

Web 服务及其在 Parlay 接口中的应用

Web Services and Their Applications in Parlay Interface

杨鑫/YANG Xin

(中国电信股份有限公司 北京研究院, 北京 100035)
(Beijing Academy of China Telecom Corporation Limited, Beijing 100035, China)

摘要: 电信网与 Internet 走向融合, 而 Parlay 接口与 Web 服务作为各自领域开放技术的代表, 也开始了互相结合。Web 服务是一种基于可扩展标记语言(XML)、面向消息的分布式计算技术, 与公共对象请求代理体系结构(CORBA)等分布式对象技术相比, 在 Internet 范围内的互操作性更好。Web 服务是实现面向服务体系结构(SOA)的最佳候选技术之一。基于 Web 服务的 Parlay 接口包括 Parlay Web 服务和 Parlay X。其中, Parlay Web 服务模拟面向对象的 Parlay 应用编程接口 (API) 定义, Parlay X 的设计遵循 Web 服务面向消息的技术发展思路。基于 Web 服务的 Parlay 接口技术为构建电信网和 Internet 融合环境下的统一业务体系提供了基础。

关键词: Web 服务; 公共对象请求代理体系结构; 面向服务对象体系结构; 开放接口; Parlay 接口; 异步交互

Abstract: Along with the convergence of telecommunication networks and the Internet, the Parlay interface and Web Services, as typical open technologies in their own field, are developing towards convergence too. Web Services is an XML-based message-oriented distributed computing technology. It provides better interoperability while compared with the Common Object Request Broker Architecture (CORBA) and other object-oriented distributed computing technologies. And it is also one of the best technologies for implementing Service-Oriented Architecture (SOA). The Parlay interface based on Web Services includes Parlay Web Services and Parlay X. Parlay Web services simulate object-oriented Parlay Application Programming Interface (API) definitions and Parlay X follows the development trend of message-oriented Web services. The Parlay interface technology based on Web Services lays a foundation for building a unified service system in the converged environment of telecommunication networks and the Internet.

Key words: Web service; CORBA; SOA; open interface; Parlay interface; asynchronous interaction

中图分类号: TN915; TP393 文献标识码: A 文章编号: 1009-6868 (2006) 03-0041-05

开放和融合是当前电信网发展的两大特点。开放指的是网络与业务之间采用标准接口, 从而实现业务开发与网络能力分离, 为计算机界开发人员进行电信业务开发提供了技术便利, 有利于引入第三方开发的业务, 这是目前网络开放的一种重要思想和具体实施措施, Parlay 接口正是这样的网络能力开放接口的代表; 融合指的是包括固定网、移动网在内的电信网本身以及电信网与 Internet 融合。采用 Internet 领域兴起的 Web 服务的 Parlay 接口技术体现了上述的开放和融合, 值得关注。

另一方面 Web 服务作为一种新兴的软件技术, 当前人

们对其在认识上还存在一些误区: 或者把它看作一项简单的接口技术, 或者把它看成和公共对象请求代理体系结构 (CORBA) 等面向对象系统等同的技术, 而且由于 Web 服务与面向服务的体系结构 (SOA) 常常联系在一起以至于混淆它们之间的关系。

1 Web 服务

万维网联盟 (W3C) 对 Web 服务的定义^[1]如下: Web 服务是由通用资源标志符 (URI) 标识的软件系统, 其接口和绑定可以通过可扩展标记语言 (XML) 进行定义、描述和发现, Web 服务支持通过基于 Internet 的协议, 使用基于 XML 的消息与 Web 服务或者其他软件系统直接交互。

基金项目: 国家自然科学基金重点项目 (60432010)

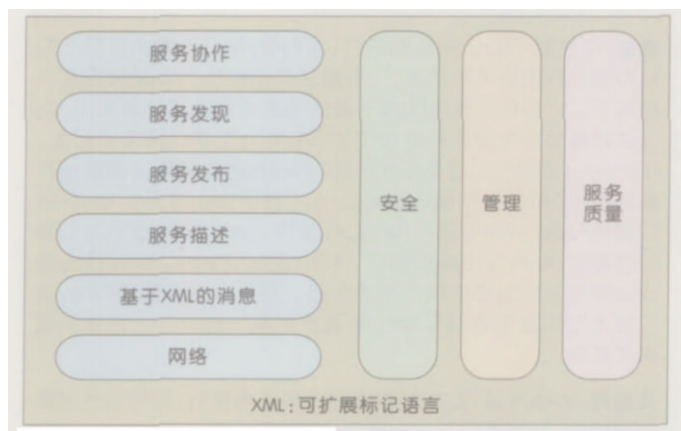


图1 Web服务技术框架

从定义可以看出, Web服务有以下几个显著的特点。

- 基于XML, 包括其接口描述、外界交互数据格式等。
- 基于消息, 而不是面向对象的方法调用。
- 采用URI标识, 并且基于Internet协议, 如超文本传输协议(HTTP), 是“Internet风格”的技术。
- 与外界的交互需要与具体协议绑定, 并且一般绑定的是Internet协议, 是在Internet范围内, 所以与外界可以进行“广域”的交互。
- 作为一种软件系统, 内部实现平台、编程语言等对外是不可见的, 外界通过接口了解系统。
- 可以被外界发现。

在Web定义中, 除了XML、URI, Web服务并没有特别指定采用什么技术实现其定义中声明的功能, 文献[2]给出了Web服务的一个最小“核”: 一个包含应用特定相关信息的XML文档, 一个可以发送或者接收和处理XML文档的软件程序, 一个绑定的协议地址。原则上包括了以上的3个部分就可以构建一个Web服务, 但是实际应用中, 还需要很多额外的具体协议和技术。图1给出了一个Web服务技术框架, 可以说用于建立一个完整和完善的系统, 涉及到的协议和技术很多, 但是Web服务最基本的技术包括以下3项: Web服务与外界交互的SOAP协议、Web服务描述语言(WSDL)、通用描述和发现集成(UDDI)规范。其中关于SOAP特别需要注意的是以下两项:

• 远程过程调用(RPC)和文档风格的SOAP消息绑定: SOAP交互本身是基于消息的, 但是也包含了过程或者方法调用的支持。RPC风格指出SOAP消息主体中包含了某个RPC调用或者响应的XML表示, 而文档风格指出SOAP消息主体包含的只是一个XML文档, 发送方和接收方需要遵循该文档的格式。

• 使用文字和使用编码的SOAP消息绑定用法: 消息绑定用法指的是用于确

定数据序列映射到普通的XML格式的序列化机制。使用文字用法, 将由XML模式定义(XSD)确定SOAP主体的XML格式, 不具有二义性, 而使用编码用法, SOAP处理器必须在运行时遍历各种SOAP编码规则以确定SOAP主体正确的序列化, 这将易于导致错误和互操作性方面的问题。SOAP编码的用法是XSD没有完善之前提出的一种SOAP消息绑定用法, 对它的支持已经可选甚至被要求不再支持^[3]。

1.1 Web服务与CORBA

从前面Web服务的定义中我们可以看出Web服务也是一种分布式计算技术, Web服务与CORBA等分布式对象系统之间存在一些类似之处, 但是它们之间也存在一些重要的差异, 正是这些差异使得Web服务能胜任一些CORBA等分布式对象系统难以完成的任务, 成为一种新的备受关注的技术。表1给出了Web服务与CORBA的一些比较。Web服务与CORBA最重要的差别在于前者面向消息, 后者面向对象, 前者基于文本, 后者基于二进制, 前者采用Internet协议, 后者采用OMG组织的标准协议。所以这使得二者比较起来, 前者是松耦合、后者是紧耦合; 前者效率低、后者效率高; 前者在Internet范围内均有很好的互操作性, 后者更适合在某个特定环境如Intranet范围内的分布式处理。CORBA依赖于一个严格管理的环境, 不同的CORBA系统可以互通, 但是很多厂商均有自己对象请求代理(ORB)实现上的一些特定的优化, 而不同ORB互操作时就没有了这种优化。

Web服务发展目的和方向并不是比CORBA更好的RPC系统, 虽然它提供了对RPC的支持, Web服务本质上是一个面向消息的系统, 它与一般面向消息的分布式系统的区别在于它建立在XML、Internet标准协议之上, 不是某个厂商的私有技术和产品, 具有开放性。Web服务的发展定位于构建Internet范围内具有良好互操作性、扩展性、动态性的服务系统。文献[2,4-6]分析介绍了Web服务的技术特点及一些认识的误区, 分析了RPC系统与面向消息系统的差别、对象封装成Web服务存在的问题, 并指出Web服务更适合位于比CORBA等面向对象系统更高抽象层次的集成和交互, 可以作为“中间件的中间件”(例如实现CORBA与DCOM系统的

▼表1 Web服务与CORBA的比较

比较项	公共对象请求代理体系结构(CORBA)	Web服务
端点命名	互操作的对象引用(IOR)	统一资源标识(URI)
交互模型	面向对象(方法调用)	面向消息(可扩展标记语言文档)
编码格式	二进制	文本(可扩展标记语言文档)
互操作协议	通用ORB间协议/Internet ORB间通信协议(GIOP/IOP)	SOAP协议/超文本传输协议(HTTP)等
接口描述	接口描述语言(IDL)	Web服务描述语言(WSDL)
有效负载数据格式	公共数据表示(CDR)	可扩展标记语言
发现发布	命名服务、交易服务、接口仓库、实现仓库	通用描述和发现集成(UDDI)
语言/平台	无关	无关

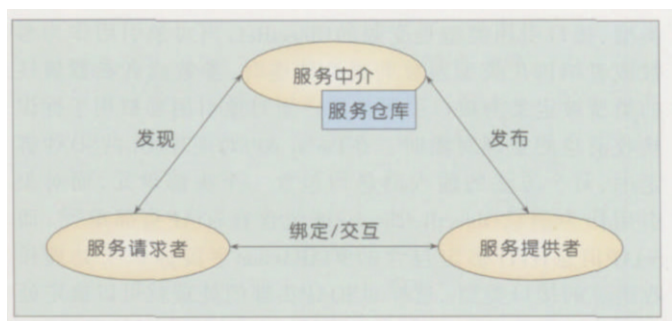


图2 SOA基本结构

互通)。文献[7]将Web服务使用分为4种场景:简单的远程服务通信、部分功能模块外包、整体流程建模和执行、自动组合和执行。

1.2 Web 服务与 SOA

SOA是一种通过已发布的、可发现的接口向分布在网络上的用户应用或者其他软件系统提供服务的软件系统构建方法,SOA的核心和基本概念是服务,W3C对服务的定义如下:“服务是一种表示执行任务的能力的抽象资源,一个服务必须被具体的软件实体实现才可以被使用”。服务作为SOA中的系统构建模块,也可以看成一种特殊的构件。SOA的基本结构如图2所示:包含3个角色(服务请求者、服务提供者、服务中介)、3个操作(服务发布、服务发现、服务绑定/交互)。服务中介一般包含了服务仓库。SOA各个角色之间是松耦合,并且交互一般是通过一种广泛接收的标准协议。

SOA、服务与基于构件的软件开发(CBSD)和构件的核心差别在于以下两点^[8]:

(1)SOA比CBSD更强调在运行时的发现、调用和集成,所以服务的描述比构件接口地位显得更重要,服务外部视图和内部实现的分离比构件更彻底。

(2)服务和构件的动态变化程度不同,对于前者,动态变化是典型的场景,而对于构件,一般并不这么假定,虽然也可能有一定的动态变化支持机制,所以支持动态变化的机制在SOA和CBSD中的重要性也是不一样的。

综上,SOA是一种非常适合应用于Internet环境下软件系统的体系结构模型和系统构建方式,并且SOA没有限定某种具体实现的技术。

SOA和Web服务仍然是在发展中的技术,Web服务对SOA的支持也不够完善,需要进一步研究。

2 Web 服务在 Parlay 接口中的应用介绍

Parlay应用编程接口(API)是Parlay组织定义的一组开放的、独立于具体实现技术的、面向对象的网络能力开放接口,其定义是基于面向对象方法采用统一建模语言(UML)进行描述的,独立于具体中间件平台,目前Parlay API的实

现一般采用CORBA平台,但是Parlay API同样可以建立在Web服务之上,这种基于Web服务实现的Parlay API被称之为Parlay Web Services。除了Parlay API,Parlay组织还定义了Parlay X,一种基于Web服务的粗颗粒网络能力开放接口(Parlay API是一种细颗粒的网络能力开放接口),Parlay组织的网络能力开放接口体系如图3所示。

本文中的Parlay接口泛指Parlay组织定义的网络能力开放接口,包括细颗粒和粗颗粒两种类型。本文中基于Web服务的Parlay接口指的是Parlay Web Services、Parlay X以及它们的集成。

需要注意的是Parlay X只包括向应用提供的网络能力接口,并不提供认证、授权、计费(AAA),业务等级协定(SLA),网络能力注册等部署环境相关的接口。Parlay Web Services除了向应用提供的网络能力接口,还包括框架等接口,相对Parlay X,更加全面。Parlay X相对Parlay Web Services更加高层,它将电信网络能力粗颗粒地提供给应用。Parlay X一般和Parlay Web Services或者其他Parlay API实现集成在一起,单独的Parlay X不构成一个完整的网络能力开放方案。当前Parlay X提供的能力包括呼叫控制、短消息处理、多媒体消息处理、终端定位、帐号管理等。

Parlay接口采用Web服务技术,除了Parlay接口定义采用WSDL,各方的交互基于SOAP协议,Web服务的UDDI可以引入到Parlay接口的部署环境,为应用对网关的发现、各种网络能力的发布等提供了新的方式。例如各种Parlay网关可以在UDDI注册中心登记,应用通过UDDI查找发现Parlay网关,选择合适的Parlay网关访问网络。

基于Web服务的Parlay接口对应用开发的影响主要体现在:计算机软件开发人员相对更容易理解和使用基于Web服务的接口,特别是Parlay X这样的抽象层次较高的网络能力粗颗粒接口;Web服务的开放性和标准性使得网络可以有更广泛的应用接入和使用;基于Parlay Web Services或者Parlay X开发的业务更容易与其他系统例如企业应用系统

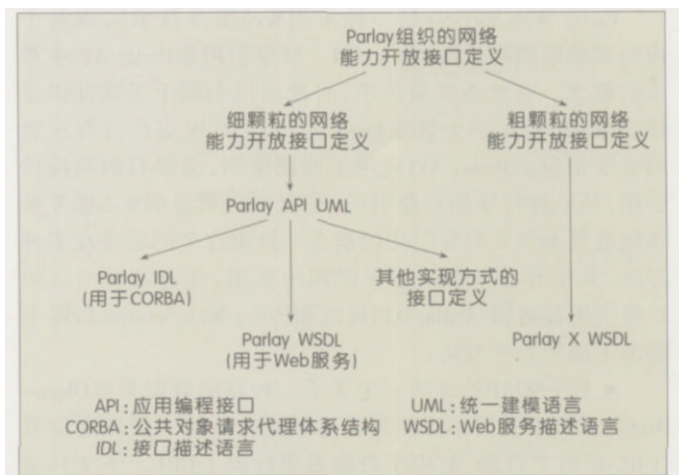


图3 Parlay组织的网络能力开放接口体系

集成,提升各自的价值。

3 基于Web服务的Parlay接口的技术分析

下面从多个方面分析总结基于Web服务的Parlay接口(Parlay Web Services和Parlay X)的技术特点:

(1)WSDL绑定

考虑到当前实现技术,Parlay Web Services和Parlay X目前规定采用SOAP/HTTP协议作为Parlay逻辑体系中框架、业务能力服务器和应用之间通信的绑定协议。

(2)WSDL消息绑定的风格和用法

前面已经提到SOAP支持RPC和文档两种风格的消息绑定,而SOAP消息绑定可以采用编码的用法,也可以采用文字的用法,这样基于SOAP的绑定一般可以提供4种风格/用法模式:RPC/编码、RPC/文字、文档/编码、文档/文字,另外还存在一种文档/文字包装模式,其中文档/编码模式是没有意义的^[9],由于Parlay API是方法调用形式,所以基于Parlay API UML规范的Parlay Web Services中的SOAP采用RPC风格进行消息绑定,规范中给出了RPC/编码模式的WSDL定义。而Parlay X在1.0版本规范中只给出RPC/编码模式的WSDL定义,在1.0.1版本规范中增加了RPC/文字模式以及文档/文字模式的WSDL定义,在最新的2.0版本规范中又只给出了文档/文字模式的WSDL定义,Parlay X WSDL定义形式的变迁反映了Parlay X的定义风格遵循Web服务面向消息的技术本质特点。

(3)WSDL文档分离

Parlay X从1.0.1版本开始在除采用编码用法的其他模式中WSDL定义不再全是在一个文档中,而是区分为服务抽象接口、服务具体绑定以及服务数据类型3种类型文档,这种分离有利于重用。虽然目前Parlay Web Services规范并没有对WSDL定义文档实行分离,但是也可以采用这种定义方法。

(4)对象引用

Parlay Web Services是一种采用Web服务技术实现基于面向对象原则制订的Parlay API,对象引用是Parlay API中基本的概念。在分布式系统中,对象引用不同于关联标识这样的数值,它是一个数据结构,包含了可以定位远程对象的必要信息。Parlay API使用了回调机制,需要有回调接口引用,是一种特殊的对象引用。但是面向消息的Web服务基本规范里面没有对象引用的概念。解决这个问题存在多种方法,其中有的仅仅限于在回调中使用,而有的对回调和对象引用都适用。Parlay API规范和Parlay Web Services白皮书给出了两种处理方法:

- 基于SOAP的方法:定义了一种新的数据类型ObjectRef表示对象引用,该数据类型被定义为统一资源定位器(URL)后附字符串,其中字符串后缀标识了URL上下文环境下特定的远程对象实例。每一个接口均有一个对应的引用

类型,接口引用类型是受限的ObjectRef,当对象引用作为参数或者结构化类型参数的属性传递时,参数或者参数属性的类型被定义为接口引用类型。当对象引用要被用于标识接收消息的处理对象时,在Parlay API对应WSDL的SOAP绑定中,每个方法的输入消息均包含一个头部单元,而对象引用作为消息ObjectRefHeader被包含在SOAP头部单元,而SOAP消息HTTP绑定包含的SOAPAction字段标识了处理接收消息的接口类型。这样对SOAP头部的处理就可以确定处理接收消息的对象。该方法依赖于SOAP的头部扩展机制,适合对象引用和回调的处理。

- 基于Web服务实现机制的方法:在WSDL的类型定义区内定义了一个引用单元,在消息定义区中当某个消息参数相当于对象引用时,那么将消息参数的类型定义为这个引用单元。引用单元包括一个端点、一个端口以及一个数据元素ObjectData。端点是服务的地址,端口被用于标识需要用到存根和用于在远程系统组装存根,ObjectData用于标识特定的对象。在使用回调接口引用或者其他对象引用时,存根将被组装,负责基于SOAP或者其他绑定协议与端点所在的服务进行通信,ObjectData作为参数通过存根传递,ObjectData可以是包含定位对象实例的ID,也可以是使对象可以重构的数据。该方法依赖于Web服务具体实现机制(例如:存根和骨架)来提供运行时必要的支持,不要求对SOAP等传送层协议进行修改,适合对象引用和回调的处理。JAX-RPC规范(2.0版本之后改名为JAX-WS)等技术支持这种方法的具体实施。

(5)异步交互

Parlay API方法根据交互是否同步分为同步方法和异步方法。生成一个对象,检索数据是否存在等均为同步交互,同步交互一般发生在不要求业务能力服务器(SCS)与网络中其他节点通信时,而当需要不定时长或者较长时间处理的情况下采用异步交互,这样可以防止交互发起方的阻塞和对交互响应方是否出错的猜测。Parlay API主要采用了回调机制实现异步交互,包括下面两种模式^[10]:通知模式、请求-响应模式。WSDL本身支持单向、请求/响应、请求/响应、通知这4种类型的操作。基于WSDL的这4种类型操作,有多种方式用于构建Web服务异步交互^[11],实现异步交互的关键在于两点:一是提供请求和响应关联的机制,二是响应的回复地址。对于Parlay Web Services来说,其异步交互经过分析可以总结为如图4、图5、图6所示的3种情形。这3种情形的异步交互均是通过创建关联标识实现请求和响应关联,均是把回调接口引用作为响应的回复地址,并且均是基于WSDL的请求/响应操作。不同点在于请求者在请求时传递回调接口引用还是单独的设置回调接口引用,是由请求者还是提供者负责关联标识的创建。Parlay Web Services的异步交互需要支持对象引用和回调接口引用概念。

对于Parlay X来说,其交互是基于XML文档的消息传递

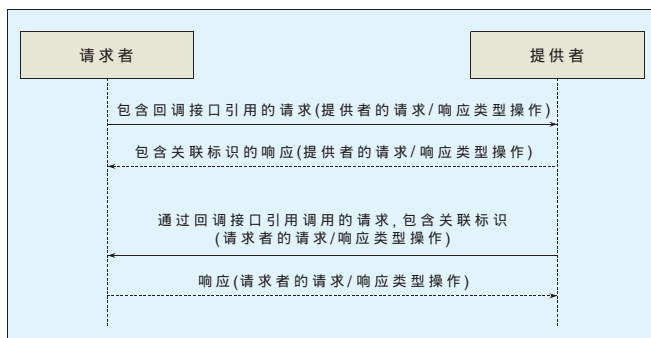


图4 Parlay Web服务异步交互情形1

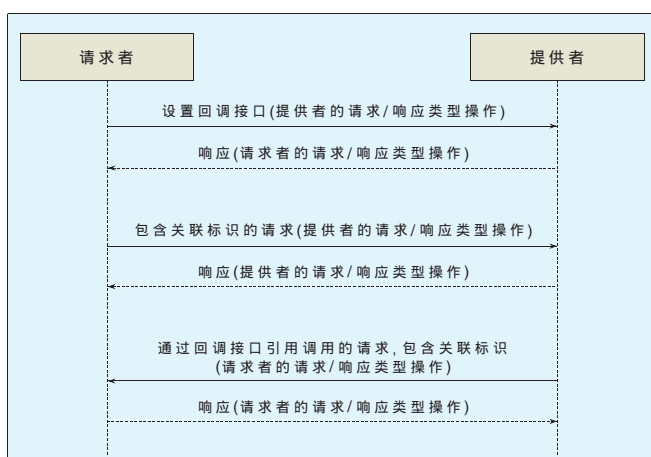


图5 Parlay Web服务异步交互情形2

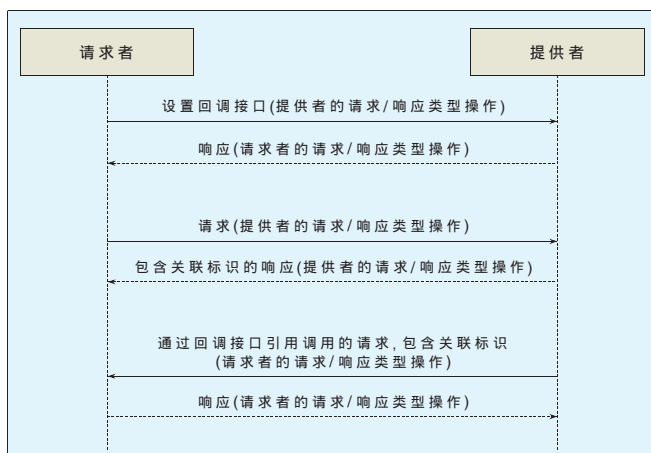


图6 Parlay Web服务异步交互情形3

而不是模拟面向对象的方法调用,并且遵循接口尽量保持简单易用的原则。Parlay X应用与Parlay X SCS之间的消息交互一般遵循同步的请求/响应操作并且是应用触发的,Parlay X Services对于应用来说应该是无状态的,应用的每次调用不存在关联关系。可以采用轮询的机制避免进行异步交互,但是有时异步交互也可以被定义,例如应用侧实现一个接受Parlay X SCS通知的Web服务,Parlay X SCS异步通知

应用,具体的每一次交互,均是同步的请求/响应类型操作。

从上面的分析总结中可以看出Parlay Web Services是让Web服务支持“面向对象”的API定义,而Parlay X则除了接口本身较Parlay API简单易用和抽象层次更高,其定义也逐渐摆脱“面向对象”的痕迹,回归“面向消息”的本质,所以可以说Parlay Web Services是一种“形似而神不似”的Web服务网络能力开放接口,而Parlay X是一种“纯”的基于Web服务的网络能力开放接口,这种“纯”不仅体现在一开始接口就是直接采用WSDL定义,而且体现在其设计思想、具体定义风格均是愈来愈符合上面提到的Web服务技术发展思路。

4 结束语

Web服务是当前计算机界的热门技术,而Parlay接口是电信界业务领域研究人员广泛关注的技术,基于Web服务的Parlay接口技术体现了上述两种技术的结合,为构建电信网和Internet融合环境下的统一业务体系提供了基础。

5 参考文献

- [1] W3C Working Group. Web services glossary[EB/OL].[2004- 02- 11]. <http://www.w3.org/tr/2004/note-ws-gloss-20040211>.
- [2] Vogels W. Web services are not distributed objects[J]. IEEE Internet Computing, 2003, 7(6):59- 66.
- [3] Ewald T. The argument against SOAP encoding[EB/OL]. [2005- 06- 07]. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsoap/html/argsoape.asp>.
- [4] Vinoski S. Where is middleware?[J]. IEEE Internet Computing, 2002, 6(2): 83- 85.
- [5] Vinoski S. Web services interaction models—part 1: current practice[J]. IEEE Internet Computing, 2002, 6(3): 89- 91.
- [6] Vinoski S. Web services interaction models—part2: cutting the “Web” into web services[J]. IEEE Internet Computing, 2002, 6(4):90- 92.
- [7] Shapara D. Approaches to web service composition[EB/OL]. [2005- 06- 25]. <http://science.unitn.it/~tomasi/think/pdf/sdmi.pdf>.
- [8] Cervantes H, Richard S. Autonomous adaptation to dynamic availability using a service- oriented component model [C]// Proceedings of the International Conference on Software Engineering (ICSE 2004). May 23- 28,2004, Edinburgh, UK. Piscataway, NJ,USA:IEEE, 2004:614- 623.
- [9] Russell B. Which style of WSDL should I use?[EB/OL]. [2005- 03- 25]. http://www-900.ibm.com/developerWorks/cn/webservices/ws-whichwSDL/index_eng.shtml.
- [10] Annex A. Asynchronous patterns in the parlay API[S]// Parlay Group. Parlay UML to Java API Rulebook 1.1. 2003: 28- 32.
- [11] Adams H. Synchronous operations and Web services[EB/OL]. [2005- 03- 25]. <http://www-106.ibm.com/developerworks/library/ws-asynch2/index.html>.

收稿日期: 2005- 12- 21

作者简介



杨鑫,北京邮电大学毕业,博士。现工作于中国电信股份有限公司北京研究院,曾参与多项国家自然科学基金、国家“863”计划、国家“973”计划项目的研究和开发工作,主要研究方向为下一代网络、新一代业务提供体系和业务开发方法。