

# 用于科技基础条件平台的服务合成研究与应用

曹会敏<sup>1,2</sup>, 林碧英<sup>1</sup>, 程振林<sup>2</sup>

(1. 华北电力大学 计算机科学与技术学院, 北京 102206;

2. 中国科学院 计算技术研究所, 北京 100080)

**摘要:** Web 服务之间的交互是面向服务的体系结构(Service-Oriented Architecture, SOA, 也称为面向服务架构)中的关键问题。单个 Web 服务提供的功能往往不能完全满足用户的实际需求, 将多个 Web 服务进行合成实现功能更为强大的服务是 SOA 必须要解决的问题之一。分析了 Web 服务合成的研究现状, 介绍了 Web 服务合成方式的分类和商业流程执行语言 BPEL4WS, 给出了“科技基础条件平台应用服务支撑系统”项目中 SOA 的设计方案, 描述了 Web 服务合成的关键步骤。该研究成果成功地应用于科技基础条件平台应用服务集成之中, 并已通过科技部的前期验收。

**关键词:** Web 服务; BPEL4WS; Web 服务合成

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1673-629X(2008)09-0091-04

## Web Service Composition and Its Application in National Science and Technology Infrastructure Platform

CAO Hui-min<sup>1,2</sup>, LIN Bi-ying<sup>1</sup>, CHENG Zhen-lin<sup>2</sup>

(1. School of Computer Science and Technology, North China Electric Power University, Beijing 102206, China;

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

**Abstract:** The interaction among web services is the key problem in service oriented architecture (SOA). Individual web service can not fully meet the demand from users. The composition of services is the essential issue in SOA. After analyzing the current solutions for web service composition, introduces the categories of web service composition and the language of BPEL4WS, presents the schema of SOA for the project of “application and service integration supporting system”, describes some key steps of web service composition, including service discovery, service categories, identification of service granularity, access to services, etc. The research is successfully applied to the project referred and passes the preliminary acceptance by the People's Republic of China Ministry of Science and Technology.

**Key words:** Web service; BPEL4WS; Web service composition

## 0 引言

由于 Web 服务支持跨平台的互操作, 能够访问各类包装成 Web 服务的应用系统, 使之成为企业应用集成框架的一个较好选择。从技术上说, Web 服务的价值在于服务重用, 单个 Web 服务提供的功能毕竟有限, 往往不能满足实际的应用需求, 不能适应开放的、动态的业务需求变化。因此, Web 服务的抽象、组织和合成显得越来越重要, 便于企业内或企业间的诸多业务服务有机地集成以提供功能更强大的服务<sup>[1]</sup>。本项目中在业务编排中采用了基于 BPEL4WS 的 Web 服务

组合技术。

文中主要研究 Web 服务合成在该项目中的应用。项目中的 Web 合成服务是建立在资源的共享与集成基础之上, 涉及到科技基础条件资源涵盖的各个领域, 跨平台、跨领域地构建 Web 合成服务, 为实现科技资源的共享提供更加强大的服务。服务组合的构建是一项时间和资源的投资, 它必将在面向服务的业务应用程序方面带来巨大的回报。对这些面向服务的应用程序可以做不同的修改以满足不断变化的业务需求。

## 1 Web 服务合成相关研究

Web 服务的定义很多, 目前比较统一的对 Web 服务的定义是由 IBM 公司提出的: “Web 服务是独立的、模块化的应用程序, 能够在网络(一般是 WWW)上被描述、发布、查找和调用。”Web 服务合成是指支持业务

收稿日期: 2007-12-03

基金项目: 国家科技基础条件平台建设基金项目(2005DKA3900)

作者简介: 曹会敏(1982-), 女, 河北保定人, 硕士研究生, 研究方向为计算机架构与组件技术; 林碧英, 教授, 研究方向为计算机架构与组件技术; 程振林, 博士, 研究方向为计算机架构与组件技术。

流程逻辑的一组 Web 服务,通过确定不同的 Web 服务的执行顺序和 Web 服务之间的复杂交互来实现。Web 服务合成本身既可以是最终的应用,也可以是新的 Web 服务。Web 服务合成作为 Web 服务一项重要的增值功能,为服务重用与自动化集成提供了应用的基础。但是,有关 Web 服务合成的问题目前尚没有统一的解决方案,也没有在生产型项目中大范围应用。

### 1.1 Web 服务合成方式分类

Web 服务合成方式大致可以分为两类:即静态合成和动态合成。

静态合成指的是,复合服务在设计阶段就被定义。合成过程对于服务请求者来说并不可见;服务请求者可以像调用基本服务那样调用复合服务。

动态合成指的是,在运行时刻选择和调用所需服务并将之合成为一个复合服务的过程。

对于动态工作流,可以进行运行时绑定,即当流程流转到活动时再进行服务查找、选择、排序和绑定,以适应环境的动态变化;但由于动态环境的复杂性,必将导致开销增加和效率降低问题,因此本系统采取关键业务动态绑定、非关键业务静态绑定的折中方式。在实际执行中,当服务由于某种原因不可得到时,选择重新执行此服务或选择替代服务重新绑定来完成。另外前一个活动执行时,为下一个活动和服务进行绑定或通知下一个服务准备数据和资源,可以节省时间。

### 1.2 BPEL4WS

商业流程执行语言(Business Process Execution Language for Web Services, BPEL4WS)是专为整合 Web Services 而制定的一项规范标准。它从本质上来说是 IBM 的 Web 服务流语言(Web Services Flow Language, WSFL)和 Microsoft 的基于 XML 语言的块结构化语言 XLANG 的结合物,目前已经成为业界标准。WSFL 支持图形化的流程,而 XLANG 在结构化构造方面有独到的方法,BPEL4WS 正是吸取了两者的优点,同时摒弃了一些复杂繁琐的部分,形成了一种较为自然的描述商业活动的抽象高级语言<sup>[2]</sup>。

由于 BPEL4WS 语言是专为商业流程的执行所服务的,因此它自然具有一门商业处理语言的特点,这体现在以下几个方面:

1) 提供了对于远程调用的同步和异步处理;

2) 提供了并行的操作,即通过 BPEL4WS,可以同时调用位于不同地方(不同城市甚至是国家)的 Web Services 进行处理;

3) 提供了补偿的操作。

由于以上 BPEL4WS 这些特点,非常适于本系统中关于服务合成的需要。

## 2 系统设计与实现

目前已有多种业界标准来规范 Web 服务互操作的基础架构,包括简单对象存取协议(SOAP)、Web 服务描述语言(WSDL)和通用描述/发现与集成协议(UDDI)<sup>[3]</sup>。其中 SOAP 为服务间能够互相通信定义了以 XML 格式表达的消息协议;WSDL 为描述服务提供了一种公共的方法;UDDI 注册表是发布和发现服务的基础设施<sup>[4]</sup>。通过这些规范,各个应用系统以松耦合的、与具体运行平台无关的方式彼此进行发现和互操作。再利用 BPEL4WS 语言将多个 Web 服务合成为一个功能单元。

由于 SOAP、WSDL、UDDI 和 BPEL4WS 技术的完备,为“科技基础条件应用服务集成及应用业务辅助生成环境建设”项目顺利完成提供了前提。

按照面向服务的思想,应用服务集成环境的软件框架的构成要素及层次关系如图 1 所示。

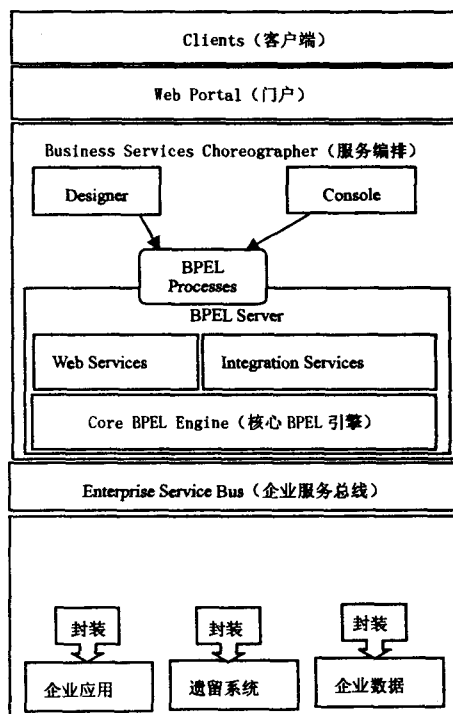


图 1 “应用服务集成环境软件框架”层次结构图

“面向服务的基础设施层”包括企业内的各种应用、遗留系统、企业数据以及企业外各类符合 Web 服务相关标准的服务。由于 Web 服务支持跨平台的互操作,包装器利用各种机制把企业内应用和遗留系统等包装为 Web 服务,便于外界访问使用<sup>[5]</sup>。其中企业数据指的是实验基地和大型科学仪器设备资源、自然科技资源、科学数据资源、科技文献资源和科技成果资源等的数据库,这些数据也被包装为 Web 服务,实现对各类关系型数据库的透明访问以及统一的数据管理和检索、查询等服务。而外部服务包括:各个面向领域的

服务体系的服务,如面向空间信息、水利、集成电路设计、生物信息、制造行业和信息服务等领域的服务体系;子平台内部提供的服务,如授权认证服务;科技基础条件平台的其它子平台提供的服务等。

企业服务总线(Enterprise Service Bus, ESB)是基于消息中间件、智能路由、数据转换等技术实现的,传统中间件技术与 XML、Web 服务等技术相互结合的产物,用于实现企业应用不同消息和信息的准确、高效和安全传递,提供了一种开放的、基于标准的消息机制,通过简单的标准适配器和接口,来完成粗粒度应用(服务)和其他组件之间的互操作,满足大型异构企业环境的集成需求。

服务编排层(Business Services Choreographer)主要利用 Oracle 的 BPEL Process Manager 进行业务流程的编排。Oracle BPEL Process Manager 包括三个主要组件:JDeveloper BPEL Designer、Oracle BPEL Process Manager Server 和 Oracle BPEL Console。其中,JDeveloper BPEL Designer 提供了一个图形化和用户友好的方式构建 BPEL 流程,支持使用 BPEL 进行业务流程的建模、编辑和设计。JDeveloper 使用 BPEL 作为其原生格式,因此构建的流程是完全可移植的,也可以直接将该流程部署到 Oracle BPEL Process Manager 中运行。Oracle BPEL Process Manager 提供了 BPEL 服务器的一个可伸缩的强健实现。Oracle BPEL Process Manager 执行标准的 BPEL 流程并提供“脱水”功能,因此长期运行流的状态在数据库中能够得以自动维护,从而实现了用于故障切换和可伸缩性的集群。BPEL Server 利用的是基础 J2EE 应用服务器。Oracle BPEL Console 为管理和调试部署到 BPEL 服务器上的流程提供了一个基于 Web 的成熟界面。审计跟踪和流程历史/报告信息可通过 BPEL Console 和 Java API 自动维护和提供。Oracle 的 BPEL Process Manager 可以通过它的适配器连接到数据源。适配器包括文件适配器、FTP 适配器、Oracle 的高级队列、数据库适配器、JMS 适配器和 Oracle 的应用程序适配器等。而 ESB 则是作为把各种应用包括传统应用暴露成 Web 服务接入 SOA 环境以及虚拟化接入系统的一个平台,当一个系统接入 ESB 后,则可以在不改变 BPEL 流程的情况下把一个系统换成另一个系统。

BPEL4WS,即 Web 服务的业务流程执行语言,是基于 Web 服务的一个新的流程描述语言的标准,也是上述 Oracle 的 BPEL Process Manager 基于的业务流程执行语言。利用 BPEL4WS 标准,创建能够完成如 Web 服务调用、操纵数据、抛出故障或终止一个流程等工作的不同活动,然后将它们连接起来,从而创建出

复杂的流程。

BPEL4WS 递归地组合结构化活动,以表达任意复杂的算法,这些算法表示了服务的实现<sup>[6]</sup>。而 SOA 的根本目的是业务的敏捷性,业务敏捷是从两个方面来理解的,一是 IT 在业务提出变化之后的一个恰当地时间段里相应这种变化;二是 IT 所做出的变化能够恰当地反映出业务的需求。SOA 出现以前,IT 与业务两个部门各自为政,不能够协同工作,经常还会出现冲突,这使得企业整体效率低下。而 SOA 的出现使 IT 与业务人员有了共同的语言。它将业务活动和业务流程向 IT 做了映射,使原本抽象并且变化多端的业务有了显性的描述和存在形式。用基本的语言描述业务并用技术手段编码实现,这样实现的业务称为服务,若干个服务被组装起来完成一个业务流程。当企业的业务流程发生变化时,只需要重新定义流程,就可以完成业务的需要,体现了 IT 灵活和业务敏捷性。因此,服务实现后接下来最重要的是如何构建组合的服务,整个过程如图 2 所示。



图 2 构建组合服务的流程图

逻辑上,构建服务组合的第一步是服务发现,即确定需要哪些服务。用于鉴别和发现候选服务的可行技术有三种,即自顶向下分析、自下向上分析以及业务流程跟踪。这些技术互为补充,每一种技术在服务发现过程中都发挥作用。自顶向下的分析一般会形成一个符合业务需要的、实际的候选服务集。不过,单凭该过程并不能发现机构内的所有候选服务。接下来自下而上的方法通常会产生大量的高级和低级候选服务。作为补充手段,对每个业务事件的生命周期进行跟踪,以便发现哪些服务是通过其生命周期处理该事件所需要的。该过程称为业务流程跟踪,它不但可以发现处理该事件所需的服务,还可以发现仅通过自顶向下或自下而上的方法操作时所遗漏的候选服务。除了识别交付项目所需的服务之外,该业务流程驱动的方法还能够提供完整性检查,并就特定服务的双重潜力给出首要指示。服务发现的最终结果将是一个概念上的服务组合,该组合包含了项目最需要的候选服务。

第二步是服务分类,发现一组候选服务之后,对它们进行分类是对其进行设计、开发和后续执行的至关重要的一环。分类可以按照功能、用途、结构和调用等标准进行。这种分类方法有助于识别属于同一功能域的服务、允许定义标准和最佳方法并对不同服务类的要求进行管理和监控。对服务进行分类的过程会发现业务服务的规则,可以将这些规则转换成一组应用于

不同类型服务的、标准的、可重用的策略。虽然服务进行分类还没有业界统一的标准,但通用描述、发现和集成(UDDI)注册表正在成为事实上的标准。通过对 UDDI 注册表中的服务进行分类,不但可以更好地为潜在的候选项分类,而且还能发现可以重用的现有服务,从而避免了功能的不必要重复。

第三步是确定服务粒度,服务的粒度是一项重要的设计要点。细粒度的接口为请求者应用程序提供了更多的灵活性,但也意味着交互的模式可能随着不同的服务请求者而不同,使对于服务提供者的支持更加困难。粗粒度接口保证服务请求者将以一致的方式使用服务,但却缺乏灵活性。针对本系统,对于外部的消耗推荐使用粗粒度的接口,而企业内部可以使用细粒度的接口。即“内部服务容器”中的 Java 应用容器组和 Web 服务容器组是细粒度的服务,而“外部服务容器”中是粗粒度的服务。

第四步是服务获取,定义了服务的组合之后,就要确定如何获取实际的服务,即内部构建、获取服务或预订外部供应商提供的服务。按照一般规则,那些关键性业务服务(即有益于业务流程、能为机构争取竞争优势地位的服务)最好是由内部提供。另外,创建一个服务之间关系的目录有助于构建服务组合。UDDI 注册表是一个价值极高的工具,利用它可以构建一个服务和相关产物的完整在线目录。

通过服务发现、服务分类、确定服务粒度以及服务获取,在资源共享与集成基础上,涉及科技基础条件资源涵盖的各个领域,跨平台、跨领域地构建 Web 合成服务,为实现科技资源的共享提供更加强大的服务。

构建的 Web 合成服务通过门户(Web Portal)提供包括内容聚合、单点登陆、个性化定制和安全管理等服

务的基础 Web 平台。客户端(Clients),即用户可以使用浏览器,支持 WAP 协议的手机或者其它的设备访问信息门户获取信息。

### 3 结束语

Web Services 技术的主要目标就是在现有的各种异构平台的基础上构筑一个通用的与开发平台、语言无关的技术平台,各种不同平台之上的应用依靠这个技术平台来彼此连接和集成。文中结合工作流技术和 Web Services 技术,在此基础上建立了基于 Web Services 的 Web 服务合成应用。目前 Web 服务合成技术在国内仍处于起步阶段,由于它需要其他相关技术的支持,如复合服务执行及事务处理等技术的协作,所以如果要使 Web 服务合成发挥出其所拥有的潜能,仍然有许多的研究工作需要开展。

#### 参考文献:

- [1] 柴晓路, 梁宇奇. Web Service 技术、架构和应用[M]. 北京: 电子工业出版社, 2003: 5-15.
- [2] 王 强. 业务流程开发新纪元——BPEL4WS 语言介绍[EB/OL]. 2003-03. <http://www.ibm.com/developerworks/cn/webservices/ws-bpel/part1/index.html>.
- [3] 麻志毅, 陈泓捷. 一种面向服务的体系结构参考模型[J]. 计算机学报, 2006, 29(7): 1012-1013.
- [4] 余名高, 贾秀峰, 林坤江, 等. 基于 Web 服务的企业应用集成[J]. 计算机技术与发展, 2007, 17(5): 56-57.
- [5] 许科峰, 高建民, 陈富民, 等. 基于 Web Services 的企业应用集成技术及实现[J]. 计算机应用, 2004, 24(3): 155-156.
- [6] BPEL4WS 语言规范[EB/OL]. 2002-07. <http://www.ibm.com/developerworks/library/ws-bpel/>.
- [7] (上接第 90 页)
- [8] Wang G, Liu M. Extending XML Schema with Nonmonotonic Inheritance[C]//In Proceedings of 1st International Workshop on XML Schema and Data Management(ER Workshop XSDM'03). Chicago, Illinois, USA: [s. n.], 2003.
- [9] 张晓琳, 王国仁. 用继承扩展 XML-RL[J]. 小型微型计算机系统, 2005, 26(2): 243-247.
- [10] Bourret R. DTD parser vision2.0[EB/OL]. 2005-04-10. <http://www.rpbouret.com/dtdparser/>.
- [11] 姚 砾, 束永安. 用 JavaCC 构造编译器的方法[J]. 计算机工程, 2003, 29(9): 39-41.
- [12] 张 昱, 付 雄. 含 Xpath 的表达式的解析与应用[J]. 小型微型计算机系统, 2004, 25(3): 442-446.
- [13] Sun o'reilly. Java Compiler Compiler - Javacc[EB/OL]. 2005-04-10. <http://javacc.dev.java.net/>.
- [14] Katz H. JavaCC、解析树和 XQuery 语法[EB/OL]. 2002-12. <http://bbs.xml.org.cn/dispbbs.asp?boardID=14&ID=8111>.
- [15] Sun o'reilly. The document of javacc[EB/OL]. 2005-04-10. <http://javacc.dev.java.net/doc/CharStream.html>.
- [16] Bansal A. Yacc 与 Lex 快速入门[EB/OL]. 2000-11. <http://www-128.ibm.com/developerworks/cn/linux/sdk/lex/index.html>.
- [17] Johnson S C. Yacc: 另一个编译器的编译器[EB/OL]. 2005-07-01. <http://mhss.nease.net/unix/yacc.html>.
- [18] 李朝阳, 潘 清. 应用 YACC 和 Lex 工具开发命令分析程序[J]. 软件世界, 1997(2): 44-45.