用于进行推理。系统使用一种类似 DAML-S 的语言来描述其知识库。该工作的主要贡献在于引入了表示用户约束的谓词 Desirable,从而达到减少规划搜索空间、提高搜索效率的目的,但其引入谓词的约束性太强,限制了组合的灵活性。

文[12]是在此基础上的进一步工作,其主要贡献在于弱化了上述 Desirable 谓词,而代之以所谓的偏好(preference),组合并不强制要将这些偏好考虑在内,而只是作为一种更好的选择,这使得组合更加灵活。

文[13,14]描述的是一种基于 HTN(Hierarchical Task Network)规划的 Web 服务组合方法。HTN 规划是一种智能规划的方法,其主要过程在于将任务不断递归的分解成子任务,直到全部成为可以直接执行的原子任务为止。系统使用 SHOP2 作为 HTN 规划的工具。SHOP2 的知识库中存在两类内容:算子(operator)和方法(method)。前者描述的是如何完成一个原子任务,而后者则描述了如何分解一个组合任务。这样,根据知识库中的这两类内容即可完成任务的分解。作者使用 DAML-S 来描述 Web 服务,并给出了将 DAML-S 描述模型转换到 SHOP2 知识域中的详细办法。

在此基础上,文[15]提出了一种基于工作流模板的 Web 服务组合方法以提高其组合效率。作者认为多数商业处理流程的大致步骤都是确定的,因此可以用一个基于工作流的模板对服务组合进行适当的约束,只要模板中描述的抽象程度恰当即可——一方面足够抽象以包含所有相关服务,另一方面明确某些特征以便进行自动组合。为此作者对 OWL-S 进行了扩展,加入了一种名为 AbstractProcess 的抽象过程类型以支持对模板的描述。同时作者还将 HTN 扩展成为 HTN-DL 以支持 AbstractProcess。

使用智能规划实现服务组合还有很多其他工作,如文[16,17]将 Web 服务表示成线性逻辑(Linear Logic)中的公理和证明,使用定理证明的方法达到构建服务组合的目的;文[18]定义了组合需要满足的语法和语义规则,然后利用这些规则进行 Web 服务的选择和组合等。智能规划的组合方法与逻辑、人工智能领域紧密相关,通过推理、搜索等手段可以动态生成具体组合流程,因而比基于工作流的方法更为灵活。但其组合效率普遍比较低,如何提高其组合的效率是一个重要问题。

3.2 基于行为表述的组合方法

无论是基于工作流的还是智能规划的组合方法,Web 服务都被看作一个类似“黑箱”的原子服务,只需要知道其前提条件和执行效果即可对服务进行组合。但是有些 Web 服务可能需要经过一系列相关的调用(或者说行为)才能完成。对于这样的服务,其内部行为不可忽略,需要用行为表述模型表现出各个调用后其内部的不同状态才能对其进行组合。此外,基于行为描述的模型能够较好地支持 Web 服务组合的复用,其主要原因是它能够统一地描述 Web 服务,包括原子服务和组合服务,于是组合服务能够同样地参与更为复杂系统的组合。

文[19]提出可以将 Web 服务根据其不同状态下可进行的操作表示成一棵执行树(通常是无穷的),执行树又分为外部执行树和内部执行树两种。外部执行树指明服务执行流程,确定服务对外输出的行为,而内部执行树表明了一个服务实例如何真正执行每个给定的行为(即行为具体由哪个服务

进行)。作者特别研究了可以表达为有限状态自动机的执行树,并用这些自动机来描述 Web 服务。在服务执行的每一步,用户根据其目标选择一个行为,服务执行这个行为。此后,服务进入下一个状态,于是又有一系列的行为可以被触发。这个模型被称为 Roman 模型,是服务组合中的一个常用模型。基于 Roman 模型的服务组合可以采用 PDL(Proposal Dynamic Logic)技术来实现。初始时,目标服务和给定服务都处于状态图的初始状态。此后,目标服务中的每一次状态转换,对应于给定 Web 服务中的某个服务的一次状态转换。也就是说,目标服务执行的每个行为实际上是由给定 Web 服务中的某个服务实现的。当达到终止状态时,目标服务和所有给定的 Web 服务都处于终止状态。

在文[20,21]中,研究小组继续对此进行了一些扩展。前者考虑了某个服务可能将其某一个行为代理给其他服务去完成的情况,这会组合带来一定的不确定性;而后者进一步考虑了对半可控(Partially controlled)服务的支持,即那些仅可通过输出观察到其状态变化的服务。

但文[19~21]所描述的组合方法缺乏一种有效的数据交换手段。文[22]从会话描述模型的角度考虑 Web 服务组合,作者将 Web 服务看作是带有消息缓冲队列的节点,节点对不同的消息做出不同的反应,并可能向其他节点发送消息;同时,每次某个 Web 服务从中取出一个消息进行处理,都会被一个全局的观察者捕捉到,进而得到一个全局的消息序列。作者认为观察者看到的全局消息序列和各个节点看到的消息序列之间存在着某种联系,利用这种联系可以指导服务组合的进行,或者说通过限制节点之间如何交换消息,可以控制组合的行为。为了研究两者之间的关系,作者用米利机(Mealy Machine,即带有输入输出的有限自动机)描述 Web 服务。对于一个目标 Web 服务,其全局消息序列可以用正则语言来描述,服务组合的任务就是要找到一系列节点,使它们之间通信的整体行为满足该正则语言。但该会话模型的主要工作集中在考查组合服务行为的确认和验证,并没有涉及自动服务组合。

从会话模型和 Roman 模型来看,后者更适合作为自动组合模型,因为消息只是系统行为的外部表征,并不能反映内部情况,而行为才真正反映系统的运行结果。

文[23]结合上述会话模型和 Roman 模型,提出了 Colombo 模型,试图建立一个统一的 Web 服务组合模型。Colombo 采用了状态转换自动机来描述服务过程,定义全局状态记录组合服务系统行为,利用本地栈暂存消息实现服务间的异步通信,同时使用“连接(link)”作为消息的通道,是一个比较完备的 Web 服务模型。但是这样的模型中服务过程只体现了服务之间的消息交互,而消息和行为的联系并没有得到很好的体现,需要通过制定 PDL 规则来确定消息触发的行为。

文[25]提出了基于 Petri 网模型的服务自动组合规则,用于自动合成组合 Web 服务。该方法的基本思想是通过一个共享的库所作为可用 Web 服务消息传递的中介。当一个可用 Web 服务的行为满足触发条件之后,目标服务也触发相同的行为,并将此行为指派为该可用 Web 服务。另外,行为触发后生成的消息又可以通过共享的库所用于其他服务行为的触发。

Petri 网的组合方式体现了消息和行为的关系,能够更为自然地描述信息流和控制流,因而较为简捷和灵活,其组合服

务是以“中介方式(mediated)”提供的,其实施倾向于集中式的。而像 Colombo 这样基于自动机的模型,通过“连接”定义 Web 服务之间的交互,因而组合服务是一种“代理方式(brokered)”提供的。

总结展望 随着近年来信息技术和互联网的飞速发展,Web 服务和 Web 服务组合的应用和研究变得越来越重要,本文的目的在于介绍一些 Web 服务组合的方法。本文首先介绍了 Web 服务的描述模型,认为可以分为两类,即基于语义网络的描述模型和表述行为的描述模型。随后在此基础上,结合国内外近年来的一些研究工作,介绍了几类不同的 Web 服务组合方法及其相关研究,内容涉及 workflow、智能规划、自动机等方面。

Web 服务组合的研究中,语义匹配是一个关键性的问题,任何方法都需要进行服务之间的行为和消息的匹配。随着语义网络研究的不断发展,如今人们越来越关注引入本体的技术来解决这个问题。而 OWL-S 标准的不断发展和补充事实上也是人们在不断努力加强 Web 服务行为和消息的语义。

Web 服务组合今后的研究方向还有以下几个方面:首先是如何提高服务组合的组合效率,特别是基于智能规划的一些组合方法;其次,在 Web 服务组合建立后,如何对组合进行评估和优化也需要进行深入研究。此外,随着无线设备的逐渐普及,由于设备和网络的异构性以及带宽的限制,如何在普适环境中应用 Web 服务组合也是一个很有意义的研究课题,目前我们正在这方面展开一些探索性的研究工作。

参 考 文 献

- Hull R, Benedikt M, Christophides V, et al. E-Services: A Look Behind the Curtain. In: Proceedings of the 22nd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, San Diego, USA, 2003. 1~14
- Rao J, Su X. A Survey of Automated Web Service Composition Methods. In: Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, San Diego, California, USA, 2004. 43~54
- Booth D, et al. Web Services Architecture. 2004. <http://www.w3.org/TR/ws-arch>
- Christensen E, et al. Web Services Description Language (WSDL) 1.1. 2001. <http://www.w3.org/TR/wsdl>
- Andrews T, et al. Business Process Execution Language for Web Services (BPEL4WS), Version 1.1. 2003. <http://www-128.ibm.com/developerworks/library/specification/ws-bpel>
- Martin D, et al. OWL-S 1.1 Release. 2004. <http://www.daml.org/services/owl-s/1.1>
- Hull R, Su J. Tools for Composite Web Services: A Short Overview. ACM SIGMOD Record, 2005, 34(2): 86~95
- Casati F, Ilnicki S, Jin J. Adaptive and dynamic service composition in EFlow. In: Proceedings of 12th International Conference on Advanced Information Systems Engineering, Stockholm, Sweden, 2000. 13~31
- Dickson C, Kamalakar K, Li Qing, et al. Workflow view based e-Contracts in a Cross-Organizational e-Services environment. Distributed and Parallel Databases, 2002, 12(2-3): 193~216
- Schuster H, Georgakopoulos D, Cichocki A, et al. Modeling and composing service-based and reference process-based multi-enterprise processes. In: Proceeding of 12th International Conference

on Advanced Information Systems Engineering, Stockholm, Sweden, 2000. 247~263

11 McIlraith S, Son T. Adapting golog for composition of semantic web services. In: Proceedings of the 8th International Conference on Knowledge Representation and Reasoning, Toulouse, France, 2002. 482~493

12 Sohrabi S, Prokoshyna N, McIlraith S. Web Service Composition via Generic Procedures and Customizing User Preferences. In: 5th International Semantic Web Conference, Athens, USA, 2006. 597~611

13 Sirin E, Parsia B. Planning for Semantic Web Services. In: Semantic Web Services Workshop at the 3rd International Semantic Web Conference, Florida, USA, 2004

14 Sirin E, Parsia B, Wu D, et al. HTN planning for web service composition using SHOP2. Journal of Web Semantics, 2004, 1(4): 377~396

15 Sirin E, Parsia B, Hendler J. Template-based composition of semantic web services. In: AAAI Fall Symposium on Agents and the Semantic Web, Virginia, USA, 2005

16 Rao J, Kungas P, Matskin M. Application of Linear Logic to Web Service Composition. In: Proceedings of the 1st International Conference on Web Services, Las Vegas, USA, 2003. 3~9

17 Rao J, Kungas P, Matskin M. Logic-based Web services composition: from service description to process model. In: Proceedings of the 2004 International Conference on Web Services, San Diego, USA, 2004. 446~453

18 Medjahed B, Bouguettaya A, Elmagarmid A K. Composing Web services on the Semantic Web. The VLDB Journal, 2003, 12(4):

333~351

19 Berardi D, Calvanese D, De Giacomo G, et al. Automatic composition of e-services that export their behavior. In: Proceeding of 1st International Conference on Service Oriented Computing, Trento, Italy, 2003. 43~58

20 Berardi D, De Giacomo G, Lenzerini M, et al. Synthesis of underspecified composite e-services based on automated reasoning. In: Proceeding of 2nd International Conference on Service-Oriented Computing, New York, USA, 2004. 105~114

21 Berardi D, Calvanese D, De Giacomo G, et al. Composition of Services with Nondeterministic Observable Behavior. In: Proceeding of 3rd International Conference on Service Oriented Computing, Amsterdam, USA, 2005. 520~526

22 Bultan T, Fu X, Hull R, et al. Conversation Specification: A New Approach to Design and Analysis of E-Service Composition. In: Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, 2003. 403~410

23 Berardi D, Calvanese D, De Giacomo G, et al. Automatic Composition of Transition-based Semantic Web Services with Messaging. In: Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 2005. 613~624

24 Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition. In: Proceedings of the 14th Australasian Database Conference, Adelaide, Australian, 2003. 191~200

25 Qian Z Z, Lu S L, Xie L. Automatic Composition of Petri Net Based Web Services. Chinese Journal of Computers, 2006, 29(7): 1057~1066

(上接第 127 页)

始点,只需分别扫描 PList'列表和 CList'列表一遍,无须维护 CQueue 队列,即可直接输出匹配结果。

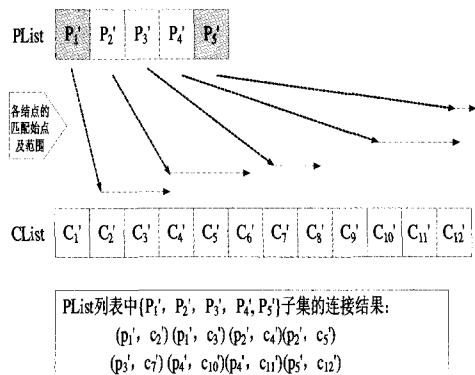


图 5 改进后算法的执行过程

2.3 算法分析及结论

假设改进前双亲列表 PList 的元组数为 m , 孩子列表 CList 的元组数为 n , 改进前算法的存储空间开销为 $(m+n+|CQueue|)$, $|CQueue|$ 表示队列的长度。在最坏情况下, $|CQueue|$ 为 n , 算法的存储空间为 $(m+2n)$; 在最好情况下(即不存在嵌套现象则无需队列), $|CQueue|$ 为 0, 算法的存储空间为 $(m+n)$ 。

又设改进后双亲列表 PList' 的元组数为 m' ($m'=m$), 孩子列表 CList' 的元组数为 n' ($n'=n$), 因为改进后的算法无论在任何情况下都无需维护队列, 所以改进后算法的存储空间开销为 $(m'+n')$ 。也就是说改进后的算法的空间开销总是和

改进前算法的空间开销最好的情况下相等。

即得出结论:改进后的算法更节省内存空间的开销。在上例中改进后的算法就比改进前的算法节省了 7 个存储结点的空间,而且当 XML 树中结点数较多并存在嵌套时,队列 CQueue 的长度与 CList 列表的元组数有关,CList 列表的元组数越多且存在嵌套,队列 CQueue 的长度也就越长,这时算法的空间存储开销也就随 CQueue 的长度的增加而增加。改进后的算法不需维护 CQueue 队列,所以效果更加明显。

参 考 文 献

1 Yoshikawa M, Amagasa T, Shimura T, Uemura S. XRel: A path-based approach to storage and retrieval of XML documents using relational databases. ACM Trans. on Internet Technology, 2001, 1(1): 110~141

2 Jiang H, Lu H, Wang W, et al. XParent: An Efficient RDBMS-Based XML Database System. In: Proceedings of the 18th International Conference on Data Engineering, 2002

3 World Wide Web Consortium. XML Path Language(XPath)1.0. W3C Recommendation. November 1999. http://www.w3.org/TR/xpath

4 Jiang H F, Lu H J, Wang W, Ooi B C. XR-Tree: Indexing XML data for efficient structural joins. In: Dayal U, Ramamritham K, Vijayaraman T M, eds. Proc. of the 19th Int'l Conf. on Data Engineering. Los Alamitos: IEEE Press, 2003. 253~264

5 Fan C, Funderburk J, Lam Hou-in, et al. XTABLES: bridging relational technology and XML. IBM Systems Journal, 2002, 41(4)

6 Al-Khalifa S, Jagadish H V, Koudas N, et al. Structural joins: A primitive for efficient XML query pattern matching. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering. Los Alamitos: IEEE Press, 2002. 141~152