

Web

服务深入编程

孙永强 编著



清华大学出版社
<http://www.tup.com.cn>

Web

服务深入编程

孙永强 编著



清华大学出版社
<http://www.tup.com.cn>

(京)新登字 158 号

内 容 简 介

XML Web 服务将在极大程度上影响今后网络技术和软件开发技术的发展,它是微软公司.NET 战略的核心。

本书主要介绍如何使用 Visual C#.NET 来开发 ASP.NET XML Web 服务。首先,本书介绍了 XML Web 服务的基本概念,并带领读者逐步创建 XML Web 服务和客户程序,然后逐个介绍 XML Web 服务中使用的各种关键技术以及 ASP.NET 提供的功能,以及如何在 XML Web 服务中使用这些技术和功能。最后本书创建了一个通用的 XML Web 服务客户程序,用户可以使用它来对 XML Web 服务进行测试。本书由浅入深地介绍了各种概念,同时提供了大量的示例程序,通过本书的学习用户可以轻松掌握 ASP.NET XML Web 服务这种全新的应用程序开发方式。

本书主要面向初中级用户,也可作为 Web 服务编程爱好者的技术指导书及专业 Web 服务编程人员的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Web 服务深入编程/孙永强编著. —北京:清华大学出版社, 2002
ISBN 7-302-06061-4

I. W... II. 孙... III. 因特网—程序设计 IV. TP393.4

中国版本图书馆 CIP 数据核字(2002) 第 086546 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn>

责任编辑: 杨海儿

印 刷 者: 北京昌平环球印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 25 **字数:** 593 千字

版 次: 2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷

书 号: ISBN 7-302-06061-4/TP·3616

印 数: 0001~4000

定 价: 38.00 元

前言

XML Web 服务是 IT 业界公认的能够影响今后网络发展的一项技术。XML Web 服务从某个角度来说，正是要解决分布式计算的问题。如果说 Windows 是操纵一台计算机的灵魂，那么 XML Web 服务则是今后网络的灵魂。总之，XML Web 服务将彻底改变软件产业的格局。

XML Web 服务是微软公司 .NET 战略的核心。XML Web 服务作为新兴的一种依赖于 Internet，为用户或其他 XML Web 服务提供单一服务功能的组件，将成为未来软件的存在形态。

当 ASP 出现之后，“软件就是服务”这个真理终于被人们认可了。但现实就是这么残酷，曾经被认为是继电子邮件出现之后，对企业用户影响最大的应用——ASP，今天却面临风雨飘摇的窘境。究其原因，主要是 ASP 仍然是一个集中式计算模型的产物，只不过披着一层互联网的外衣。但 XML Web 服务却迥然不同，它秉承“软件就是服务”的真理，同时顺应分布式计算模式大行其道的潮流。它的存在形式与以往软件不同，这种组件模式小巧、单一，开发成本较低。

XML Web 服务在某种意义上就是结合了 ASP 和组件产品两方面的特性，它通过标准协议在互联网上提供单一特定的服务。两者的区别在于 XML Web 服务面对的用户群更广，而且更加通用。而 ASP 服务的对象则需要签订一定的合作协约。XML Web 服务的出现，给软件产业又一次带来新的商业模式。XML Web 服务的供应商，既可以将 XML Web 服务一次卖断给软件开发商，也可以通过租赁的方式按月收取服务费用。试想一下，如果有人开发的 XML Web 服务被微软公司选中应用在 Windows XP 平台里，那他很有可能将一夜成为百万富翁的美梦变为现实。

XML Web 服务就是通过标准的 Web 协议可编程访问的 Web 组件。如果不进一步地加以解释，没有多少人能够明白它讲的是什么。当用户使用 Word 编写报告的时候，需要调用一个 Excel 制作好的报表，这时可以使用对象嵌入功能。在执行这个操作的时候，实际上是在使用 Word 的过程中去调用另外一个应用程序。而 XML Web 服务的功能也与此十分相近。只不过它不是发生在一个单一的 PC 机上，而是发生在客户端与服务端之间或是服务器与服务器之间。

例如浏览一个英文网页，随着鼠标指针的移动，屏幕上显示出相应的中文解释，不要以为这个用户正在使用金山词霸，他只不过是通过网络调用了一个提供即时翻译功能的 XML Web 服务。此时的即时翻译程序，变成了一个远在异地的应用组件。这些基于后端的 XML Web 服务被微软公司认为将会成为未来互联网的主导，而抢占了这个阵



地也就赢得了互联网的明天。由此引发出很多新的标准、协议，以便让这些 XML Web 服务组件能够顺利地调用。

其实 XML Web 服务是一次分布式计算体系的跃进，是分布式计算模式对集中式计算模式的胜利。它强调的是不同组件协同工作来为用户提供服务。分布式对象结构是将标准化的组件对象放在远端的计算机上，客户端在调用对象时使用分布式对象结构的调用标准来获取对象。在这里人们要清楚一点，此时的客户端内涵已经被扩大，变为任何一个对调用对象提出服务请求的组件。面对互联网这个广义的分布式计算体系，XML Web 服务实际上更像是一种远程访问的标准。

CORBA 曾经是最热门的分布式对象结构。它的优势就在于可以跨平台、跨开发语言来调用服务器端某个对象模块提供的服务。XML Web 服务和 CORBA 这些分布式应用技术的目的都是要解决远程目标之间的通信问题。和其他的解决方案不同，XML Web 服务提供基于开放式标准上的完全终端对终端的解决方案。对于需要 XML Web 服务解决方案的用户来说，没有任何特殊的要求。

面对 XML Web 服务这样的发展前景，微软公司坚信打赢 XML Web 服务这场战斗将确立在整个互联网中的霸主地位。为了能够实现这个理想，微软公司做出了很多的努力。首先是力推 XML，因为 XML Web 服务所要涉及到的信息已经不仅是简单的文本，而更多的是数据。此外 XML Web 服务需要跨平台，原有的 HTML 是无法满足需要的，通过 XML 可以使程序之间更容易进行沟通。

微软公司对 XML Web 服务的核心 SOAP 也颇费心机。SOAP 是对象间信息交换的通信协议，可以把它看成是用户端与服务器端之间进行沟通的特殊语言。在按照 SOAP 协议封装的信息里，包含了用户端申请 XML Web 服务所必要的内容，例如，该 Web Service 所涉及到的名称、参数等。当然服务器端也会按照 SOAP 通信协议返回相应的结果。

但新的一个问题又出现了，就是用户端的程序如何才能知道 XML Web 服务提供了什么可以进行调用的服务呢？这时 WSDL(Web Service Description Language, Web 服务描述语言)出现了，它是用来描述 XML Web 服务的相关信息。现在一切就绪，可那些 XML Web 服务开发商还需要一个方法将自己开发的 XML Web 服务进行发布。于是 UDDI(Universal Description, Discovery and Integration)应运而生。UDDI 是一个跨产业、跨平台的开放性架构。它可以帮助 XML Web 服务开发商在 Internet 上发布自己推出的 XML Web 服务。

针对开发 XML Web 服务应用，微软公司推出了 Visual Studio .NET 集成开发工具。在 Visual Studio.NET 平台下，用户可以利用 Visual Basic、Visual C++和 Visual C#等多种开发工具共同开发各种符合 XML Web 服务标准的应用，并且这些应用可以保持相互兼容。Visual C#.NET 是微软公司针对 XML Web 服务推出的一种全新的开发语言，而 ASP.NET 则提供了支持 XML Web 服务的全新基础框架。将这两者结合使用，用户可

以简单方便地开发出功能强大的 XML Web 服务。

本书主要介绍如何使用 Visual C#.NET 语言并充分利用 ASP.NET 的强大功能来开发 XML Web 服务,同时还介绍了各种关键技术,比如 WSDL、SOAP 和 UDDI 等的应用。本书要求读者对 XML 和 HTTP 等 Internet 标准规范和协议有一些基本的了解。关于这些内容,读者可以参考作者的另外一本书《Visual C# .NET 中文版 Web 服务开发基础》。在该书中,详细地介绍了一些常用的 Internet 标准规范和协议。

本书主要包含以下内容:

第 1 章概括介绍 XML Web 服务的概念以及与之相关的各种技术。

第 2 章带领用户逐步创建一个简单的 ASP.NET XML Web 服务

第 3 章介绍如何创建用于访问 XML Web 服务的不同类型的客户程序,包括 Windows 窗体客户程序和 Web 窗体客户程序。

第 4 章介绍 XML Web 服务的描述语言——WSDL。

第 5 章介绍如何使用 WSDL 来进行接口方式化的开发。

第 6 章介绍 XML Web 服务所使用的标准通信协议。

第 7 章介绍如何定制 SOAP 消息以使之满足特殊的需求。

第 8 章介绍如何使用 SOAP 扩展来在 SOAP 消息的处理过程中添加处理功能。

第 9 章介绍 XML Web 服务的标准发现机制——UDDI,并实现一个简单的 UDDI 浏览器。

第 10 章介绍如何在 XML Web 服务的方法中使用各种不同数据类型的参数和返回值。

第 11 章介绍如何在 XML Web 服务中充分利用 ASP.NET 提供的强大功能。

第 12 章介绍如何在 XML Web 服务中使用各种安全技术。

第 13 章创建一个通用的 XML Web 服务客户程序,它只需知道一个 WSDL 文档的地址就可以访问任意的 XML Web 服务。

本书除了封面署名的作者外,其他参加本书制作的人员有吴涛、王卫国、宋艳华、陆波、杨丽坤、周涛、庞爱民、王军、郭颖浩和吕伟臣等。由于 XML Web 服务是一种全新的技术,作者掌握的水平有限,所以在本书的介绍中难免会存在不足之处,恳请读者批评指正。

孙永强

2002 年 9 月

目 录

第 1 章 概述	1
1.1 XML Web 服务简介	1
1.2 XML Web 服务的基础结构	3
1.3 XML 标准	4
1.4 SOAP 协议	9
1.5 XML Web 服务的目录——UDDI	12
1.6 描述 XML Web 服务——WSDL	14
1.7 Global XML Web 服务体系结构	16
1.8 小结	18
第 2 章 使用 Visual Studio.NET 创建 ASP.NET Web 服务	19
2.1 ASP.NET XML Web 服务项目的构成	21
2.2 ASP.NET XML Web 服务的代码模型	24
2.2.1 入口点文件	24
2.2.2 后台代码文件	25
2.3 测试 ASP.NET XML Web 服务	27
2.4 WebService 类	30
2.5 在 XML Web 服务类中使用属性	32
2.5.1 使用 WebService 属性	32
2.5.2 使用 WebMethod 属性	34
2.6 配置 XML Web 服务	40
2.7 部署 XML Web 服务	41
2.7.1 通过创建安装项目来部署 XML Web 服务	41
2.7.2 通过复制项目来部署 XML Web 服务	45
2.8 小结	46
第 3 章 创建访问 Web 服务的客户程序	47
3.1 创建 Windows 窗体客户程序	47
3.1.1 使用模板创建新项目	47
3.1.2 创建代理类	48
3.1.3 在客户程序中通过代理类来访问 XML Web 服务	52
3.2 创建 ASP.NET 客户程序	59



3.3	查看代理类	64
3.4	使用异步方式访问 XML Web 服务	71
3.4.1	使用回调函数	72
3.4.2	使用 WaitHandle 类	74
3.5	小结	77
第 4 章	XML Web 服务描述语言	78
4.1	WSDL 文档的结构	78
4.1.1	definitions 元素	78
4.1.2	types 元素	79
4.1.3	message 元素	82
4.1.4	portType 元素	85
4.1.5	binding 元素	87
4.1.6	service 元素	90
4.2	应用 WSDL 文档	91
4.3	使用属性定制 WSDL	100
4.3.1	WebServiceAttribute 属性	107
4.3.2	WebMethodAttribute 属性	108
4.3.3	WebServiceBindingAttribute 属性	110
4.4	使用配置文件改变 WSDL	113
4.5	小结	114
第 5 章	基于接口的 XML Web 服务编程	115
5.1	定义接口	116
5.2	实现接口	122
5.2.1	创建接口类	123
5.2.2	实现接口	125
5.3	创建使用接口的客户程序	128
5.4	定义多个接口	132
5.5	小结	137
第 6 章	XML Web 服务的通信协议	138
6.1	HTTP 协议	138
6.1.1	HTTP 消息的格式	138
6.1.2	HTTP GET 方法	140
6.1.3	HTTP POST 方法	141
6.1.4	示例程序	143

6.2	SOAP 协议	148
6.3	SOAP 协议与 HTTP 协议	152
6.4	在 ASP.NET XML Web 服务中使用 SOAP Header	158
6.4.1	SoapHeader 类和 SoapUnknownHeader 类	159
6.4.2	SoapHeaderAttribute 属性类	160
6.4.3	在 XML Web 服务中实际使用 SOAP Header 元素	160
6.4.4	创建使用 SOAP 报头的客户程序	166
6.5	小结	169
第 7 章	定制 SOAP 消息	170
7.1	SOAP 消息的编码样式	170
7.1.1	document/literal	171
7.1.2	document/encoded	172
7.1.3	rpc/encoded	173
7.2	使用属性定制 SOAP 消息	174
7.2.1	SoapDocumentServiceAttribute 属性	175
7.2.2	SoapRpcServiceAttribute 属性	179
7.2.3	SoapDocumentMethodAttribute 属性	180
7.2.4	SoapRpcMethodAttribute 特性	186
7.3	使用 XML 序列化定制 SOAP 消息	188
7.3.1	使用属性控制 XML 序列化	188
7.3.2	XML Web 服务的 XML 序列化	193
7.4	小结	198
第 8 章	使用 SOAP 扩展	199
8.1	SOAP 扩展概述	199
8.2	创建 SOAP 扩展属性类	201
8.3	创建 SOAP 扩展类	203
8.3.1	GetInitializer 方法	204
8.3.2	Initialize 方法	205
8.3.3	ChainStream 方法	205
8.3.4	ProcessMessage 方法	207
8.4	使用 SOAP 扩展	209
8.4.1	使用 SOAP 扩展属性为方法指定 SOAP 扩展	209
8.4.2	使用配置文件指定 SOAP 扩展	209
8.5	示例程序	210



8.5.1 创建使用 SOAP 扩展的 XML Web 服务	211
8.5.2 创建客户程序	225
8.5.3 为客户程序添加 SOAP 扩展	227
8.6 小结	229
第 9 章 XML Web 服务的发现	230
9.1 UDDI 概述	230
9.1.1 UDDI 商业注册表	231
9.1.2 UDDI 接入节点	232
9.2 UDDI 的数据结构	232
9.2.1 tModel 结构	233
9.2.2 bindingTemplate 结构	234
9.2.3 businessService 结构	235
9.2.4 businessEntity 结构	236
9.3 UDDI 的 API	238
9.3.1 查询 API	239
9.3.2 发布 API	240
9.4 从 UDDI 注册表中查询信息	241
9.4.1 使用 Microsoft 接入节点提供的界面进行查询	241
9.4.2 编程查询 UDDI 注册表	248
9.5 向 UDDI 注册表中发布信息	264
9.5.1 确定要注册的信息	264
9.5.2 向 UDDI 中添加注册项	265
9.6 小结	272
第 10 章 XML Web 服务中的数据类型处理	273
10.1 使用 XML 的内建数据类型	273
10.2 使用枚举类型	276
10.3 使用结构类型	278
10.4 使用数组	281
10.5 使用二进制数据	282
10.6 使用 DataSet	283
10.7 使用自定义复杂类型	287
10.8 小结	291
第 11 章 充分利用 ASP.NET 的功能	292
11.1 状态管理	292

11.1.1	会话状态	292
11.1.2	应用程序状态	300
11.2	使用缓存	304
11.2.1	使用缓存时的注意事项	304
11.2.2	使用缓存的方式	305
11.2.3	ASP.NET 提供的缓存支持	306
11.2.4	示例程序	310
11.3	使用事务	317
11.4	小结	319
第 12 章	创建安全的 XML Web 服务	320
12.1	基础结构中的安全性	320
12.2	身份验证与授权	329
12.2.1	配置身份验证	330
12.2.2	Windows 身份验证示例程序	332
12.2.3	客户证书身份验证	338
12.2.4	XML Web services 的授权选项	342
12.3	使用 ASP.NET 模拟	344
12.4	SOAP 安全性	347
12.4.1	XML 签名	348
12.4.2	XML 加密	350
12.4.3	SOAP 安全性	352
12.5	小结	358
第 13 章	通用 XML Web 服务客户程序	359
13.1	设计思路	359
13.2	程序设计	360
13.2.1	动态生成程序集的类型	360
13.2.2	输入方法参数的对话框	373
13.2.3	主窗体的设计	376
13.3	小结	384

第1章 概 述

近年来，随着 Internet 的蓬勃发展，它已渗透到了各个行业和领域中，逐渐成为一种新的应用平台和软件开发环境。另外，软件开发技术和 XML 等技术也取得了长足进步，所有这些促成了 XML Web 服务技术的出现。XML Web 服务是 XML 的一个实际应用，是 Internet 环境中的分布式组件开发技术。正是有了 XML Web 服务技术，当前的 Internet 才从“可浏览 Web”发展成为“可编程 Web”。

XML Web 服务是 Microsoft .NET 策略的核心，也是 .NET Framework 的关键组件。XML Web 服务是 Web 分布式应用程序的基本构建块，同时也构成了 Microsoft .NET 的可编程 Web 理念的基础。

1.1 XML Web 服务简介

简单地说，XML Web 服务就是一种可以通过 URL 地址来访问的网络资源。具体地说，XML Web 服务是一种完全建立在 XML 等 Internet 标准协议或规范上的应用程序，客户程序可以通过 HTTP 和 SOAP 等标准协议来访问它。从上面的描述中可以看出，XML Web 服务首先是一种提供服务的应用程序逻辑；其次它建立在被广泛接受的标准协议之上，因此它可以被任意的系统和开发语言所支持；最后，它主要由程序代码使用，而不是最终用户。

众所周知，Internet 是一个异构的环境。在 Internet 中，存在着各种各样的硬件设备和操作系统，运行着使用不同语言、不同技术开发的应用程序。但是这些异构系统之间却存在着一个共同之处，即它们使用标准的 HTTP 协议连接到 Internet 上，并传送着标准的 HTML 页面。正是这些标准的使用，Internet 才会如此迅速地发展，而 XML Web 服务正是建立在标准的协议和规范之上。任何系统，只要支持这些标准的协议和规范，就可以使用 XML Web 服务，因此 XML Web 服务天生具有跨平台的特性。

XML Web 服务是应用于 Web 环境中的分布式组件开发技术，它与传统的组件一样，也表示一个封装了一定功能的黑盒子，用户可以使用它而不用关心它是如何实现的。XML Web 服务提供了定义良好的接口，这些接口描述了它所提供的服务。用户可以通过这些接口来调用 XML Web 服务提供的功能。开发者可以通过把远程服务、本地服务和代码结合在一起起来创建应用程序，图 1-1 显示了使用 XML Web 服务应用程序的一般模型。

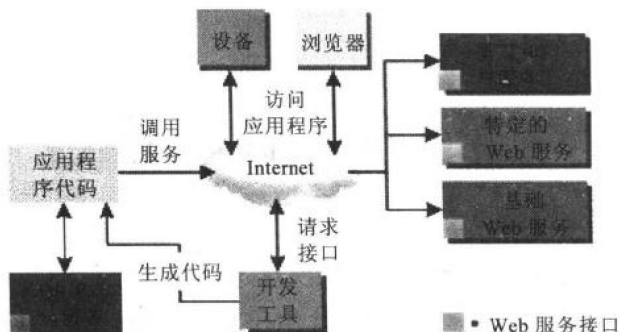


图 1-1 使用 XML Web 服务的应用程序模型

从图 1-1 中可以看出，用户可以在多种不同类型的应用程序(包括桌面应用程序、浏览器、其他设备以及其他的 XML Web 服务)中通过 Internet 标准协议来访问 XML Web 服务，这些 XML Web 服务可以位于不同的位置并且在不同的系统中实现，它们的功能通过接口来调用。用户还可以在应用程序中同时使用本地服务和位于 Internet 上的 XML Web 服务。

与当前传统的组件技术不同，XML Web 服务不使用特定于对象模型的协议(比如 DCOM、RIM 或 IIOP，这些协议要求在客户机和服务器上同时具有特定的基础结构，即需要在可控制的环境中使用)。在 Web 环境中，客户机和服务器千变万化，它们使用不同的操作系统和不同的技术，因此把组件的实现紧密地绑定在特定的组件技术上是不切实际的。XML Web 服务是使用 HTTP、SOAP 和 XML 等普遍存在的标准协议和数据格式来进行通信的，因此所有支持这些协议和数据格式的系统都能支持 XML Web 服务。

XML Web 服务以消息的形式来提供服务，它使用基于 XML 的消息来作为数据通信的基本方法。这使 XML Web 服务完全与语言、平台和对象模型无关。XML Web 服务可以使用任何语言和对象模型在任意的平台上实现，任意的应用程序都可以使用 XML Web 服务。只要描述 XML Web 服务功能的接口和消息序列以及协议保持不变，XML Web 服务和客户应用程序可以任意改变而不会相互影响。

提示:

在 XML Web 服务中大量地使用 XML 来表示命令和类型化数据，比如 SOAP、WSDL 等都是基于 XML 的，也可以说 XML Web 服务完全就是 XML 技术的一种实现。正因为 XML 在 XML Web 服务中起着至关重要的作用，所以才把这种技术称作“XML Web 服务”。

用户可以使用 XML Web 服务实现许多功能。比如用户可以使用 XML Web 服务来实现一个股票查询系统，它接收客户程序传送的股票代码，然后返回最新的股票报价；用户也可以使用 XML Web 服务实现一个操作数据库的中间层，它接收客户程序传送的

查询语句然后返回查询结果；甚至用户也可以使用 XML Web 服务来实现一个 Web 服务器，它接收客户程序传送的 URL 地址，然后生成相应的网页并返回给客户程序。

1.2 XML Web 服务的基础结构

因为 XML Web 服务要在异构的 Web 环境中使用，在这个环境中存在着多种不同的操作系统、对象模型和编程语言，所以 XML Web 服务必须具有以下特点。

- 松散耦合：客户程序仅使用自描述的、基于文本的消息与 XML Web 服务进行通信。
- 方便的通信：连接到 Internet 上的任意系统或设备都可以与 XML Web 服务进行通信。
- 通用的数据格式：XML Web 服务使用广泛支持的 XML 标准来描述其数据格式，所有支持这个标准的系统都可以识别 XML Web 服务的消息。

XML Web 服务的基础结构提供的功能有：定位 XML Web 服务的发现机制、定义服务用法的服务描述以及与 XML Web 服务进行通信所使用的标准消息格式。图 1-2 显示了 XML Web 服务的基础结构。

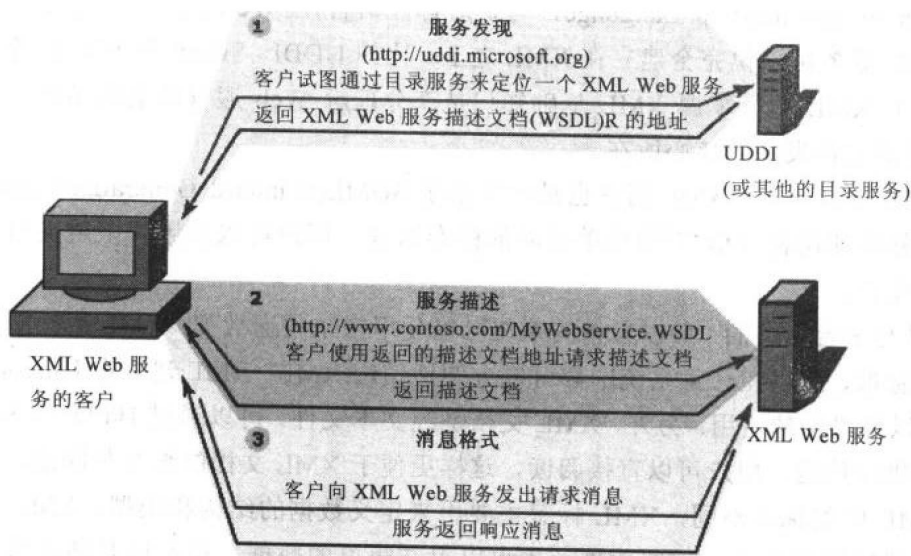


图 1-2 XML Web 服务的基础结构

从图 1-2 中可以看出，XML Web 服务的客户通过 UDDI 来查找符合需求的 XML Web 服务，这个过程将返回相应 XML Web 服务的描述文档(WSDL 文档)的地址(这个过程也可以返回一个其他服务文档的地址，比如 DISCO 文档或 WS-Inspection 文档，然后客户再通过这些文档来获取描述文档的地址)；获得 XML Web 服务的 WSDL 文档



地址之后, 客户程序就可以通过它来请求描述 XML Web 服务的 WSDL 文档, 它详细地描述了 XML Web 服务提供的接口以及进行通信所使用的消息格式; 最后, 根据 WSDL 文档, 客户程序就可以构建符合特定通信协议(通常为 SOAP 协议)的请求消息, 并可以识别 XML Web 服务返回的响应消息, 因此就可以向 XML Web 服务发出请求并接收它的返回值。

提示:

如果用户已经知道 XML Web 服务的地址, 则可以省略服务发现的过程, 而直接去请求 XML Web 服务的 WSDL 文档。另外, Visual Studio.NET 也为用户提供了强大的工具来帮助发现服务并可以根据 WSDL 文档自动生成在客户端使用的代理类。

下面简要介绍 XML Web 服务所使用的 XML、UDDI、WSDL 和 SOAP 等标准规范与协议。另外, 用户可以参照作者的《Visual C# .NET 中文版 Web 服务开发基础》一书来详细了解相关协议与规范的具体内容, 这本书也是清华大学出版社出版的。

1.3 XML 标准

XML(Extensible Markup Language, 可扩展标注语言)在 Web 服务中起着至关重要的作用, Web 服务可以说完全建立在 XML 之上, 比如 UDDI、SOAP 和 WSDL 等基本规范都是基于 XML 的。掌握 XML 基础知识是充分理解 Web 服务的前提条件, 而且 XML 也是今后软件发展的核心所在。

同 HTML 语言一样, XML 语言也是一个基于 SGML(Standard Generalized Markup Language, 标准通用标注语言)的简单灵活的标准语言, 用户可以使用它来定义自己的标签和数据结构。

XML 是与平台无关的, 所以异构的系统可以使用它来传递数据; XML 也是一个被广泛支持的标准, 所以在任意系统中都可以方便地使用 XML; XML 建立在 Unicode 基础之上, 所以可以全球通用; 另外, XML 文档是纯文本文件, 可以通过 HTTP 或 SMTP 等标准协议进行传送, 用户可以直接阅读, 这样更便于 XML 文档的编写与修改。

与 HTML 中的标签不同, XML 标签主要用来定义数据的结构和类型。XML 使用一组标签来描述数据元素, 这些数据元素可以封装简单的数据, 也可以封装非常复杂的数据。另外, HTML 中的标签是预先定义好的一些标准标签, 而 XML 中的标签则可以任意的定义(通过 XML 架构进行定义), 比如 SOAP 和 WSDL 规范就是使用自定义的标签。

下面给出一个简单的 XML 文档:

```
<?xml version="1.0" encoding="GB2312" ?>
```

```
<!-- 最后更新日期 2002-1-14 -->

<bookcase xmlns="http://www.sunyongqiang.com/bookcase.xsd">
  <图书 bookid="cp-001">
    <标题>C#基础</标题>
    <简介>由浅入深地介绍 C#语言，包括基础知识、
    Windows 应用程序的开发、访问数据库和
    Web 应用程序的开发。</简介>
    <作者>孙永强&陈宗斌</作者>
    <出版社>清华大学出版社</出版社>
    <类别>计算机</类别>
    <价格>40</价格>
  </图书>
  <图书 bookid="cp-002">
    <标题>使用 C#开发 Web 服务</标题>
    <简介>
    介绍如何使用 Visual C#.NET 来开发 Web 服务，并详细地介绍了
    Web 服务所使用的 XML、SOAP 和 WSDL 等标准规范与协议
    </简介>
    <作者>孙永强</作者>
    <出版社>清华大学出版社</出版社>
    <类别>计算机</类别>
    <价格>39</价格>
  </图书>
</bookcase>
```

从上面的示例文档中可以看出，与 HTML 语言一样，XML 中的标签也位于<和>中(XML 甚至可以使用中文标签)，注释也使用<!--和-->表示等。

与 HTML 不同，XML 对格式的要求非常严格，XML 文档首先必须要格式正确(well-formed)，主要包括以下几点：

- XML 文档的第 1 条语句必须是<?xml version="1.0" encoding="GB2312" ?>形式的 XML 声明语句(用来指出版本号和所使用的编码等信息)，并且在它之前不能包含任何内容，包括空格。
- XML 文档必须包含惟一的根元素，比如上面示例文档中的<bookcase>元素，XML 文档中的其他元素都必须位于这个根元素中。
- XML 文档中的所有标签必须具有对应的结束标签，例如<bookcase>和</bookcase>。
- 所有的属性值必须使用引号(双引号或单引号)。



- 用户不能在 XML 文档的文字部分使用 <、> 和 & 等字符，必须使用转义字符序列，比如 <、> 和 & 等。

如果 XML 文档的格式不正确，XML 解析器就会报告一个错误并停止文档的解析过程。另外，要使 XML 文档能够起到实际的作用，还要求它必须是有效的(valid)，即 XML 文档中的所有标签都必须是通过 XML 模式明确定义的标签。

1. 元素和属性

XML 文档由元素组成，元素构成了 XML 文档的框架。在 XML 文档中，元素使用标签表示，每一个元素都有开始标签和结束标签。在标签中，用户可以指定元素的名称和元素的属性。元素中可以包含其他的元素或文字内容。例如：

```
<myelement>
<data>1</data>
<data>2</data>
</myelement>
```

不包含其他元素或任何文字内容的元素称为空元素，但是空元素中可以包含属性。空元素也具有开始标签和结束标签，但是在开始标签和结束标签之间不能有任何内容。例如：

```
<空元素 attr1="属性 1" attr2="属性 2" ...></空元素>
```

空元素也可以采用一种缩写方式来表示，即直接把表示结束标签的斜线“/”放到开始标签中。例如：

```
<空元素 attr1="" ... />
```

属性允许用户使用名称-值对的方式向元素中添加相应的信息。属性只能出现在元素的开始标签中，并且属性的值必须使用引号括起来。例如：

```
<myelement att1="value1" att2="value2" ...>
.....
</myelement>
```

2. 命名空间

XML 命名空间主要用来限定元素名称的范围。通过使用 XML 命名空间，用户可以惟一地限定元素的名称，因为即使元素的名称相同，只要它们位于不同的命名空间中，就可以明确地区分它们。

在 XML 中，通常使用 URI(Universal Resource Identifier，统一资源标识符)作为命名空间的名字。因为 URI 通常由国际组织进行维护，可以确保它的惟一