

新网络体系结构—Web Services 研究综述^{*}

饶元 冯博琴

(西安交通大学计算机科学与工程系 西安 710049)

摘要 Web 服务作为一种新型的松耦合分布式计算范式而成为目前业界研究的热点。本文从分布式计算、Grid 计算和 XML 等技术发展融合的角度提出 Web 服务是分布式技术发展的第四个阶段,并将它与前三个阶段的技术特点进行了综合比较。提出了一种 Web Services 新架构模型:RSRPM 模型,在形式化定义的基础上,对实现步骤、方法和协议进行了描述。另外,本文对目前 Web 服务的应用情况与主要开发平台进行了比较分析,并指出了 Web 服务目前存在的主要技术挑战和发展趋势。

关键词 分布式计算, Web 服务, XML, SOAP, UDDI

New Generation of Network Architecture—Web Services Review

RAO Yuan FENG Bo-Qin

(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049)

Abstract Web Services is the method of building software that allows companies with different computing systems to interact and conduct transaction. As the fourth developin stage of distributed computing, the evolution process and the definition of Web Services from the views of the distribution computing, Grid computing and XML technique points are presented and compared with the others stage in different aspects. A new Web Services architecture model: RSRPM (Resources, Services, Roles, Protocols and Methods) model has been provided and formulated definiton. The main steps to build, to apply and to realize Web services application and core protocol stacks are analyzed in RSRPM schema. Furthermore, the technique challenges and trends of the Web service are pointed out.

Keywords Distribute computing, Web services, XML, SOAP, UDDI

Internet 环境下的异构应用系统间的交互问题一直制约着电子商务发展。为了实现跨平台、语言独立、松散耦合的动态商务,人们将注意力转向了新出现的一种分布式计算体系结构—Web Services^[1]。它通过将业务逻辑功能封装成 Internet 的软件组件——服务,并发布该服务供其它用户使用,从而使得 Web 不仅成为信息共享的平台,而且成为服务共享的平台。本文从 Web Services 的发展历程、体系架构、目前应用情况以及存在的技术挑战和发展趋势进行总体的论述。

1 发展历程

Web Services 技术是分布式计算^[2]、Grid 计算^[3]和 XML^[4]等技术发展和相互促进的结果。它吸收了各种技术的优点,成为目前解决 Web 环境下跨平台、跨语言、松散耦合的分布式系统最佳解决方案,并受到 Microsoft^[5]、IBM^[6]、SUN^[7]等业界巨头们的大力支持。

分布式计算概念自 20 世纪 80 年代被提出以来,大致经历了分布式数据、分布式组件、分布式体系架构和目前正得到广泛研究的分布式服务这四个主要的发展阶段,其中,前两个阶段分别是以 OLE/XDBC (ODBC/JDBC)与 COM/DCOM 和 EJB/RMI 为代表的两层体系结构中的分布式数据共享处理以及多层体系结构中的远程自动化分布式调用的技术方案。它们为分布式技术以及随后的分布式体系架构的发展奠定了技术与实现基础。20 世纪 90 年代出现的以 DNA 和 CORBA 为代表的分布式体系架构使分布式技术逐渐走向成

熟。它们将系统分为表示逻辑、业务逻辑和数据逻辑,通过多层次设计,提高了分布式系统的可伸缩性。但这种紧耦合的分布式环境,要求通过特定的通信协议在对等体系结构间才能进行通信,且对组件定位后才能访问。而无法满足 Internet 环境下的无状态、松耦合、异构系统间的通信要求。

与此同时,Grid 技术无论是从概念还是应用技术都得到了巨大的发展^[3,8]。Grid 是一种集成的计算与资源环境,它具有资源的分布性、结构的自相似性、节点的动态可伸缩性和管理的多重性和自治性等特点。针对这一些特点,Foster 等人提出了以协议为中心的五层沙漏模型^[9]和以服务为中心的 OGSA 模型^[10]。此外,以 Globus 组件^[11]为核心的网格底层支撑技术为解决跨操作系统的通信和跨文件系统的访问等异构分布式问题提供了有效手段。

XML 技术为 Internet 的数据内容描述与管理提供了一种与平台无关,且可伸缩的元语言描述机制^[4]。它对信息采用树状结构和嵌套规则的描述,并支持 Unicode 而实现语言的独立性。因此,使用 XML 有利于 Web 上的数据分发和集成,并能以可移植方式共享信息。

分布式服务(Web Services)技术正是融合了 Grid 计算中“无处不在的集成的计算与资源环境”的思想^[12]和 OGSA 体系结构的一些应用技术,并在 HTML、SMTP 等 Internet 标准协议的基础上,使用基于 XML 的文本消息传送模型进行通信,从而真正实现分布式 Web 系统间跨平台、跨语言的无缝融合,从而解决了传统分布式体系架构无法解决的在 Internet 环境下的松耦合分布式异构问题。不同阶段的分布

饶元 博士生,主要研究领域为新型网络与分布式计算;冯博琴 博导,研究领域为智能网络。

式技术特点的综合比较详见表 1。

表 1 不同分布式技术发展阶段的技术特点综合比较

	分布式数据	分布式组件	分布式体系架构	分布式服务
代表技术	OLE /ODBC	COM/DCOM & EJB/RMI	DNA/CORBA	Web Services
面向的问题域	数据共享与处理	远程自动化分布式调用	Intranet 环境下紧耦合、跨平台、异构应用的互操作	Internet 环境下的松耦合异构应用的互操作
体系结构	单机或 C/S 两层结构	C/AS/DS 三层应用结构	三层或多层式应用结构	多层式结构
技术特点	面向数据: 采用 IOS/IEC 的 SQL 级调用接口规范以及 DDE 服务来实现数据的共享	面向对象: 通过远程自动化调用 (RPC) 分布式组件, 为无状态系统设计提供了可伸缩性	面向组件: 通过消息传递的方式, 将对象技术与分布式技术相结合, 实现了跨平台系统间的互操作	面向服务: 通过基于 XML 的互操作协议栈 (SOAP, UDDI, WSDL) 完成基于 Web 的松耦合异构系统间的互操作
应用特点	简单易用, 维护时易出现版本控制的问题	需要对远程组件进行定位后, 才能实现调用	紧耦合, 同构系统间才能完成调用, 且系统的伸缩较强, 但维护与部署较为繁琐	系统的伸缩性强, 可靠性高, 且易于维护和部署
扩展性	差	较好	好	很好
实际贡献	提供了大量的桌面应用, 并实现了数据的共享与处理	提供了大量的面向对象的分布式应用系统, 为分布式技术的发展奠定了重要的应用基础	建立起了分布式集成体系架构, 以及相应的技术标准框架, 使分布式技术趋向成熟	目前正在快速地发展, 它提供了一个系统架构以及一系列的技术标准与规范

2 Web Services 体系架构

2.1 Web Services 定义

Web Services 作为一种新的分布式计算范式, 不同的研究人员曾给出不同的定义, 其中, Tsalgaidou 等^[12]认为: Web Services 是一种自包含、模块化的应用程序, 它通过网络化访问为企业与个人提供了一系列的功能。Enrique^[14]认为: 一个 Web Services 就是一个可以被 URI 识别的软件应用, 它的接口和绑定可以被 XML 描述和发现, 并可通过基于 Internet 的协议直接支持与其它基于 XML 消息的软件应用交互。

我们认为 Web Services 是由一个五元组组成的一个分布式有机智能软件体。即: $\xi = (\Omega, \Sigma, \Pi, \Psi, \Delta)$ 。其中, ξ 表示一个由 Web Services 组成的系统架构; Ω 是 ξ 中业务逻辑单元 (资源对象) 组成的非空集合, 即 $\Omega = \{r_i | 1 \leq i \leq n, r_i \in \phi, n \in R\}$; Σ 是 ξ 中所提供的 Web 服务 (服务对象) 组成的非空集合, 即 $\Sigma = \{s_j | 1 \leq j \leq m, s_j \in \phi, m \in R\}$, 它是对资源对象的一种抽象和封装; Π 是 ξ 中的角色组成的非空集合, 即 $\Pi = \{p_f | 1 \leq f \leq k, p_f \in \phi, k \in R\}$ 。在 ξ 中, 包含请求方 (Requester), 中介方 (Broker) 和提供方 (Provider) 三类角色; Ψ 是 ξ 中的协议栈组成的非空有限集合, 即 $\Psi = \{z_k | 1 \leq k \leq c, z_k \in \phi, c \in R\}$, 其中 SOAP、WSDL、UDDI 是 Ψ 中的核心协议; Δ 是 ξ 中的角色 p_i 施加在服务对象 s_j 上的非空操作指令集, 它由 Σ 和 Π 中的对象元素组成的有序对偶来表示, 即 $\Delta = \{\langle p_i, s_j \rangle | p_i \in \Pi, s_j \in \Sigma\}$ 。

无论具体的定义如何, 在本质上, Web Services 是采用普遍使用的 Internet 协议与其它分布式的软件组件 (封装资源的服务接口) 进行交互的自包含、自适应和自描述的软件组件。

2.2 Web Services 体系结构

根据上述定义, 我们提出了一种由资源对象 (RO)、服务对象 (SO)、角色对象 (RO)、协议栈 (P) 和指令方法 (M) 五个要素构成的 RSRPM 系统服务结构模型, 如图 1 所示。其中, 虚线代表了实体间存在着耦合。

在 RSRPM 模型中, 服务对象是对资源对象的抽象和封装, 并通过一个 Web 服务的形式向外部提供由 WSDL 描述的通用组件接口。提供者将创建的服务注册和发布到 UDDI 中心后, 服务请求者就可以通过 UDDI 中心进行查找和引用,

最后实现与提供者的绑定和调用。整个服务的创建与调用过程参见图 1 中的步骤流程, 另外, 由于消息的传输与响应都是利用基于 XML 的协议栈完成, 从而实现了真正意义上的跨平台和语言无关性。

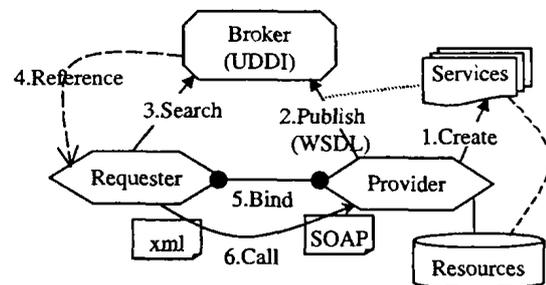


图 1 RSRPM Web Services Architecture

2.3 核心协议

Web Services 协议栈是以 XML 为基础, SOAP、WSDL 与 UDDI 为核心的标准集。它主要解决了在实现中所面临的服务的描述、发布与发现机制和消息传输与安全以及工作流等各类问题, 并从技术上保证了协议的开放性、通用性和兼容性, 其主要协议见表 2。其中, 可靠与安全性、协作与可管理性与 QoS 等功能是协作型 Web Services 与基本型 Web Services 主要区别^[15], 但核心协议内容基本相同。

表 2 Web services 基本协议栈

交互栈	服务应用 (工作流)	WSFL...	安全 (WS-Security)	管理 (Management)	质量 (WS-QoS)
	服务发布、发现	UDDI			
	服务描述	WSDL			
	消息传递	SOAP			
	数据描述	XML			
	网络	HTTP, FTP...			

SOAP 协议 SOAP 是 Web Services 中进行数据传递与通信的桥梁^[12, 16]。它是一组基于 XML 的无状态、单向、轻量级的消息传递协议, 用于无中心、分布式远程过程的调用。SOAP 主要由三部分组成: 一个基于 XML 的消息内容的 Envelope 式封装机制; 一组用于各种数据类型的编码规则和一个提供远程过程调用 (RPC) 和响应的机制。同时, SOAP 可以有效地嵌入 HTTP 协议中, 通过结合单向交换和底层协议

来创建请求/响应或请求/多重响应等交互模型进行发送和获取消息。其基本的消息传送机制如图 2。SOAP 发送者将远程过程所需要的信息编组(marshaling)为 SOAP 消息,并封装成 SOAP 信封发送到 SOAP 接收站点;在 SOAP 接收端,SOAP 信封通过处理器提取并取消编组(un-marshaling),将消息转化为 SOAP 应用可理解的方法调用,从而完成整个消息的传送。

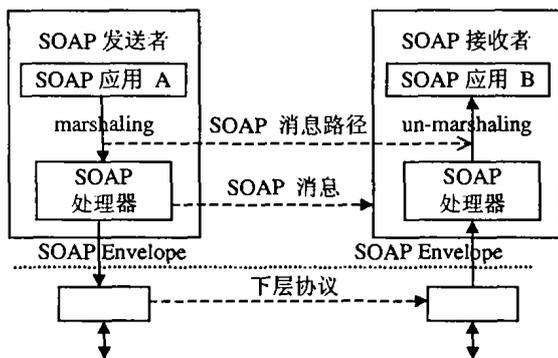


图 2 SOAP 的消息传送机制

WSDL 协议 WSDL 完全基于 XML,它把 Web Services 抽象地描述为一组包含在面向文档或面向过程信息的消息上执行操作的端点的集合^[17],而具体的端点必须绑定到具体的网络协议和消息格式上来定义。这种抽象描述为端口重用提供了一种有效的实现机制。多个相关的具体端点结合在一起就构成了服务。WSDL 文档通过 6 个元素来具体定义一个服务及其属性,即:

- definitions 元素:用来定义 Web 服务的名称以及声明名称空间;
- types 元素:一个使用特定类型规则定义的数据类型容器;
- messages 元素:对具有一定规则定义的交互数据的抽象描述。
- PortType 元素:为端口提供不同抽象操作的集合;
- binding 元素:指定一个具体端口类型的协议和数据格式。
- service 元素:一组端口地址与特定绑定的集合。

例如,下面这段代码是我们在开发 OPEN-CLASS 网络教学管理平台中,调用班级信息查询服务的 WSDL 描述文档,及调用操作“GetClassInfo”。

```
<? xml version="1.0" encoding="utf-8" ?>
<DataSet xmlns="http://tempuri.org/">
<xs:schema id="GetClassInfo" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
<xs:element name="GetClassInfo"
msdata:IsDataSet="true" msdata:Locale="zh-CN">
<xs:complexType>
<xs:choice maxOccurs="unbounded">
<xs:element name="CTC">
<xs:complexType>
<xs:sequence>
<xs:element name="TeachID" type="xs:string" minOccurs="0"/>
<xs:element name="ClassID" type="xs:string" minOccurs="0"/>
<xs:element name="CourseID" type="xs:string" minOccurs="0"/>
<xs:element name="Teachname" type="xs:string" minOccurs="0"/>
<xs:element name="Semester" type="xs:string" minOccurs="0"/>
<xs:element name="Place" type="xs:string" minOccurs="0"/>
<xs:element name="Schedue" type="xs:string" minOccurs="0"/>

```

```
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1"/>
</DataSet>
```

图 3 班级信息查询服务的 WSDL 描述文档

UDDI 是一个基于 SOAP 协议的,为 Web Services 提供信息注册中心的实现标准,同时也包含一组提供 Web 服务注册、发现和调用的访问协议^[18]。它通过 XML 将用户提供的 Web 服务,注册在 UDDI 中心的公共注册表内供其它用户查询和使用,一般它可通过三个方面的信息来结构化地描述一个服务:

- 一般信息(白页):Web 服务的 URL 和提供者的名称、地址等基本信息。
- 分类信息(黄页):基于标准分类法(如相关产业、产品或服务类型以及地域等)的分类系统。
- 技术信息(绿页):是发现潜在 Web 服务的关键,提供发现的接口与发现的机制。

UDDI 本身就是一个 Web 服务,它通过一组基于 SOAP 的 UDDI API 进行访问。其中 Inquiry APIs(查询 APIs)用来查询定位商业实体、服务、绑定等信息。Publishing APIs(发布 APIs)被用来在注册中心发布或者取消服务。同时,UDDI 提供了基于 XML 的轻量级的数据描述和存储方式,并通过以下 5 种数据结构来具体实现,这些数据结构间的关系见图 4。

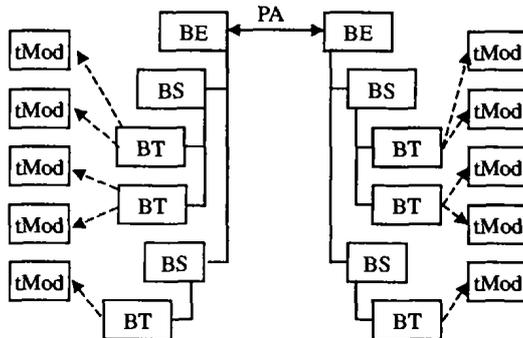


图 4 UDDI 基本数据结构间关系图

- BusinessEntity (BE):包含公司及其提供的服务的一般性描述性信息,如名称、联系方式等。
- BusinessService (BS):业务实体提供的更为详细的服务或业务处理过程信息。
- BindTemplate (BT):包含实际调用服务所需的具体技术和接口信息。
- TModels (tMod):描述服务的具体技术规范 and 标准,代表了技术指纹、接口和元数据的抽象类型。
- Publisher Assertion (PA):UDDI 注册中心中 2 个业务实体之间的关联信息。

3 目前应用的情况与工具比较

由于 Web 服务与 Web 应用的对象与范围存在许多不同,这为新应用模式的开发提供了一个广阔的空间,同时也吸引了越来越多的企业和研究机构的注意,其中,UDDI Community 从最初的 36 个成员,已迅速发展到现在超过 300

家公司和机构加盟^[19]。整个 UDDI 的开发和支持团体已经覆盖了像波音和福特这样的工业巨头和大批追随者,以及 Microsoft、IBM 和 Sun 等技术提供者,这些技术提供者通过不同的开发平台对 Web 服务的标准协议进行封装与实现。其中平台的主要性能指示比较如表 3 所示。

表 3 Web Services 开发平台比较

	J2EE	.Net
支持语言	Java	C#, Vb. Net, C++, Perl, COBOL
运行时环境	JVM	CLR
数据访问	JDBC, Java Connectors	ADO. NET
通信协议	RMI/IIOP	XML/SOAP
异步消息处理	Java Message Service (JMS)	MSMQ
表示层技术	Servlets, JSP	ASP. NET
安全访问	JAAS	COM+ Security
事务处理	Java Transaction Server (JTS)	MS-DTC
开发工具	Visual Café, Borland JBuilder, IBM Visual-Age 等	Visual Studio. NET
应用特点	规范集合, 扩展性强, 采用开放标准, 多平台支持, 配置复杂	产品集合, 界面友好, 易用, 多语言支持, 跨平台性能差

4 目前存在的技术挑战

由于 Web Services 的松耦合分布式的设计理念,使其在企业应用集成(EAI)、B2B 的电子商务集成、代码和数据重用,以及在通过 Web 进行跨越防火墙的客户端和服务器通信等应用领域中得到迅速推广。但在某些情况下,Web 服务也会降低应用程序的性能。例如,在单机或 LAN 中同构的应用程序使用 Web 服务进行通信的效率较低,而采用 COM/DCOM 或 CORBA 等技术则会保证系统具有较强的性能^[20]。

另外,Web Services 整个服务体系还不完善,在应用中存在的技术挑战主要反映在以下几个方面:

- 性能评价:如何对 Web 服务的质量和性能进行评价和测试以及如何选取测量基准并进行相应测量工具的开发,这将会直接影响到整个系统架构的应用与推广。另外,在服务被挂起或新建时,动态地提高系统的可靠性、容错性以及可伸缩性将是一个重要的研究内容。

- 完善系统架构:如何根据用户的需要主动地去发布服务以及如何实时、按需地发现和查询一个用户提供的服务,目前还存在一些技术问题,但 Sun 公司为此提出的 Smart SunOne 战略为这些问题的解决提供了一个可行的未来解决方案^[21];此外,Web 服务虽然在目标上致力于解决 Internet 分布式系统的跨平台、跨语言的互操作问题,但目前的 J2EE 与 .NET 两大技术阵营在互联、互操作上还存在许多问题需要进一步解决,尤其是要解决不同 Web 服务平台间的互操作性问题。

- 安全性:Web 服务一旦发布,如何能够有效地保证不同粒度级别的数据机密性、完整性和可用性是一个重要的研究问题,Http 和 SOAP 为 Web 服务提供了一定级别的安全性,WS-Security^[25]提供了一个框架级别的安全标准,在其实实施过程中还需要进一步提供有效的论证机制和加密授权方式,其中,Microsoft Passport^[22]和 Liberty Alliance^[23]在统一安全认证服务的提供与协议的制订方面做出了大量的工作,但在企业级别的集成应用还远远不够。

- 可管理性:如何对松耦合高度分布式的系统进行有效地管理,对长事务所占用的资源进行有效地并发调度,以及对

Web 服务的收费与经营模式制订等为人们提供了较为广阔的研究空间。其中 IBM 的 WSTx^[24]项目与 HP 的 BTP 协议对长事务的处理与协同进行了一些有益的探索和研究,但与实际应用仍存在一定的距离。

总结 Web 服务是一种新出现的、重要的分布式计算范式,它利用 Internet 协议(如 HTTP 和 SMTP)实现远程调用的应用程序组件,是可编程网络或 URL 的一种新模型,它强调基于 XML 标准来解决异构分布式计算以及代码与数据重用等问题。目前,Web 服务正向服务的智能化、计算的普适化、管理的简化与透明化和系统的高安全性和可靠性等方向迅速发展。除了 Sun 公司的 Smart SunOne 外,IBM 和微软也开始将注意力转向智能化的主动服务。Web 服务对无线设备的支持,使得“无处不在的计算”正成为现实。解决目前 Web 服务架构中存在的诸如安全性、可靠性、可伸缩性等问题也为下一代 Web 服务的发展提供了良好的技术支持与储备。

本文比较分析了分布式计算发展的四个主要阶段所采用的技术。提出了一种 Web 服务架构模型:RSRPM 模型,并在此模型的基础上分析了实现部署 Web 服务所需要的核心步骤、方法以及核心协议栈。另外,本文对目前 Web 服务主要开发工具进行了比较分析,并指出了目前面临着性能评价、完善系统架构、安全和管理等四类技术挑战和发展趋势。

参考文献

- 1 Chai F. The Web Services Simple Page. <http://www.wsc.org/simplepage/>
- 2 Ogbuji U. The Past, Present and Future of Web Services. <http://www.webservices.org/index.php/article/articleview/663/1/61/>
- 3 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 2001, 15(3)
- 4 <http://www.xml.org/xml/resources-cover.shtml>
- 5 <http://msdn.microsoft.com/webservices/>
- 6 Tidwell D. Web Services: The Web's next revolution. <http://www-900.ibm.com/developerWorks/cn/education/webservices/wsbasics/tutorial-eng/index.shtml>
- 7 <http://gceclub.sun.com.cn/>
- 8 Allen G. Cactus Grid Computing. <http://www.cactuscode.org/Presentations/Europar-August01.ppt>
- 9 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 2001, 15(3)
- 10 Foster I, Kesselman C, Nick J M, Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. <http://www.globus.org/research/papers/ogsa.pdf>
- 11 <http://www.globus.org>
- 12 Tsalgatidou A, Pilioura T. An Overview of Standards and Related Technology in Web Services. *Distributed and Parallel Databases*, 2002, 12(2-3): 135
- 13 都志辉,陈渝,刘鹏编著,李三立审. 网格计算. 清华大学出版社, p32
- 14 Castro-Leon E. A perspective on Web Services. <http://www.webservices.org/index.php/article/articleview/113/1/61/>
- 15 Malks D, Sum M. Developing Web Services with ebXML and SOAP: An Overview. <http://www.webservices.org/index.php/article/articleview/1015/1/24/>
- 16 Testimonials for W3C's SOAP 1.2 Recommendation. <http://www.w3.org/2003/06/soap12-testimonial>
- 17 Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language. <http://dev.w3.org/cvsweb/~checkout~/2002/ws/desc/wsdl12.html>
- 18 Newcomer E. Finding Web services with UDDI. <http://www.webservices.org/index.php/article/articleview/596/1/24/>
- 19 <http://www.uddi-china.org/>
- 20 <http://tips.9hao.com/liulang/article/readnews.asp-id=4615.htm>
- 21 <http://gceclub.sun.com.cn/NASApp/sme/controller/bottom-learning?cat-id=0105#>
- 22 <http://www.passport.net/consumer/consumerqa.asp?lc=2052>
- 23 <http://www.projectliberty.org/>
- 24 <http://www.iturils.com/enterpriseinfo/ei-64.asp>
- 25 Specification: Web Services Security (WS-Security). <http://www-106.ibm.com/developerworks/library/ws-secure/>