

Web 服务技术的分析^{*}

罗 玲¹ 白晓颖²

(清华大学计算机科学与技术系 北京 100084)

摘 要 Web 服务提出了一种新的分布式环境中应用系统的集成模式,并基于 XML 标准定义了一组简单、开放的协议规范。它的主要目标就是构筑一个通用的平台无关、语言无关的技术层,以实现各种异构平台上的应用间的互连互通。文章首先介绍了 Web 服务架构以及该架构中的角色和操作,分析了 Web 服务技术系列的标准协议 SOAP/WSDL/UDDI/WSFL,并讨论和比较了主要的 Web 服务的解决方案和支撑平台。Web 服务是不断发展的体系,Web 服务技术还不成熟,文章分析了 Web 服务在服务质量、安全、性能等方面存在的技术问题,以及今后主要发展方向。

关键词 Web 服务,XML 可扩展置标语言,SOAP 简单对象访问协议,WSDL Web 服务描述语言,UDDI 统一描述、发现和集成,WSFL Web 服务流语言

A Survey of Web Services Technology

LUO Ling¹ BAI Xiao-Ying²

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract Web Services technology introduces a new model for application integration in a distributed environment. To support the model, a set of simple and open protocols and standards are defined based on XML. The objective is to establish a universal technical layer independent of hardware/software platforms and programming languages, and thus to enforce intercommunication and interoperability among Web applications. The paper first reviews Web Service Architecture and the roles and operations involved. It then analyzes Web Service Protocol Stack, including SOAP/WSDL/UDDI/WSFL. The paper also discusses and compares major vendor solutions and platforms supporting Web Services. The paper finally analyzes the existing problems, such as quality, security and performance, and points out the directions for future research and development.

Keywords Web Services, XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration), WSFL (Web Services Flow Language)

1. 概述

Internet 技术的广泛普及和迅速发展极大地促进了 Web 应用的发展,其目标是最大限度地实现各种网络资源的共享,向用户提供更丰富的资源、更方便快捷的信息交互。由于各种 Web 应用软/硬件平台的千差万别,开发语言、通信协议和数据交换的格式也不尽相同,如何提高异构的网络结点之间的互操作能力成为实现 Web 应用的互连互通、降低系统集成代价的关键。Web 服务技术就是为了更简单更有效地解决这一问题而产生的。

Web 服务由 W3C 联盟制定的一系列标准组成,用来促进跨平台的程序间的通信^[1]。它基于 HTTP 等网络传输协议,采用结构化的信息建模语言 XML,定义了信息交换协议 SOAP(Simple Object Access Protocol,简单对象访问协议)、服务描述规范 WSDL(Web Service Description Language, Web 服务描述语言)、服务发布注册机制 UDDI(Universal Description, Discovery and Integration,统一描述发现集成)、以及描述 Web 服务工作流的 WSFL(Web Services Flow language, Web 服务工作流语言)。目前,针对 Web 服务的服务质量、安全性、可靠性、事务处理、路由等方面的协议都在不断的研究和完善中。

与传统的分布式计算技术(如 DCOM、CORBA 以及 Java

的 RMI 等)相比,采用 Web 服务技术的解决方案具有以下优点:

- 松散耦合性。采用 Web 服务技术,客户端不需在程序中绑定连接对象,而是动态获取服务端的地址,动态建立连接和获取服务。只要服务调用的界面不变,Web 服务地址或服务实现方式的变化对调用者都是透明的。这种松散耦合关系提高了基于 Web 服务架构建立的系统的灵活性和可扩展性。

- 简单性。Web 服务采用 XML 技术定义其协议栈,XML 是开放环境下数据描述标准,它的简单性、可扩展性和规范化使 Web 服务的实现比传统的实现技术更简单高效,并且便于设计、开发、集成和维护业务系统。

- 高度可集成性。Web 服务采取简单易理解的标准 Web 协议作为服务界面和协议描述的规范,屏蔽了不同平台的差异,而且对于用 CORBA、DCOM、EJB 等不同技术实现的组件都能够高度地集成。

- 规范性。由于使用规范的 XML 格式描述服务,Web 服务具有较强的机器可理解性,易于程序实现动态的发现、连接和调用异地服务。

- 开放性和行业支持。Web 服务基于 XML,建立了开放的协议族和技术规范,并得到工业界的广泛支持。大型软件公司如 IBM、微软等均提供免费工具可以使开发者快速创建和部署 Web 服务。

^{*} 国家科技攻关项目:奥运会信息系统集成测试总体方案及集成测试管理平台的预研(041250001)。罗 玲 硕士研究生。白晓颖 讲师。

本文首先介绍了 Web 服务的架构和 Web 服务实现的核心技术 SOAP、WSDL、UDDI 和 WSFL。然后分析和比较了 Web 服务的主要解决方案及其支持平台,包括微软的 .Net 架构,Sun 的 Sun ONE 架构,以及 IBM 的策略。Web 服务的协议族在不断发展之中,已发布的规范在不断地修订;新的协议和规范逐渐出台以完善 Web 服务功能,提高服务性能。文章最后概述了 Web 服务当前主要问题和今后发展方向,并列出了 W3C 制订的 Web 服务的发展框架。

2. Web 服务技术

2.1 Web 服务的基本概念

图 1^[3]描述了 Web 服务中所涉及的主要角色和构件及其相互之间的关系。

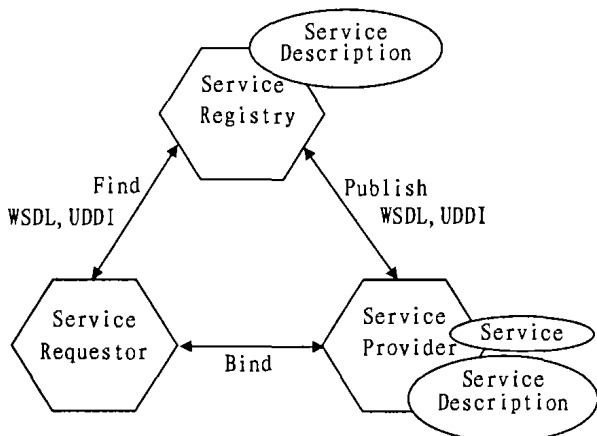


图 1 Web 服务角色、操作和构件

Web 服务是由服务描述和服务实现两部分构成的。服务描述(Service Description)用于描述服务调用的接口信息,如服务的数据类型、操作、绑定信息、网络位置等。此外还可以包括方便服务请求者发现和利用的分类及其他元数据。服务即服务描述的实现,是一个软件模块,部署在可通过网络访问的服务平台上。

Web 服务的应用通常涉及到服务提供者(Service Provider)、服务请求者(Service Requestor)及服务注册中心(Service Registry)三种应用程序。服务所有者通过向服务注册中心注册服务描述来发布(Publish)服务,并通过服务访问平台提供服务;服务提供者在服务注册中心搜索(Find)满足需求的服务,根据其服务描述解析服务调用方式,并动态绑定(Bind)服务提供者,获取服务。

图 2 描述了 Web 服务的概念性协议栈^[4](Web Services Conceptual Stack)。

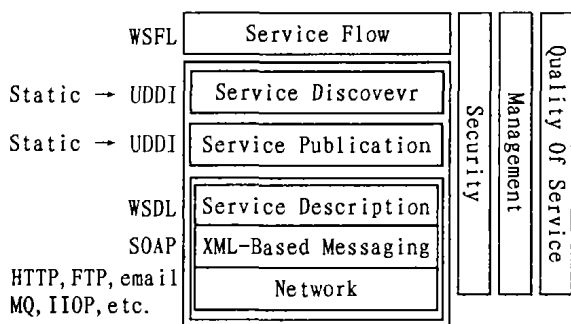


图 2 Web 服务概念性协议栈

(1)Web 服务协议栈的最底层是网络层(Network),该层

根据 Web 服务应用程序的需要选择适当的网络协议,可以是 HTTP、FTP、email、消息排队(MQ, Message Queuing)、因特网 ORB 间协议(IIOP, Internet Inter ORB protocol)上的远程方法调用等。对 Web 服务开发者来说,网络技术的选择是透明的。

(2)网络层之上是消息传递层,由 SOAP 协议实现。SOAP 是一种简单轻量级的基于 XML 的用于在网络应用程序之间交换结构化数据的机制。

(3)服务描述层使用 WSDL 作为基本的服务描述规范,是降低服务提供者和请求者之间耦合程度,实现动态绑定服务、自动集成应用的关键。

(4)协议栈的中间两层是 Web 服务的发布和发现,推荐的规范是 UDDI。Web 服务的发布包括服务描述的生成和服务的发布;Web 服务发现包括获取服务描述和使用服务描述。

(5)最上面的一层是业务流程的建模,描述如何执行服务到服务的通讯、合作和工作流。

(6)Web 服务的服务能力的提高有赖于安全性、管理和服务质量的综合提高。协议栈的垂直条描述了这些方面,它们是协议栈的每一层都必须解决的问题。

下面详细介绍关键的几个协议:SOAP、WSDL、UDDI、WSFL。

2.2 SOAP

SOAP 基于 XML,为在分布的环境中对等地交换结构化、类型化的信息提供了一个简单且轻量级的消息传递机制^[5]。SOAP 包括四部分(如图 3 所示):一个用于定义描述消息内容的框架的信封(SOAP Envelope);一组表示应用程序定义的数据类型实例的编码规则(SOAP Encoding);表示远程过程调用和响应的约定(SOAP RPC);以及使用底层协议实现信息交换的 SOAP 绑定(binding)。

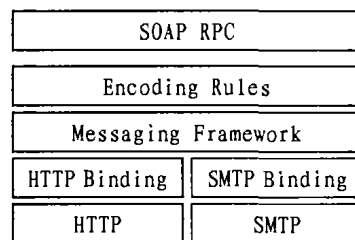


图 3 SOAP 结构

SOAP 信封定义了以 SOAP Header、SOAP Body 为子主体的描述结构。SOAP Header 提供可伸缩机制能够扩展 SOAP 消息,如在通讯双方没有约定的前提下可以实现安全认证、事务管理等 Header 条目。SOAP Body 元素描述与消息最终接收者交互所必须的调用、响应的详细信息。

SOAP 编码方法基于一个简单类型系统,该系统是程序语言、数据库和半结构数据中类型系统的公共特性的泛化。SOAP Encoding 为 SOAP Header 和 SOAP Body 中的数据表示提供了默认的方式,使类似整数(integer)、字符串(string)、数组(array)、结构(struct)等数据类型得以在 SOAP 消息中表示。默认的 SOAP Encoding 定义和 XML Schema 是相容的。

SOAP RPC 表示利用 XML 的可扩展性和可伸缩性来包装和交换 RPC 调用,定义了统一的远程过程调用和响应的规范。RPC 调用和响应信息均在 SOAP Body 中描述,包括目标

SOAP 节点的 URI、方法或过程的名称、可选的方法或过程的特征、方法或过程的参数以及可选的头数据等。理论上,可以使用任意一种网络传输协议来实现基于 SOAP 的 RPC。例如,在使用 HTTP 作为 SOAP 协议绑定的情况下,RPC 调用和响应可以分别映射为 HTTP 的请求和响应。

SOAP 绑定定义了将 SOAP 消息在底层网络传输协议之上或之内传输的规范和规则。SOAP 的传输无关性使其可以与各种已有的网络通信协议进行绑定,如 HTTP、FTP、SMTP、POP3 等。一般而言,SOAP 协议依附于其绑定的通信协议的数据部分来传送 SOAP 消息,并利用该通信协议的控制部分实现其调用交互模式。

2.3 WSDL

WSDL 用来描述 Web 服务,并提供了对辅助信息的说明能力,包括 Web 服务可以识别的数据类型、消息模式、交互方式、服务的位置、错误信息和标头信息等。它将服务描述成为一组端点(port),客户端可以通过这些端点访问面向文档信息或面向过程调用的服务^[6]。WSDL 将操作和消息抽象描述,并和一个具体的网络协议和消息格式绑定,以最终定义具体部署的端点。

WSDL 定义的服务描述由服务接口定义(Service Implementation Definition)和服务实现定义(Service Interface Definition)两部分组成(图 4^[3])。

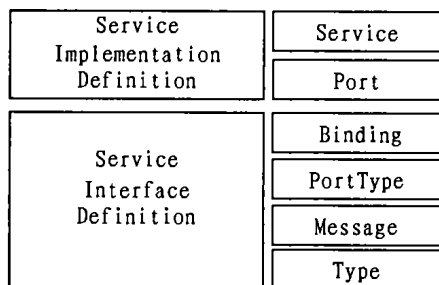


图 4 基本服务描述

服务接口定义类似 IDL (Interface Definition Language 接口定义语言),是一种抽象的可重复的服务定义,并且类似编程语言中定义的抽象接口,可以被多个服务实现定义实例化和引用。服务接口定义中包含的 WSDL 元素组成了服务描述中的可重用部分:WSDL: Binding, WSDL: PortType, WSDL: Message 和 WSDL: Type 元素。Binding 描述特定服务接口的协议、数据格式、安全性和其他属性;PortType 元素是端口类型,其中定义了 Web 服务的操作,操作定义了输入/输出数据流中可以出现的 XML 消息;Message 元素指明组成消息各个部分的数据类型。Type 描述消息中复杂数据类型的使用。

服务实现定义是描述特定服务提供的访问入口信息细节的 WSDL 文档。Web 服务被建模成 WSDL: Service 元素。Service 元素包含一组(通常是一个)WSDL: port 元素。Port 将端点(URL)与来自服务接口定义的 WSDL: Binding 元素相关联。

2.4 UDDI

UDDI^[7]技术规范是由 IBM, Ariba 和 Microsoft 为促进商业性 Web 服务的互操作能力而推出的一项计划。UDDI 提供了标准化的、透明的、服务注册、发现和访问调用机制,目的是在 Internet 这个广泛信息操作网络空间中,建立商务信息的集散地,使得企业用户可以方便、快速、动态地互相发现彼

此提供的服务,并完成服务间的应用集成。

UDDI 的核心是 UDDI 商业注册。UDDI 注册采用 XML 标准,从白页(White Page)、黄页(Yellow page)、绿页(Green Page)三个层次描述企业及其提供的 Web 服务:“白页”内容包括了地址、联系方法和已知的企业标识;“黄页”包括了基于标准分类法的行业类别,支持美国政府的 NAICS 工业代码分类法、UN/SPSC 产品服务代码分类法以及地理位置的分类法三种途径来区分公司;“绿页”则包括了关于该企业所提供的 Web 服务的技术信息,如电子商务规则、服务描述、应用的调用方法及数据绑定等。UDDI 注册的描述形式可能是一些指向文件或是 URL 的指针,这些指针是为服务发现机制服务的。所有的注册信息都是通过 UDDI 注册中心存储和发布的,包括公共的和私有的注册中心。

UDDI 机制提供了服务发布和服务发现两种功能。服务发布可以采用直接发布和间接发布等不同的机制。直接发布是服务提供者直接把服务描述发布给服务请求者。间接发布是通过 UDDI 运营商结点发布服务描述。IBM, Microsoft, Ariba 都支持、复制和托管 UDDI 运营商结点。服务发现包括获取服务和使用服务。与服务发布相对应,服务请求者可以直接向服务提供者获取服务,或是间接通过查询 UDDI 运营商结点获取。

2.5 WSFL

Web 服务的目的是使应用程序能够在网络上进行无缝集成,而不必考虑它的编程语言和运行环境。而 Web 服务工作流的目的就是使商业流程和利用了大量 Web 服务的交易生命周期能够进行同样的无缝集成。Web 服务流语言^[8](WSFL)是 IBM 定义的一种建立在 Web 服务描述语言(WSDL)和 XML 语法的基础之上的语言。通过它可以在 Web 服务架构的框架内定义软件的工作流过程。

WSFL 中的每一个活动是一个 Web 服务,活动之间通过控制链和数据链机制贯通。WSFL 的核心是定义工作流模型,主要有流程模型(flowModel)和全局模型(globalModel)两种模型。

流程模型是用 XML 表示的一个对商业流程建模的指示性图表。其结构被用于把多个 Web 服务组成工作流,而每个 Web 服务则由各自的 Web 服务描述语言(WSDL)文档来定义。

全局模型用来明确指定在整个过程中 Web 服务相互之间如何实现预期的交互作用。当流程被执行时,全局模型指定 Web 服务之间的消息在流程模型中传递的一系列必需链接。

WSFL 商业流程本身可以定义成 Web 服务。WSFL 定义生命周期接口,描述在一个特殊的 Web 服务应用程序中所有 WSFL Web 服务支持的一组基本操作。这些操作包括:商业流程的激活、挂起、恢复、阻塞、终止,以及查询其当前状态。生命周期接口是由 WSDL 定义的 Web 服务接口。

3. Web 服务的主要解决方案

3.1 Microsoft .NET

Microsoft .Net 是 Microsoft 推出的开发 Web 服务的整体的技术框架(图 5)^[9]。.NET 框架是开发、部署、运行 XML Web 服务和网络应用程序的编程模型。其中包括最低层的 Common Language Runtime(公共语言运行环境)和上几层的 Class Libraries (类库)。

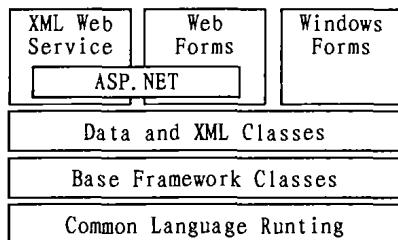


图5 .Net 框架

Common Language Runtime 提供了一个支持中间代码的统一的运行平台,是 .NET 架构中解决跨平台和可移植性的核心。VB, VC 以及 C# 等各种编程语言均可以编译成 Common Language Runtime 能够执行的中间代码,Common Language Runtime 提供了运行时语言集成、安全、内存、过程管理、线程管理等服务。并实现了开发过程中的生命期管理、异常处理、动态绑定、类型系统转换等。采用这种方式,一方面提高了函数库、运行库等组件的重用性,降低了开发工具和平台支持的代价;另一方面提高了系统的平台无关性和可移植性。

Class Libraries 提供了简化编程工作的一些重要的类。Base Framework Classes 提供了如输入/输出、字符串操作、安全管理、网络通信、线程管理、文本管理等功能。Data and XML Classes 支持一致性的数据管理,如支持 SQL 的类、完成 XML 数据的操作/查找/解析的 XML 的类等。ASP.NET 提供开发 Web 应用的类。XML Web Services 提供开发 Web 服务相关的类。Web Forms 提供快速开发 GUI(Graphic User Interface, 图形用户界面)应用程序的类;Windows Forms 提供开发 Windows GUI 应用程序的类。

Microsoft.NET 框架的优点是:类库丰富;具有基于 XML 的通信的灵活的体系结构;支持多种编程语言;支持 Web 服务,缺省的互操作的通信标准使用的是 SOAP,可以自动生成 WSDL;具有集成开发环境 Visual Studio.NET 的工具的强有力的支持。

Microsoft 的 Web 服务的一个最突出的问题是,难以在 Windows 平台之外的平台上使用。Visual Studio.NET 缺乏对 Java 的支持;对当前使用的 Visual Basic 和 ASP 语言需要进行强制转换^[2]。

3.2 Sun ONE

Sun ONE (Open Network Environment, 开放网络环境)^[10]针对分布式的网络环境为 Web 服务提供了一个基础平台,具有开放性和标准化的优点。它的目标是为所有接入设备提供访问平台、可以使众多的网络设备和 Web 服务供应商之间进行通信。

如图 6^[10]所示,Sun ONE 的框架共分为三层。最底层是操作系统、硬件、存储器、网络等构建的平台。最高层提供构建、集成、发布服务的开发工具。中间层有三部分。前端是服务分发(Service Delivery),即服务的外部表示,包括界面、服务描述等;中间是服务容器(Service Container),类似于 Java2 平台下的 J2EE 技术中的应用程序服务器,是服务的载体;最后端是服务集成(Service Integration),是后台数据的交换过程,包括访问数据库和应用程序以及其他的服务交互等。

Sun ONE 是在现有的 J2EE 框架上,将 Web 服务的一系列技术 XML、SOAP、UDDI 并入并融为一体。Sun ONE 的目标是实现任何信息接入设备均可在任何时间、任何地点对网

络进行访问,因此,其体系结构的设计重点是标准化和开放性,技术核心是 Java 和 XML。目前,它最主要的缺点是对 Web 服务的描述和发现的两个标准(WSDL 和 UDDI)的支持尚不完善。

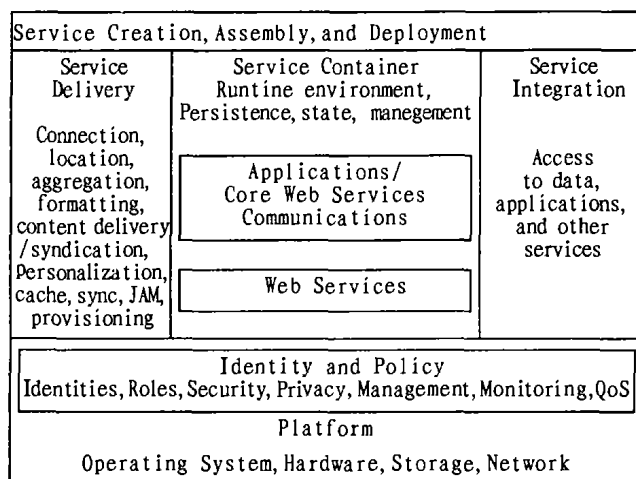


图6 Sun ONE 框架

在开发环境方面,Sun ONE 的开发策略是不依赖一种工具,Borland 的 Jbuilder Enterprise, IBM 的 Visual Age for Java 等都是针对 Java 的开发环境。Sun 提供的开发平台 Forte for Java^[11]可以广泛支持 XML 和 XSL 文件,具有丰富的数据库编程和源代码控制工具,但缺乏对 SOAP 的支持和其他标准 Web 服务技术的支持,只是一般的编辑环境。

3.3 IBM Web Services

在 Web 服务的规范上,Microsoft 是 IBM 的主要合作伙伴,诸如 SOAP、WSDL 和 UDDI,IBM 和 Microsoft 都是绝对的技术先入者。而在内部实现技术,与微软和 Sun 的战略不同,IBM 并没有推出一个整体构架。它采用 J2EE 架构,除了依靠自己的 alphaWorks 的力量外,还博采各类开放源代码组织的成果(诸如 Apache SOAP 和 jUDDI 等),在自身的 Websphere 平台上提供了完整而且领先的 Web 服务的开发工具。

IBM 总的目标是将其产品都升级为支持 Web 服务技术,实现自身软件系列向 Web 服务时代的变迁。目前,IBM 的 Web 服务领域主要有以下开发软件^[2]:

- WebSphere, Web 服务的运行平台和部署平台。其中的 WSDT(Web Services Development Environment Toolkit)是 IBM AlphaWorks 提供的免费的 Web 服务的开发工具,用于 Web 服务的创建、部署和测试。WebSphere Application Server 中内置有对 UDDI 注册中心的客户访问端。
- Visual Age for Java 是 Web 服务应用的开发和集成工具。它支持 Web 服务的创建,测试,由服务接口定义的 WSDL 文档生成 Java 类的服务导入功能以及 UDDI 注册中心的浏览、发现、下载和发布功能。
- DB2 UDB 7.2. 为 Web 服务环境提供数据管理能力。
- Tivoli 提供对运行 Web 服务的 IBM 中间件的管理等。

4. Web 服务的主要问题和发展方向

4.1 当前主要问题

Web 服务本质上是一种消息传递体系结构,通过在不同的系统/操作环境/程序语言间传送 XML 格式的数据,实现松散耦合的应用程序之间的相互协作。要实现在企业级的部

署和应用,Web 服务还必须解决下面的问题:

- 安全性/隐私权。Web 服务提供访问接口和访问方法,这些信息发布后增加了 Web 服务的安全隐患,提高了被系统被攻击的机会。此外,对于所调用的分散、异构的外部 Web 服务的安全性的管理更为困难。Web 服务体系结构安全性有两个安全级别:网络级安全和内容级安全。网络级的安全考虑的是信息在网际传输的安全性和完整性,如进行数据加密,使用数字签名等。内容级的安全性包括 Web 服务访问权限的控制,请求者身份的认证,以及服务提供者的确认等。

- 消息传递/路由。由于分布式的应用系统间的服务发现/发布/集成借助于大量的消息传递,因此,消息传递/路由的效率对提高 Web 服务的质量至关重要。

- 服务质量/可靠性。迄今为止,Web 服务的一致性和可靠性机制都还不完善。服务间协作能力、事务处理能力、消息路由的控制管理能力和差错控制能力等都是服务质量要考虑的问题。

- 事务处理。事务是构建可靠性的分布式应用程序的基本概念,是确保应用程序的各方都能得到已协议的结果的机制。传统的事务处理都具有:原子性、一致性、隔离性和持续性。Web 服务的松散耦合性,使得 Web 服务对应用程序操作和结果的控制能力较弱。随着 Web 服务的发展,服务间的交互更为复杂,对服务可靠性要求提高,Web 服务处理事务的能力也需要不断提高。

- 管理。传统的分布式计算环境有很多的实用程序/工具来深入了解网络和系统情况,以及应用程序的状态和行为模式。与 Web 服务模式相当的实用程序/工具的发展还不足。此外,Web 服务的松散耦合性,增加了对于分布式应用程序的管理难度。

- 性能。对于事务处理、网络/系统/应用程序的管理,W3C 联盟一直把重点放在协议和基础架构上。所以当前缺少对分布式 Web 服务应用程序及服务器的调优工具和实用程序的规范约定。检查 Web 服务的性能优劣的测试方案、工具以及评价体系都不健全。

- 互操作性。Web 服务的优点就在于可以实现异构软/硬件平台的互操作性。Web 服务在理论上是互操作性良好的。但是在实际应用中,越来越多的供应商和产品之间的组合对 Web 服务互操作性的挑战还是很大。而且对于互操作性的测试的研究也还不深入。当前 WS-IO (Web Services Interoperability Organization, Web 服务互操作性组织)组织成立,致力于解决供应商之间问题以及为互操作性测试提供地点。

4.2 完整的 Web 服务栈和 Web 服务的发展方向

W3C Web Services Architecture WG (W3C Web 服务体系结构工作组)在 2002 年 11 月提出了完整的 Web 服务体系结构的工作草案^[3]。对 Web 服务的体系结构进行了扩展,规划了 Web 服务更多方面的功能特性(图 7)。包括基本的服务间通信、服务描述、服务发布查找机制,也包括服务安全性、可靠性、服务质量、事务管理等方面的考虑。

- Wire 层,是实现底层的数据打包传输的机制。

- Description 层,是服务的描述层。实现数据建模、规范定义数据格式、描述服务以及一些扩展的服务细节信息等。

- Discovery Agencies 层,实现服务的检查、发布、查找机制。

纵向是服务在各个层都要实现的关于服务质量、安全性和管理的功能。

图 7 所显示的 Web 服务的整体技术体系,有些是已经成熟的技术,有些还是正在发展中的技术。Web 服务的发展是在这一体系框架下,逐渐丰富技术协议栈,提高服务水平。

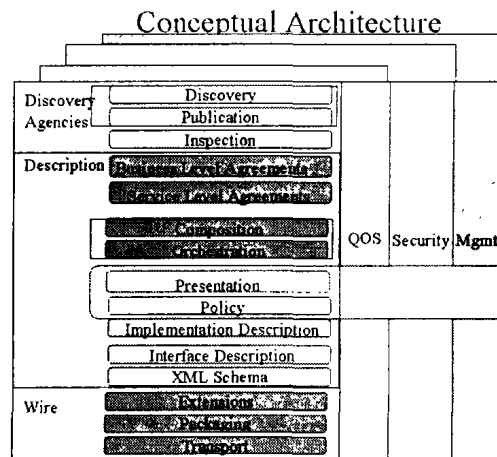


图 7 Web 服务栈

结论 Web 服务引入了一种新的 Web 应用的开发、部署和集成的模式。它通过采用基于 XML 的消息传递机制,实现了位于分布式环境中松散耦合的应用程序之间跨越不同网络、操作系统、软/硬件平台的互操作。采用 Web 服务的系统集成模式,可以有效地提高组件的重用性,实现服务的动态发现和绑定,降低系统集成的复杂性以及费用。

由于 XML 技术的简单、灵活、可伸缩、易定制,基于 XML 数据描述和 XML Schema 数据建模定义的 Web 服务的协议栈具有标准化和开放性的特点,各协议规范得到了工业界的广泛支持,有效地促进了 Web 服务的不断发展,使 Web 服务具有强大的生命力,成为未来分布式系统中应用集成的主流。

参考文献

- 1 Clabby J. Web Services Gotchas, An executive summary. <http://www-900.ibm.com/developerWorks/cn/webservices/wsgotcha/index-eng.shtml>. July 2002
- 2 柴晓路. Web 服务架构与开放互操作技术[M]. 清华大学出版社,2002
- 3 W3C Working Draft. Web Services Architecture [s]. <http://www.w3.org/TR/ws-arch/>. 14 November 2002
- 4 IBM Software Group. Web Services Conceptual Architecture (WSCA1.0)[s]. May 2001
- 5 Simple Object Access Protocol (SOAP) 1.1 [s]. <http://www.w3.org/TR/SOAP/>. W3C Note 08 May 2000
- 6 Web Services Description Language (WSDL 1.1)[s]. <http://www.w3.org/TR/WSDL/>. W3C Note 15 March, 2001
- 7 UDDI Version3.0, published specification [s]. <http://uddi.org/specification.html>. 19 July,2002
- 8 IBM Software Group. Web Services Flow Language (WSFL 1.0) [s]. May 2001
- 9 Microsoft. .NET Framework Technical Overview. <http://www.getdotnet.com/team/framework/default.aspx>. 2001
- 10 Sun ONE Architecture Guide. Chapter2 The Sun Open Network Environment (Sun ONE) Architecture
- 11 Sun ONE Architecture Guide. Chapter7 Development tools