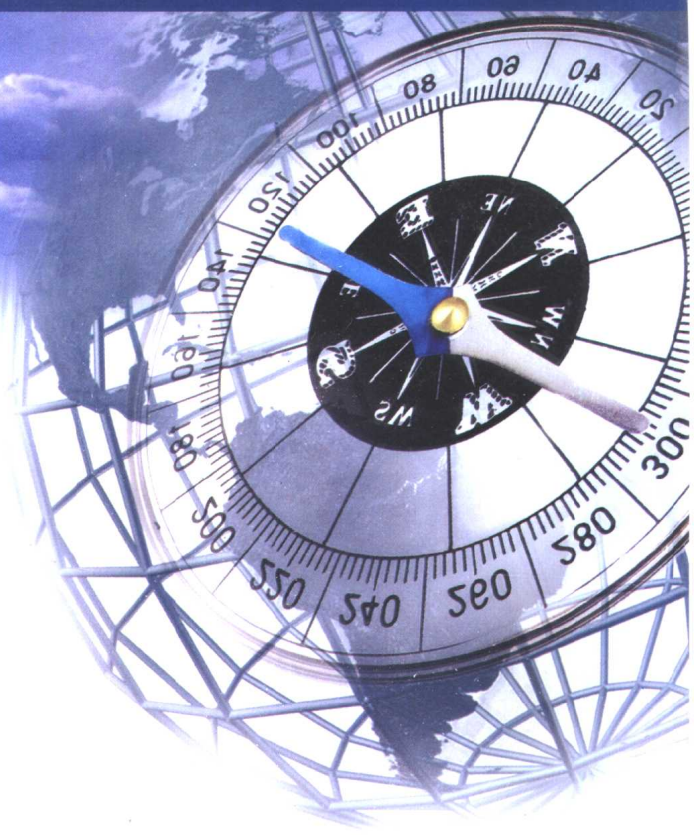
中科院软件所电子商务中心
李安渝　等 编著



Web Services
Technology and Implementation

Web Services
技术与实现

國防工業出版社
National Defence Industry Press
http://www.ndip.com.cn

Web Services 技术与实现

李安渝　等编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

Web Services 技术与实现/李安渝等编著 .—北京:国防工业出版社,2003.1
ISBN 7-118-03031-7

Ⅰ.W... Ⅱ.李... Ⅲ.互联网络-网络服务器 Ⅳ.TP368.5

中国版本图书馆 CIP 数据核字(2002)第 096959 号

国防工业出版社出版发行
(北京市海淀区紫竹院南路 23 号)
(邮政编码 100044)
北京奥隆印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 23 526 千字
2003 年 1 月第 1 版 2003 年 1 月北京第 1 次印刷
印数:1－4000 册 定价:32.00 元

序

Internet 是一个巨大的快速变化的异构的分布计算环境。近年来随着电子商务的广泛应用和深入发展，对网络环境下异构系统之间的互操作和应用集成提出了更加迫切、更加现实的需求，极大地推动了以 Web 为基础的分布计算技术的发展，从而催生了 Web Services 这一个新的热门领域。

什么是 Web Services，Web Services 的技术内容是什么，与 Web Services 相关的知识、技术和标准规范有哪些，Web Services 能解决哪些问题，Web Services 有哪些特点和优势，如何实现 Web Services，有哪些实现 Web Services 的方法，支持 Web Services 的主要产品和工具有哪些，诸如此类的问题是广大计算机工作者从专家教授、专业技术人员、软件开发人员到广大的计算机用户、在校学生都十分希望了解和学习的内容。

作者邀我为本书写序，实际上给了我一个认真阅读全书的机会。读完全书我感到：

这是一本 Web Services 的入门书。作者以通俗的语言来介绍和讲解 Web Services 的基本概念和涉及的众多知识。

这是一本 Web Services 的技术书。作者勾画了 Web Services 的体系架构，描述了在这个体系架构中涉及的技术和相互关系。重点介绍了 Web Services 的核心技术。

这是一本 Web Services 的开发指导书。本书介绍了 Web Services 的多种实现方法以及开发 Web Services 的流程，给出了许多实例和可运行的代码，最后还描述了一个完整的 Web Services 实现过程的实例。

这也是一本 Web Services 的工具书。本书汇总整理了与 Web Services 有关的标准和规范，如 SOAP、WSDL、UDDI 等。讲解了这些标准和规范在 Web Services 架构中的作用。本书最后还给出了许多参考资料，例如 UDDI API 参考手册。

本书的作者所带领的技术团队，中科院软件研究所电子商务技术研究中心，在 Web Services、电子商务技术、电子政务技术、XML 技术的研究和标准化等方面承担了许多国家课题，进行了大量研究，具有深入的技术知识和丰富的开发经验，取得了诸多成果。本书就是他们成果的一个代表。

Web Services 是一种涉及面很广的技术，涉及到多个领域、多种技术；Web Services 是一个快速发展的技术，不断更新不断发展；Web Services 是一个实用的技术，有巨大的应用需求和市场潜力；Web Services 代表了下一代网络计算和企业应用的必然趋势。本书的出版必将为 Web Services 技术的研究和发展，应用与推广作出贡献。

中国人民大学信息学院院长
中国计算机学会副理事长　王珊
2002 年岁末

前　言

欢迎进入 Web Services 的世界！

自从 1987 年 Sun Microsystems 为它的网络文件系统(Network File System,NFS)开发作为底层通信机制的开放网络计算(Open Network Computing,ONC)远程过程调用(RPC)系统以来,分布式计算已经有了 10 多年历史了。RPC 和消息传递机制为普及分布式计算立下了汗马功劳。随着分布式计算的普及和企业系统集成需求的日益复杂,越来越多的应用需求希望能够访问不同架构、不同平台下、用不同语言实现的应用程序和数据,Web Services 的概念应运而生。

本书从商业和技术的角度介绍 Web Services 的思想,并且提供一个描述 Web Services 的整体框架,使大家能够更清楚地了解与 Web Services 相关的许多标准和规范及其在 Web Services 架构中的作用,比如简单对象访问协议(Simple Object Access Protocol,SOAP),Web 服务描述语言(Web Services Description Language,WSDL)和统一描述、识别和集成(Universal Description,Discovery and Integration,UDDI),以及 Web Services 是如何帮助解决商业问题,特别是应用集成方面的实际问题。

本书的另一个目的是帮助开发人员了解开发 Web Services 时面临的问题及其细节。比如在设计 Web Services 的时候需要考虑哪些方面,都需要具备哪些条件等。我们提供了大量的例子和可运行的代码来加深对 Web Services 的理解。

读者的背景要求

我们不要求大家是分布式计算的专家,但最好能够熟悉基于 Web 的应用架构和技术,比如 HTTP 和 HTML 等。关于这方面我们不会详细介绍,现在有很多相关的书籍可供大家参考。另外,由于书中的例子选择了 Java 作为开发语言,所以读者最好熟悉 Java 语言,特别是 Java Server Pages(JSP)和 Servlet 技术,如果了解 EJBs 就更好。另外,XML 是 Web 服务的核心,虽然我们用了整整一章来讲述,但如果对 XML 了解越多,则对 Web Services 的理解就越深刻,所以对 XML 了解和掌握得越多越好。

本书的编排

本书的第 1 章由介绍 Web Services 的概念入手,描述了什么是 Web Services,与 Web Services 相关的标准和技术都有哪些,以及 Web Services 所能解决的问题。我们在这一章里引入 SOA 框架的概念并说明 Web Services 的各种标准,如 SOAP,WSDL 和 UDDI 是如何协同工作的。本章给读者提供了阅读其余各章的概念基础。本章由李安渝编写。

在深入核心 Web Services 标准之前,我们在第 2 章介绍一下 XML。XML 是一个很大的话题,有许多分支和相关的标准,我们把注意力主要集中在与 Web Services 有关的部

分,包括 XML、XML Namespace 和 XML Schema 等。本章作者为孙一中。

了解了 XML 之后,第 3 章开始涉及 Web Services 的核心问题——SOAP。SOAP 是 Web Services 里服务请求者和服务提供者进行通信的核心。在这一章里,我们先介绍它的基础知识,主要是 SOAP 的消息封装机制。接着通过实例分别介绍了如何用 Java 实现 Web Services 以及如何用 SOAP 编程来访问 Google 的 Web Services。

至此,我们已经了解了 SOAP 的背景和它的实现方式,第 4 章讨论了实现 Web Services 的多种方法,分别用 Perl,Apache SOAP,.NET 实现了简单的 Web Services。以上两章作者为林立杰。

第 5 章介绍 Web Services 描述的一个重要概念——描述语言(WSDL)。WSDL 是 Web Services 作为建立松耦合系统的应用集成技术的一个关键因素。本章讨论了如何描述 Web Services 来解决服务请求者了解服务提供者并调用 Web Services 的问题。本章作者为傅朝晖。

现在,我们需要知道服务请求者是如何得到服务描述的。第 6 章讨论 Web Services 的发现机制(UDDI)。这章介绍了有关查找 Web Services 有关的标准及商业上的支持。本章作者为黄向阳。

第 7 章举例说明如何用 UDDI 的 API 来发布和查找 Web Services,以及有关 UDDI 注册中心(Registry)的讨论。本章作者为匡宏波。

最后,我们在第 8 章以 IBM 产品为代表,介绍了目前对 Web Services 支持的产品和工具。而且,通过一个虚拟的 Smart 公司使用 Web Services 的过程为例,详细地描述了 Web Services 实现的架构,见证了 Web Services 的无穷魅力。本章作者为白钢。

本书总体架构由李安渝设计,白钢、孙一中和傅朝晖共同完成本书的统一审稿工作。

鉴于编者的水平有限,疏漏和错误之处在所难免,请不吝赐教。

内 容 简 介

电子商务的市场需求推动了分布式计算技术的蓬勃发展，Web Services 应用而生，它是业界第 1 次把 XML、SOAP、WSDL 和 UDDI 等标准综合起来的技术，是新一代电子商务核心。本书从商业和技术的角度介绍 Web Services 的思想、概念和来源，系统而全面地介绍了与 Web Services 相关的标准和规范。内容包括 Web Services 概述、XML 统一信息描述、SOAP 基础、创建 Web Services、WSDL 基础、UDDI 基础、UDDI 编程和 Web Services 过程为例，详细地描述了 Web Services 实现的架构，为读者生动地展现了 Web Services 的无穷魅力。

本书内容完整，实例丰富，实用性强。适用于任何对 Web Services 感兴趣的人士，是广大计算机软件开发爱好者、在校学生、专业技术人员了解和运用 Web Services 技术的入门必备。

目　录

第 1 章　Web Services 概述

在数字经济的时代，任何企业都希望通过互联的网络切实提高企业的核心竞争力，让企业在变幻莫测的市场竞争中获得优势。比如某家国内手机厂商在日益激烈的手机市场竞争中意识到，必须利用信息化途径来提高企业运营效率。于是，他们建立了客户服务中心，使得合作伙伴（供应商、代理商和分销商等）及时得到供求信息，他们还建立了公司的网站，不仅为一般消费者提供产品宣传，同时也为分销商提供在线的订单处理系统，利用电子化方式简化了实际业务流程。该电子商务系统运行的初期，获得很好的效果，得到了很多客户的关注和利用。

但是，系统投入使用以后，该系统的问题便逐渐暴露出来了。该公司的订单处理系统虽然复杂，但是却不能连接到其他的在线应用，于是从客户的订单到向供应商发送订货单不得不依赖人工的干预。也就是说，合作伙伴不能很好地连接到他们的供应链系统（背后的技术原因是系统平台和数据结构等问题）。随着公司业务的增长，订单收集和库存供应等人工劳动耗费了大量的人力和物力，延缓了产品推向市场的时间（time to market），这显然成为了限制公司成长的瓶颈。

为了能够集成企业内部和企业之间的信息系统，并且能对某些流程进行自动化处理，该公司意识他们需要改变原有的分布计算的模式。该模式需要突破具体平台（操作系统、编程语言、应用服务器等）的限制，使用统一灵活的数据格式，并且使用开放的标准协议，以满足业务的复杂化和规模的扩展，尤其重要的是，需要有成熟工具的支持。

当他们评估了多家公司的解决方案之后，得出了一致的结论：无论企业电子商务系统将建立在怎样的平台之上，Web Services 将成为共同的技术趋势。遵照 Web Services 架构的分布式系统可以方便地实现应用的集成，满足企业发展的需要。

那么 Web Services 究竟是什么？Web Services 如何满足企业的新需求呢？我们将在本书中为您逐渐揭开谜底。作为 Web Services 的基础概述，我们将在本章中介绍 Web Services 的概念和特点，说明 Web Services 的架构以及组成的协议规范，并分析 Web Services 的优势以及各大公司对 Web Services 的支持。

1.1　什么是 Web Services

21 世纪伊始，Web Services 获得了巨大的发展。许多软件公司纷纷宣布了对 Web Services 的支持和应用。许多组织参与了 Web Services 标准的完善。虽然对 Web Services 的理解正慢慢地趋同，但是目前仍然没有统一明确的定义。这种情况非常类似于以前的面向对象编程（OOP）。在继承、封装和多态等概念被明确定义之后，人们才把 OOP 引入到主流开发方法的殿堂。

Web Services 是在 Internet 上进行分布式计算的基本构造块。开放的标准以及用户和应用程序之间的通信协作产生了一种新的环境，在这种环境下，Web Services 成为应用集成的平台。应用程序通过使用多个不同来源的 Web 服务构造而成，不管这些服务到底位于何处或者如何实现，它们都可以相互协同工作。

因此，Web Services 就是由服务组件通过某些网络协议提供的远程调用接口。Web Services 通常是使用 SOAP 协议，而 SOAP 本身是一种基于 XML 的高层协议，它需要绑定到某种底层网络通信协议上。Web Services 并不是一种新的服务端组件，而是原来的服务端组件提供了一种新的通过 SOAP 协议来调用的统一接口。

如图 1-1 中所示，服务端组件本身并没有改变，只是在前面增加一个 Web Services 接口层，使得客户端可以在不需要知道服务端组件具体是由什么技术来实现的情况下透明地透过 Web Services 来调用它。

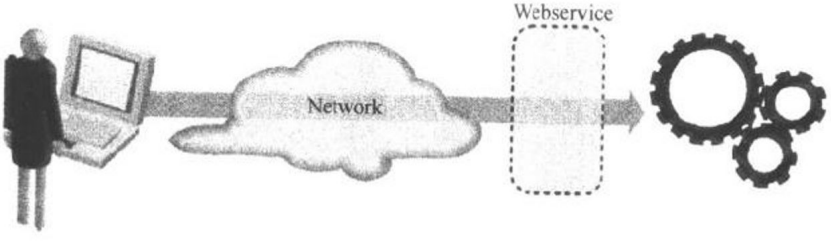


图 1-1 Web 服务提供统一接口

那么，究竟什么是 Web Services 呢？目前的现实是：有多少支持 Web Services 的公司，就可能有多少种 Web Services 定义。不过几乎所有定义都具有以下共同点：

- Web Services 通过标准的 Web 协议向 Web 用户提供有用的功能。多数情况下使用 SOAP 协议。
- Web Services 可以非常详细地说明其接口，这使用户能够创建客户端应用程序与它们进行通信。这种说明通常包含在称为 Web 服务描述语言（WSDL）文档的 XML 文档中。
- Web Services 可以进行注册，以便潜在用户能够轻易地找到这些服务，这是统一描述、发现和集成协议（UDDI）来完成的。

1.1.1 Web Services 的定义

目前几大主要的 Web Services 基础架构提供者已经发布了他们对于 Web Services 概念的理解。IBM 对 Web Services 的定义是：

“Web Services 是描述了操作集合的接口，它可以通过标准的 XML 消息机制在网络中进行存取。Web Services 实现了特定的任务或者是一系列任务的集合。Web Services 使用标准的 XML 形式来描述，称为业务描述。业务描述提供了与业务进行交互的所有必要信息，包括消息格式（详细描述操作）、传输协议和位置等。该接口隐藏了业务实现的细节，因此它可以用和实现语言独立的编程语言来使用业务。这允许并且鼓励基于 Web Services 的应用在实现时耦合松散、面向对象并且能够跨不同的技术。Web Services 能单独使用，也可以和其他的 Web Services 联合来完成复杂的商务交易。”

Microsoft 有两种对 Web Services 的定义。第 1 个定义是：

“Web Services 是一个应用逻辑单元，它为其他应用提供了数据和业务。应用通过通用的 Web 协议和数据格式，如 HTTP、XML 和 SOAP 来访问 Web Services，不必考虑每个 Web 服务是如何实现的。Web Services 结合了基于组件开发和 Web 的优势，是组成 Micrsoft.NET 编程模式的基石。”

Microsoft 对 Web Services 的第 2 个定义是：

“Web Services 是编程的应用逻辑，它可以通过标准 Internet 协议进行访问。Web Services 集组件开发和 Web 技术之所长。Web Services 和一般组件的类似之处在于它也代表了可以重用的黑盒，而业务实现的细节则不必关心。而和目前的组件技术不同的是，Web Services 不通过特定的对象模型来进行访问，比如分布式组件对象模型 DCOM、远程方法调用 RMI 或者是 Internet ORB 互操作协议 IIOP。相反，Web Services 可以通过通用的 Web 协议和数据格式，如 HTTP、XML 和 SOAP 来访问。而且，Web Services 的接口根据其接口和产生的消息严格定义。Web Services 使用者可以在任何平台以任何编程语言实现，只要他们可以创建并且使用由 Web Services 接口所定义的消息。”

Sun 提供以下的 Web Services 定义：

“Web Services 是软件组件，它可以被自动地查找、组合、重组，以提供用户请求的处理方案。Java 语言和 XML 是 Web Services 中最重要的技术。”

由以上可以看到，对于 Web Services 的含义已经取得了广泛的共识，但是还没有一致的定义。许多开发人员都会认为他无法确切定义什么是 Web Services，但是能够判断具体的服务是否是 Web Services。从我们的角度看来，Web Services 是与平台和实现独立的软件模块，它们能够：

- 使用业务描述业务来进行描述；
- 发布到业务注册库；
- 通过标准机制进行查询（在运行时和设计时都可以进行查询）；
- 通常在网络中，通过声明的 API 进行调用；
- 能够和其他服务进行组合。

1.1.2　Web Services 的特点

需要澄清的是，Web Services 提供的服务不一定要存在于 Web 上。“Web Services”的名称只是习惯性的约定俗成而已，并不能仅从字面上来理解。Web Services 可以位于任何网络中，不仅是外部的 Internet 网络，还可以是内联网，甚至可以在相同的操作系统进程中，用简单的方法来调用，或者可能在运行于相同机器上的紧耦合进程之间使用共享内存。实际上，Web Services 并不一定非要依靠 HTML 为主的 WWW 浏览器。其用户界面和表现形式有多种，比如手机、PDA 或者其他智能终端中。

Web Services 中另一重要之处在于，Web Services 实现平台的细节和业务调用程序无关。Web Services 可以用其声明的 API 和调用机制（网络、数据编码模式等）进行访问。这种方式类似于浏览器和 Web 应用服务器之间的关系。Web 服务器不必关心使用它的哪类客户——可以是不同种类的浏览器或者甚至是不使用浏览器的客户。这种方式使得 Web Services 可以形成松散耦合的组件系统。

Web Services 还要求具有以下特性：

■ Web Services 应该是自描述的。如果你发布了一个新的 Web Services，应该为该服务提供公共接口。至少，你的服务应该包括了可读性的文档，这样其他开发人员可以更方便地集成你的服务。如果你创建一个 SOAP 服务，也应该包括一个用通用 XML 语法编写的公共接口。

■ Web Services 应该是可查找的。如果你创建一个 Web Services，应该有一个相对简单的机制来发布该服务。或者说，应该有比较简单的机制使得感兴趣的人员可以发现该服务并且定位其公共接口。确切的机制集中在注册系统中。

■ Web Services 应该是可以互操作的。Web Services 通过 SOAP 实现相互间的访问，任何 Web 服务都可以与其他 Web 服务进行交互，避免了不同协议之间的相互转换。Web 服务可以用任何语言编写，因此开发者不需要更改开发环境就能开发新的 Web 服务，同时还可以在新的 Web 服务中使用已有的 Web 服务，而不必考虑 Web 服务的实现语言、运行环境等具体实现细节。例如，用 Delphi 编写的 Web 服务，可以使用由 Visual C + + 编写的 Web 服务，反之亦然。

■ Web 应该有普遍性。Web 服务使用 HTTP 和 XML 进行通信，任何支持这些技术的设备都可以拥有和访问 Web 服务。Web 服务不仅在计算机网络上出现，而且将在电话、汽车、家用电器等设备中出现。现在，各主要设备和软件供应商都已宣布支持 SOAP 和周边 Web 服务技术，相信在未来，Web 服务将普遍存在于社会生活的各个领域。

对于 Web Services 的外部使用者而言，Web 服务是一种部署在 Web 上的对象组件，它具备以下特征：

■ 良好的封装性

Web 服务是部署于网络上的对象，具备对象组件自然具有的良好封装性。而对于使用者而言，他仅能看到该对象提供的功能列表。

■ 使用标准协议

相比一般对象而言，Web Services 的接口规范更加规范并且易于机器理解。首先，作为 Web Services 的对象接口所提供的功能应当使用标准的描述语言来描述(如 WSDL)；其次，由标准描述语言描述的服务接口应当能够被发现的，因此这一描述文档需要被存储在私有的或公共的注册库中。同时，使用标准描述语言描述的使用协约将不仅仅是服务界面，它将被延伸到 Web 服务的聚合、跨 Web 服务的事务、工作流等，而这些又都需要服务质量(QoS)的保障。另外，对于松散耦合的系统环境安全机制非常重要，因此我们需要对诸如授权认证、数据完整性、事务处理的不可抵赖性等用规范方法来描述和交换。

■ 可集成能力

由于 Web 服务采取简单的、易理解的标准 Web 协议作为组件界面描述和协同描述规范，完全屏蔽了不同软件平台的差异，无论是 CORBA、DCOM 还是 RMI 都可以通过这一种标准协议进行互操作，实现了在当前环境下最高的可集成性。

1.2 Web Services 带来的机会

1.2.1 电子商务的技术发展

广义上的电子商务是指所有电子化形式的商务活动。目前，电子商务已经触及了我

们生活的各个方面，并且仍在不断地发展之中。电子商务发展的源动力来自电子商务技术的不断创新和演进。20 多年来，电子商务技术三次变革使得电子商务发生了巨大的变化，深刻地改变着我们的生活方式。

电子商务技术第 1 次主要的变革就是 TCP/IP 的诞生。它是网络互联平台建立的基础。这一次变革使得 C/S 计算成为重要和普遍的计算架构。WWW 的到来被看做是第 2 次重要的转变。由 HTML 和 HTTP 带来的 WWW 提供了第 1 个真正开放的用户界面。之后，Java 给我们带来了真正开放的编程语言；20 世纪末，XML 则带来了开放的数据交换标准。其中，分布式对象技术的发展热潮是这些技术发展的集中表现。

分布式对象系统通常是 3 层结构体系的。它有一个面向最终用户的图形用户界面层；有一个中间层，在这个中间层上面运行企业的商业逻辑分布式对象；最后一层是数据库系统，用来集中管理和存储商业数据。

在商业系统中，通常是将商业逻辑封装到商业对象中来。商业模型是易变的，这就意味着商业产品将随着时间推移和不断发展。如果为商业建模的软件能够被封装到商业对象中，它就有可能成为可变的、可扩展、可重用的，而且可以随着商业模型的发展而不断发展。

服务端组件模型定义了开发分布式商业对象的结构。这些商业对象将分布式对象系统的易用性和对象化商业逻辑的易变性结合了起来。服务端组件模型用于中间层的应用服务器上，这些应用服务器管理运行时的组件并使远程客户端能够访问这些组件。这些组件模型提供了最基本的功能，使开发分布式商业对象并将它们融入到商业解决方案中变得非常简单。

服务端组件可以像独立的软件那样买卖。它们遵循标准的组件模型，而且能够在支持这种组件模型的服务器上不经修改而直接运行。假设有这样一个商业系统，它是一些客户、产品、预订和仓库等概念建模的服务端组件组成的。每一个组件都像一个能够与其他组件结合建立一个商业解决方案的积木。产品能够存储在仓库中或者交给客户；客户能够预订或者购买一个产品。你可以组合、分散这些组件，用不同的组合方式来使用组件或者改变组件的定义。一个基于服务端组件的商业系统是非常灵活、具有很好的扩展性的，因为它完全是用一些服务端组件粘合起来的。

1.2.2　Web Services 的技术优势

不论是 DCOM、CORBA 或是 EJB 组件，由于每种组件都必须使用自己特定的规范来开发，组件之间的通信也必须使用特定的协议。这样不同组件之间无法进行直接的数据交换和数据共享。如果一定要在不同的服务端组件之间通信的话，就必须在组件之间加一个特殊的连接器来转换信息。

如图 1-2 所示，EJB 组件想和 COM 组件通信，就需要通过一个 COM 连接器；同样，如果希望和一个 CORBA 组件通信，也需要使用一个特定的 CORBA 连接器。

这样，就使得在不同的服务之间进行数据交换和数据共享变得仍然非常复杂和不易实现，由于每种服务端组件的实现方式都不同，那么每两种不同的组件之间通信都需要使用特定的连接器，而这些连接器并不是通用的，只能是针对某种类型特定的组件。因此，也只是不得已而为之的权宜之计，并没有从根本上解决不同组件之间进行透明的数据共享的问题。

当前电子商务应用对不同服务之间的互操作要求也是越来越重要，特别是像供应链

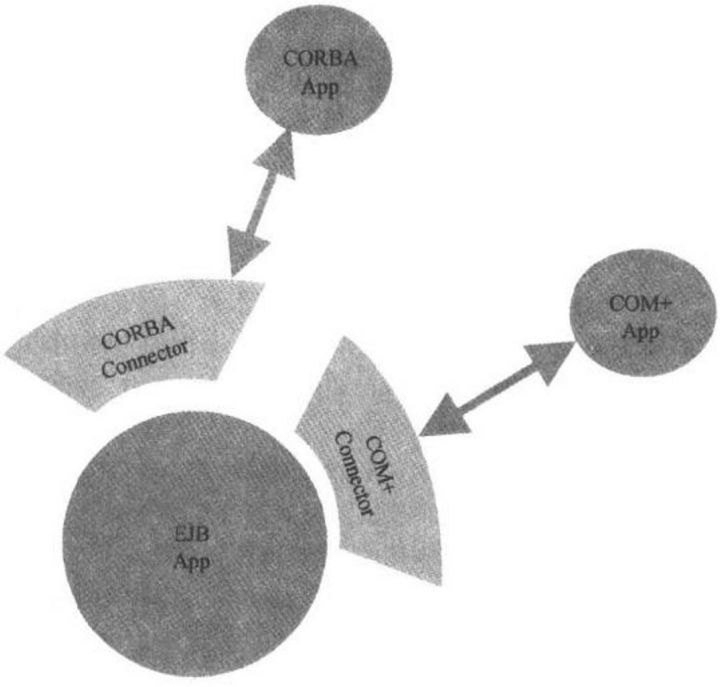


图 1-2 不同组件的通信方式

管理系统(SCM)、客户关系管理系统(CRM),需要将自己的服务系统和客户、供应商的商业系统对接起来,协同工作,实现自动化的数据共享。然而由于不同公司的企业解决方案并不会完全一样,所以实现起来仍然是困难重重。

但是如果所有的服务端组件都以 Web Services 的形式提供服务的话,那么从使用者的角度来看,不管底层是使用 EJB,还是 CORBA,或是 COM,最终面向用户的都是一个统一的 Web Services 调用接口,因为通过 Web Services 统一的接口调用,完全屏蔽了不同服务端组件之间的区别。

如图 1-3 所示,每种服务端组件都提供了统一的 Web Services 界面,因为这些不同的服务端组件之间的数据交换就变成了 Web Services 组件之间的数据交流。

1.2.3 Web Services 带来的市场机会

新的电子商务产生于企业的新需求。企业更需要利用网络各类商务合作伙伴(如供应商、制造商、销售商和客户等)建立更为紧密有效的联系,以降低成本,提高效率,提升企业的核心竞争力,最终获得更多的商业机会。

在过去的几年中,企业内部各部门之间的应用集成已成为人们关注的重点。传统的架构比较脆弱,而且随着规模增长、要求提高、交易量上升以及交易速度的提高,这样的脆弱性日益显现。交互性,尤其是异构分布式系统组件已成为软件工程的主要课题之一。

前几年中,在企业应用集成(EAI)领域对交互性的要求尤其明显。遗憾的是无缝的交互性依然是个梦想,所有当前架构的脆弱性使得这个目标难以实现。脆弱性产生的原因是系统的耦合紧密,对系统的所有层次都产生了依赖性。作为开发者和架构设计师,我