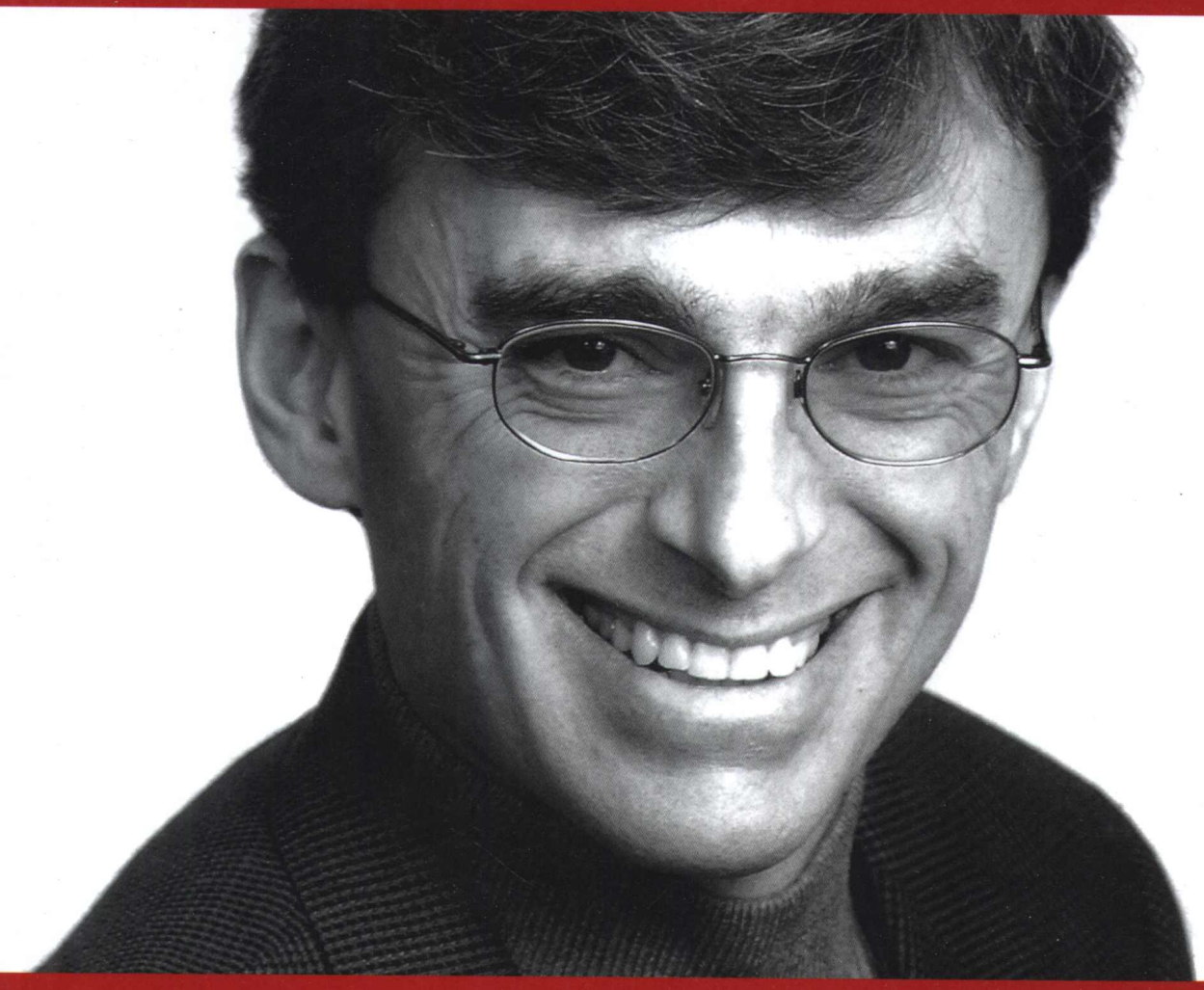
Professional Java SOAP



Java SOAP 编程指南

〔美〕 Henry Bequet 著

魏海萍 于晓菲 毛 选 等译

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

wrox 乐思
PROGRAMMER TO PROGRAMMER™

Professional Java SOAP

Java SOAP编程指南

〔美〕 Henry Bequet 著

魏海萍 于晓菲 毛 选 等译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

致　谢

如果没有许多人的努力和支持，出版这样一本书将是不可能的。因此，首先向在本书写作的过程中给予我帮助但在本致谢中被遗漏的所有人士表示歉意。

特别感谢Sagent Technology股份有限公司的主席和创始人Ken Gardner，不仅感谢他支持本书的构思，而且还感谢他很早就认为XML和SOAP是适合于有性能意识应用系统的可行技术。另外，感谢Ken愿意为本书撰写序。

感谢Sagent公司Boulder办事处日常帮助过我的所有人士，特别是Chuck Bardeen、Derek Brouwer、Rich Dobbs、Tee Parham、Wentao Liu、Jay Bourland和Tom Hayward。

感谢Wrox Press出版公司在本项目中支持我的出版队伍：Chanoch Wiggers、Helen Callaghan和Laura Hall。感谢本书的专业审查人员：Chris Crane、Dave Writz、David Whitney、Jeremy Crosbie、John Davies、Kapil Apshankar、Phil Powers De George、Romin Irani、Steve Baker、Justin Foley、Tee Parham、Jay Bourland和Phil Hanna。

感谢Beach Clow和John Biard，谢谢他们听取我的意见，并同意我参与了基于XML的分布式应用系统：Centrus Real-Time项目。

最后，特别感谢我的家人，Anne-Françoise、Julie和Christophe，他们使这趟旅行很值得。

在此，向所有人员献上我最诚挚的谢意和最深的感激。

译 者 序

随着Internet的迅猛发展和Web在各行各业中的日益普及，以及电子商务、电子政务等的悄然兴起，各种各样的新技术不断涌现，叫得出名字就有几项技术：XML、XSL、XSLT、XML Schema、SOAP、WSDL和UDDI，而SOAP是Web应用程序编程所不可或缺的一个协议，是一个用一种语言与系统独立方式来访问分布式环境中的远程对象与数据的系统。

本书的作者是一家商业情报（BI）公司Sagent Technology 的Internet开发副总裁，并且是一位通过Sun认证的Java 2程序员，擅长于使用XML、SOAP、Java和C++开发可缩放的分布式Web应用系统。

本书从实用的角度出发，围绕一个具体的示例应用系统LeSavon.com，全面而深入地介绍了如何开发与部署基于SOAP的、要求性能的分布式应用系统。作者的意图是让大家分享作者在开发与部署基于SOAP的、要求性能的分布式应用系统中所获得的经验，以帮助大家重复作者的成功。

本书共分3部分：分布式应用协议、示例应用系统以及Web服务。

第一部分包括第1章至第3章。首先，介绍了目前流行的分布式应用协议，并比较它们的优劣。接着，详细分析了SOAP规范，并仔细描述了第2章中要使用的SOAP实现。然后，深入探讨了如何下载并配置必要的软件来安装并运行SOAP服务器。

第二部分包括第4章至第10章。首先，讨论示例应用系统的需求，接着，设计SOAP Server，并编写了开发示例应用系统时将要使用的一个客户程序框架。然后，介绍了一个可能的集成策略，即利用一个基于LDAP的企业级安全系统来集成安全性与个性化，并讨论了增加一个基于规则的高速缓存怎样极大地改善示例应用系统的性能。然后，分析了示例应用系统的性能。最后，介绍了如何给示例应用系统添加基于Web的图形用户界面。

第三部分包括第11章和第12章。首先，正规地定义了Web服务，并解释示例应用程序为什么是一个Web服务。然后，介绍WSDL（Web服务描述语言）新技术。最后，将示例应用系统出版为一个Web服务，并讨论潜在的用户怎么才能查询UDDI注册表来发现该服务。

本书内容丰富，图文并茂，组织合理，示例实用，定能满足国内软件开发人员和爱好者的需求以及促进其软件开发水平的提高。无论是初来乍到的新手，还是富有经验的老手，相信都能从本书中受益。

本书由毛选、魏海萍和于晓菲翻译，并得到王瑞东、乌鹤柏、盛毅、郭得利、丁健和丁明的帮助，全书由毛选统稿和审校。译者在翻译过程中力求信、达、雅，但由于水平有限且时间仓促，错误与不妥之处在所难免，恳请广大读者谅解，并欢迎批评指正。

作 者 简 介

Henry Bequet

Henry Bequet是一家商业情报（BI）公司Sagent Technology的Internet开发副总裁。他是一位通过Sun认证的Java 2程序员，擅长使用XML、SOAP、Java和C++开发可缩放的分布式Web应用系统。

Henry出生于多雨的比利时，但现在和他的妻子及两个孩子生活在阳光灿烂的科罗拉多州的Boulder市，在这里，他享受着软件开发、滑雪和溜冰的乐趣。

简　介

我涉足分布式XML应用系统，是从1999年开发一个针对销售行业的基于文档的Web应用系统开始的。如果不考虑与领域相关的那些问题，该应用系统的需求可以归结为下列几项：

- 该应用系统或服务必须是开放的：运行在任一平台上的、用任一编程语言写成的客户程序必须能够访问该服务。
- 该应用系统在Internet上必须是可访问到的，并且客户程序必须能够通过防火墙访问该服务。
- 该应用系统必须是每年365天、每周7天和每天24小时可使用的。
- 该应用系统必须安全地加密通过Internet传输的数据。
- 该应用系统必须能够升级为每年处理至少几十亿个事务。

根据我以前开发Web应用系统和XML的经验，这些需求意味着HTTP上的XML：

- XML和HTTP都是标准协议，它们在当今正使用着的所有平台上都得到了实现。
- 标准HTTP通信通常被允许通过防火墙，至少系统管理员非常熟悉HTTP，并能配置它来满足他们的安全要求。
- HTTP的无状态特性使开发人员能够轻松地复制计算机来获得最大可使用性。
- HTTPS规定了一种实现起来容易、可广泛使用而又安全的加密机制。
- HTTP的无状态特性提供了直截了当的可缩放性：当服务器上的负载变大时，可以增加新的计算机。

那时，已经有大量介绍XML作为Web出版工具的图书，但很少有人使用XML和HTTP作为一个分布式应用协议。与该题目相关的绝大部分可用信息必须到专业杂志和Internet新闻组上才能找到。如果那时能够读到一本探讨使用XML作为一种编码机制和使用HTTP作为一种传输工具的分布式体系结构的图书，我的工作就会得到极大的简化。可惜的是，我未能找到这样一本书。

在上述应用系统被发布和部署之后，我开始从事一个Business Intelligence（BI，商业情报）应用系统，该系统涉及到一个基于Web的应用编程接口（API）的开发，而且它的需求与我们的开发队伍刚刚完成的销售应用系统相似。实际上，我们使用了一个相似的体系结构，但是该应用系统不是基于文档的，而是基于过程的。

这也是我发现SOAP（Simple Object Access Protocol，简单对象访问协议）的地方。SOAP是一个用一种与语言和系统无关的方式访问分布式环境中的远程对象与数据的系统。我也遇到了XML-RPC，这是一个有点重叠的规范，它定义了使用XML在Internet上完成远程过程调用的方式。

我立即开始寻找与XMP-RPC和SOAP有关的图书，但是我对自己找到的东西感到很失望。我未能发现有任何一本书介绍了在要求性能的生产性环境中进行SOAP开发时应注意的警告与缺陷。虽然有两本书介绍了XML-RCP或SOAP，但是就一个现实的、可缩放的、安全

的应用系统将面对的那些挑战而言，这两本书并未解开我的开发队伍和我本人心中的谜团。在这个基于SOAP的应用系统的第一个α版本发布之后，我认为其他人将会从我们的经历中受益，因此我就与Wrox Press出版公司联系，打算写一本关于SOAP的实用图书，而几周后，我就开始忙于把这一想法变成现实。

我们将在本书中介绍许多新技术，如：XML、XSL、XSLT、XML Schema、SOAP、WSDL和UDDI。由于本书是一本实用图书，所以根据作者使用这些技术的经验，大家在阅读本书的过程中将会发现许多对自己有帮助的示例。除了简单的介绍性示例之外，我们将围绕着一个示例应用系统来组织本书，它就是LeSavon.com。试图用一个示例应用系统来模仿现实经验有两大缺点：一是增加了复杂性，二是开发一个关键系统时必须考虑的多余细节。我们已经在写作过程中尽可能地避免这两个缺陷，但我们不敢说已完全取得了成功。我们希望得到大家的谅解，并希望大家记住：我们的最终目标是帮助大家重复我们的成功。

由于本书的任务是让大家分享我们在部署基于SOAP的、要求性能的分布式应用系统中所获得的经验，因此大家找不到示例应用系统的开发中所使用的任何α和β产品。但是，由于Web服务领域还不成熟，并且正处于快速发展之中，所以大家将会发现关于对没有生产价值的工具与实用程序的讨论。我们已十分小心不要让示例应用系统的关键方面依赖这些工具。试验工具的一个不错的例子是IBM的Web Service Toolkit（WSTK）在本书中的使用，我们将使用该工具来加速WSDL文档的开发，但是也可以通过手工编写这些WSDL文件来绕过该工具。

在本书Web服务的开发与部署中，我们将只使用可免费和容易获得的工具。正如大家在本书中将会发现的，这一限制在任何方面都不会约束我们的创造性和工作效率。

Ken Gardner，Sagent Technology公司主席和创始人

阅读本书的条件

在本书的正文中，我们假设了一些必须具备的知识：

- 面向对象概念的知识，比如对象、类、覆盖与重载方法以及接口。
- 一般性Java开发与Java服务器端开发（Servlet和JSP）的知识。
- 对Internet及其使用的了解将有助于理解本书的内容，但不是必需的。

本书将在第2章中简要介绍HTTP和XML。

本书的内容组织

本书被组织成3部分：分布式应用协议、示例应用系统以及Web服务。

在第1章“分布式应用协议”中，我们将介绍目前流行的分布式应用协议，并比较它们的优劣。然后，详细分析SOAP规范，并较仔细地看一看我们将要在第2章“SOAP”中使用的SOAP实现。

在第3章“安装SOAP服务器”中，我们将深入探讨下载并配置必要的软件来安装并运

行SOAP服务器的细节。

从这里开始，我们进入本书的第二部分。首先讨论示例应用系统的需求（第4章），接着设计SOAP Server，其中我们设计并实现一些SOAP服务（第5章）。在第6章中，我们编写一个客户程序框架，该框架在我们用用户界面包装示例应用系统的开发时将会派上用场。

如果不涉及安全性与个性化，应用系统开发的讨论就是不完整的。在第7章“安全性与个性化”中，我们将介绍一个可能的集成策略，即利用一个基于LDAP的企业级安全系统来集成安全性与个性化。第8章“高速缓存技术”将讨论增加一个基于规则的高速缓存怎样极大地改善示例应用系统的性能。

我们还将看一看示例应用系统的性能（第9章），进而分析示例应用系统来验证我们在前几章中所做的设计与实现选择是有效的。最后，第10章“Web应用程序”通过给示例应用系统添加一个基于Web的图形用户界面（GUI）来结束本书的第二部分。

在本书的第三部分中，我们正式地定义Web服务，并发现示例应用程序确实是一个Web服务。第11章“WSDL”向大家介绍Web Service Description Language（WSDL，Web服务描述语言），这是一项对软件组件在万维网上的互操作性有促进作用的技术。在第12章“UDDI”中，我们将示例应用系统出版为一个Web服务，并讨论潜在的用户怎么才能查询UDDI注册表来发现我们的服务。

约定

为了帮助大家从正文中获取最大的信息量和跟踪正在发生的事情，我们在整本书中使用了一些约定。

例如：

这表示与周围正文有直接关系的重要而又不能忘记的信息。

而这种格式用于对当前讨论的旁注。

如何下载本书的示例代码

当大家访问Wrox站点http://www.wrox.com/时，请直接通过我们的Search功能查找书名，或者使用书名清单之一。单击Code栏中的Download，或本书详细信息页面上的Download Code。

从我们的站点上可下载到的那些文件已经使用WinZip实用程序做过压缩。当大家把这些附件保存到自己的硬盘上时，需要使用WinZip或PKUnzip之类的解压缩程序抽取这些文件。当抽取这些文件时，代码通常被抽取到章文件夹中。当开始抽取过程时，要保证大家的软件（WinZip、PKUnzip等）已被设置成使用文件夹名。

p2p.wrox.com

要想参加与作者和同伴的对等讨论，请加入P2P邮件清单。除了我们的一对一电子邮件支持系统之外，我们的特有系统还提供了关于邮件清单、论坛以及新闻组的programmer to

programmer™联系人。如果把一个查询张贴到P2P上，大家可以确信许多在场的Wrox作者和其他业界专家正在分析它。不仅在阅读本书期间，而且在开发自己的应用系统时，大家都可以在p2p.wrox.com上找到许多对自己有帮助的不同清单。特别适合本书的是j2ee和pro_java_server清单。

要想预订一个邮件清单，只需按照下列这些步骤操作即可：

1. 转到http://p2p.wrox.com/。
2. 从左菜单栏中选择合适的类别。
3. 单击希望加入的邮件清单。
4. 按照说明预订并填写你的电子邮件地址与密码。
5. 回复你所接收到的确认邮件。
6. 使用预订管理器来加入更多的清单，并设置你的电子邮件选择参数。

本系统为什么提供最佳支持

大家可以选择加入那些邮件清单，也可以把它们作为一个周刊性的文摘来接收。如果大家没有时间或便利工具来接收邮件清单，可以搜索我们的联机存档文件。无用而过时的邮件已被删除，而且大家的电子邮件地址受独特的Lyris系统保护。关于加入或离开清单的查询以及关于清单的其他任何普通查询都应该发给listsupport@p2p.wrox.com。

目　录

第1章 分布式应用协议

本章将简要介绍分布式应用协议（如CORBA和DCOM）及其发展历史。如果已经熟悉这些协议，并希望直接了解SOAP，可以跳过本章，直接阅读第2章来学习SOAP规范。

本章不提供任何代码，只是进行一般性的讨论。如果有兴趣较详细地了解本章所讨论的协议，则可以访问本章中所提供的Internet链接。

本章并不打算详尽地描述CORBA或DCOM，而是从较高的层面上看一看这些协议，目标是让大家对此技术的发展状况有一定的了解，并对在分布式应用系统环境中定义应用系统体系结构的得与失有初步的了解。

同大多数设计决策一样，为分布式应用系统选择一种通信协议也是一种权衡得失的行为，即在应用系统的需求与在开发时间、系统性能、开放度等方面做多大的牺牲之间进行选择。例如，如果开发人员正在设计某个专用网络上的一个实时应用系统，SOAP可能就不是他们的最佳选择。相反，如果开发人员正在开发一个企业到企业的综合工具，以便让它在因特网上提供数据净化服务（地址、拼写等的验证），SOAP可能就是一个不错的选择。还有其他一些例子。本章的后面将进一步描述这些折衷方案。

本章的其余部分将对下列这些协议做一番简要分析：

- CORBA
- COM/DCOM/COM+
- RMI
- XML-RPC
- SOAP
- ebXML

文档与过程

在开始分析上述主流分布式应用协议之前，需要先来看一看往返于远程计算机之间的那些消息中所含的有效负荷的语义。

分类分布式应用有许多种方法，经常使用的一种分类法是基于信息包的内容类型。面向过程的有效负荷，也称做Remote Procedure Call（RPC，远程过程调用），含有一个过程、函数或方法调用的表示或编码。面向文档的有效负荷含有一个描述数据结构而非过程调用的文档。此类文档的一个例子就是产品或服务的购买者发送给供应者的定单。

大家可以争辩说它们只是理论上存在差别，因为这两种负荷的内容彼此重叠。一方面，面向文档的负荷类似于面向过程的负荷，因为大家可以把一个串行化的方法调用表示为一个数据结构。实际情形正是如此，大家在第2章中就能见到SOAP把方法调用编码为一个结构。另一方面，面向过程的负荷可以含有基于文档的负荷，因为一个文档的内容可以被作为一个

变元来传递。本书将把讨论的重点放在RPC负荷上。

下面首先从最流行的分布式对象协议之一CORBA开始分析分布式协议。

CORBA

Common Object Request Broker Architecture（CORBA，公用对象请求调度程序体系结构）是由Object Management Group（OMG，对象管理组）于1990年首先提出来的。它是一个针对在网络上分布对象的规范。OMG是一个联盟，由代表了计算领域内所有大型公司的800多家公司和数百家中小型公司所组成。除了CORBA规范之外，OMG还负责其他各种技术，比如Unified Modelling Language（UML，统一建模语言）或Common Warehouse Metamodel（CWM，公用仓库元模型）。

CORBA是一个针对客户机/服务器体系结构的规范。正如图1.1中所显示的，客户机通过一个称做Object Request Broker（ORB，对象请求调度程序）的运行时组件来调用服务器上的远程对象。

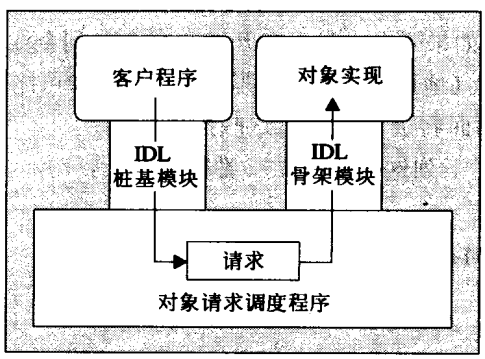


图1.1

图1.2显示了对象的远程调用涉及到使用Internet Inter-ORB Protocol（IIOP，Internet ORB间协议）的ORB到ORB通信。IIOP是General Inter-ORB Protocol（GIOP）的TCP/IP串行化，而GIOP是一个没有定义传输映射的抽象协议。其他协议也可以使用，但为了OMG兼容性，需要使用一个基于GIOP的协议。

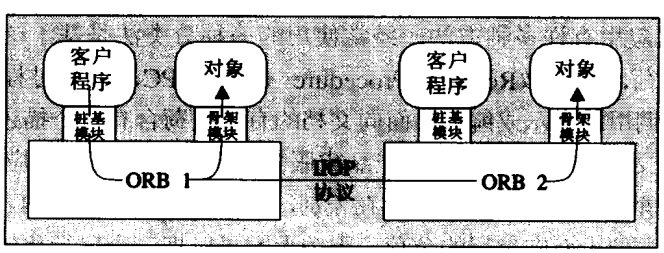


图1.2

图中的那些对象使用Interface Description Language（IDL，接口描述语言）来描述，这是一种以C++为模型的语言。目前，也有使提供与调用远程对象的过程变得更容易的语言联编工具。IDL的编译产生代理设计模板的一个实现，这个实现也称做客户机桩基模块（Stub）或对象骨架程序（Skeleton），它们由客户机和CORBA服务的提供者使用。

针对CORBA的、尤其与介绍Java开发的图书有关的一个语言联编工具是Java IDL。Java IDL能让Java开发人员以一种与平台无关的方式访问CORBA。使用Java IDL，用Java写成的分布式应用系统能够使用支持CORBA的服务来调用远程服务器上的对象。同任何一个CORBA实现一样，Java IDL也包含了像Java Object Request Broker（Java对象请求调度程序）之类的运行时组件。

大家可以在Object Management Group网站（http://www.omg.org）上找到关于CORBA的进一步信息。大家也可以访问CORBA网站本身（http://www.corba.org）。

CORBA在非UNIX平台上的可接受性一直受到限制，主要原因是存在下一节将要介绍的COM/DCOM。

COM/DCOM

在开发Windows的32位版本期间，Microsoft公司一直在寻找一项技术，以便开发人员能够重用不同开发小组用不同语言所编写的软件组件。CORBA不可能是这项技术，部分是政治和营销方面的原因，但也有技术方面的原因：Microsoft公司的开发人员需要一项当所有组件都在同一个进程中时不会给一个直接的C++调用招致任何额外开销的技术。Microsoft公司的工程师们选择了从CORBA那里借用关键概念（其中包括一个运行时环境和IDL语言），同时又没有给进程中调用增加任何额外开销的Component Object Model（COM，组件对象模型）。1997年，提出了COM的网络化版本：Distributed COM（DCOM）。

COM+是COM和DCOM的细化。COM+的主要贡献是引入了简化COM开发人员工作的运行时服务。

在COM及其派生技术中，一个对象就是一个软件组件，而且该组件至少支持一个叫做IUknown的接口。该对象可以用一个DLL或一个可执行文件来创建，而这个IUknown接口含有三个方法，其中两个方法用来控制着一个对象的生存期（AddRef()和Release()），而第三个方法用来发现该对象在运行期间内的能力（QueryInterface()）。正如图1.3中所显示的，在客户机与服务器处于同一个进程中时，COM不需要任何运行时环境。

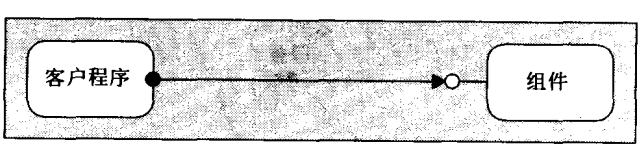


图1.3

但是，如果进程或计算机之间有通信，体系结构就类似于CORBA。对于同一台客户计算机上的进程到进程通信来说，调用通过COM运行时组件来进行，而串行化则由Local Pro-

cedure Call（LPC，本地过程调用）协议来处理。

Distributed Computing Environment（DCE，分布式计算环境）为分布式计算提供服务，其中包括一个用来使时钟保持紧密同步的分布式时间系统。

那些远程调用是相似的，而且LPC组件正由DCOM网络协议来取代，如图1.4中所示。

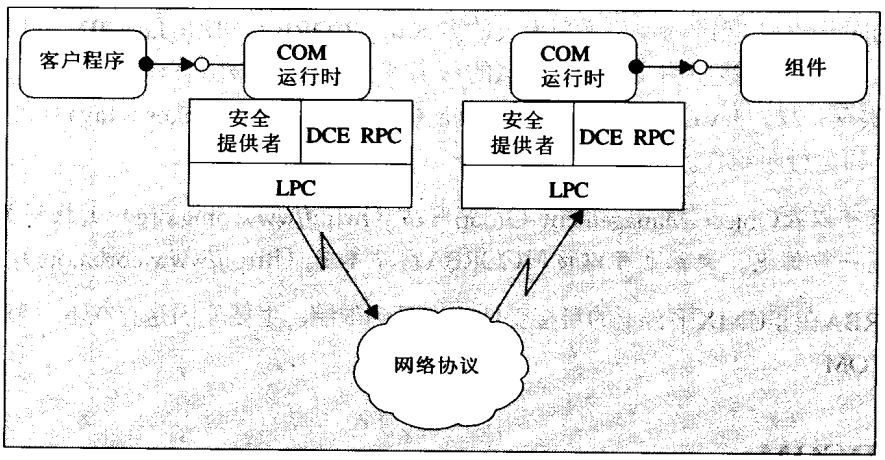


图1.4

由于COM和DCOM从CORBA那里继承了它们的基础，所以它们有相似的得与失：它们都是完整而复杂的协议，而且同样需要显著的计算与联网威力。由于它是一个二进制标准，所以DCOM是语言无关的（有一个Java 1.X实现），但事实上，它只能与Windows平台上的C++和Visual Basic对象一起使用。同CORBA一样，DCOM的学习难度较大。

关于COM的进一步信息可以在http://www.microsoft.com/com/tech/com.asp网页上找到。

RMI

Java的Remote Method Invocation（RMI，远程方法调用）提供Java对象的远程调用。对象的定义与实现完全在Java的范畴内。这种专一性提供了一种能让Java开发人员快速实现的一流解决方法。它的主要弱点是缺乏对其他分布式协议的开放性。这种有限的互操作性常常是RMI在企业内难以获得成功的一大障碍。

在RMI中，客户机与服务器使用同一个Java接口的两种不同实现，如图1.5中所示。

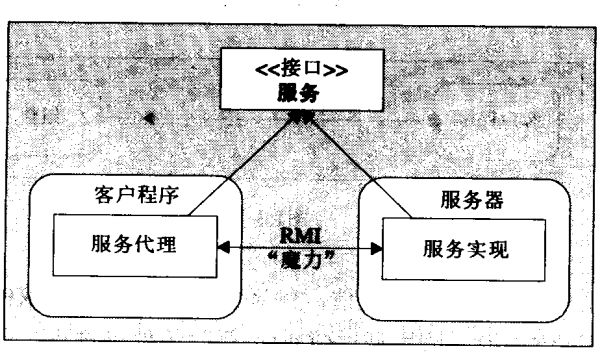


图1.5