

# 基于 Web Services 的电子政务系统集成研究

## Integration of E - Government System Based On Web Services

何诗广 王卫平 (中国科学技术大学管理学院 230052)

**摘要:**电子政务中对关键业务的处理提出了较高的一致性、可扩展性和健壮性要求,传统的做法难于满足这种变化。Web Services 是一种面向服务的体系结构,具有简洁性、开放性、灵活性等特点,已在商业化项目中有广泛的应用。该文基于 Web Services 的特点,提出了一个基于 Web Services 的电子政务系统集成方案,并分析该方案各个环节的关键技术实现。

**关键词:**Web Services 电子政务 系统集成

### 1 引言

由于市场经济的发展、公共事务处理效率的不断提高和技术发展的原因,政府在信息化建设过程中,各政府职能部门常常根据自身的情况与需要,建立了自己的信息和业务系统,这些业务系统往往是采用不同的技术标准、不同的软硬件平台、并由不同的软件提供商开发的,这些跨平台异构系统形成了一个信息孤岛。随着电子政务的进一步发展,如何将分散的信息孤岛进行有机的集成是非常迫切的问题。这就导致了电子政务系统集成的出现。从事务角度看,电子政务系统集成指的是政府将各业务系统应用集成到一个统一的信息系统中,实现数据共享并能够对业务流程提供支持;从技术角度看,电子政务系统集成指的是将不同的应用程序和数据集成在一起的过程,从而在不对已有的应用程序做出过多修改的情况下,实现数据共享和业务流程的集成。

Web Services 提供了在不同体系结构下构建的系统之间相互提供应用接口服务、数据的一种方案。Web Services 的主要目标就是在现有各种异种平台的基础上构筑一个通用的平台无关、语言无关的技术层,各种不同平台之上的应用依靠这个技术层来实施彼此的连接和集成。它采用通用的 SOAP、UDDI、WSDL 以及 HTTP 等技术,把原本互不相干的站点服务形成一整套分布的、自动化和智能化的网络应用,大大减轻了程序员的开发工作量,充分地利用了已有的网络资源和开发资源。

Web Services 是一种面向服务的体系结构,具有简洁性、开放性、灵活性等特点,已被很多公司采用并在商业化项目中有广泛的应用。本文基于 Web Services 的特点,提出了一个基于 Web Services 的电子政务系统集成方案。

### 2 Web Services 及其集成特征

Web Services 是近几年提出的一种新的面向服务的体系结构,其中定义了一组标准协议,用于接口定义、方法调用、基于 Internet 的构件注册以及各种应用的实现。Web Services 具有自包含、自描述以及模块化的特点,可以通过 Web 发布、查找和调用<sup>[6]</sup>。

#### 2.1 Web Services 的体系结构

Web Services 采用 SOA (Service - Oriented Architecture) 模型作为它的体系结构<sup>[1]</sup>,如图 1 所示。

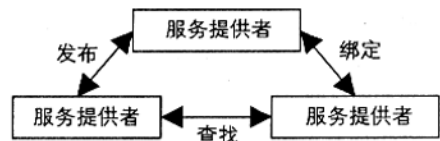


图 1 Web Services 体系结构

为了实现这个模型,Web Services 使用了一系列协议,这些协议组成了如图 2 所示的协议栈<sup>[1]</sup>。首先,服务提供者所能提供的服务接口用 WSDL 和 WSFL 描述出来,然后使用 UDDI 在服务中介者处将这些服务接

口进行注册;服务中介者则使用 UDDI 注册这些服务接口,并接受服务请求者的查询;服务请求者通过使用 UDDI 在服务中介者处查询,得到所需的服务接口的描述文件,然后绑定到这些服务,最终完成调用。参与者之间的三种操作都采用 SOAP (Simple Object Access Protocol) 完成。

.....	.....	.....	Management	Quality of Service	Security
Routing, Reliability and Transaction	.....	路由、可靠性及事务			
Workflow	WSFL	Web服务的流程语言			
Service Discovery, Integration	UDDI	Web服务的发现、集成			
Service Description	WSDL	Web服务的描述			
Messaging	SOAP	Web服务的访问			
Data	XML	Web服务的数据格式			
Transport	HTTP, FTP, SMTP	Web服务的协议簇 (HTTP、SMTP……)			
Internet	IPv4, IPv6				

图 2 Web Services 技术的层次结构

## 2.2 Web Services 的相关协议、技术

(1) SOAP。Web Services 的技术主要建立在 XML 的规范之上<sup>[2]</sup>,这保证了这一体系结构的平台无关性、语言无关性和人机交互性能。SOAP 用 XML 来格式化消息,用 HTTP 来承载消息。这主要是由于 HTTP 协议是 Internet 上应用最为广泛的通信协议,可以穿越多数防火墙。SOAP 包括 3 部分:定义了描述消息和如何处理消息的框架的封包 (SOAP 封包)、表达应用程序定义的数据类型实例的编码规则 (SOAP 编码规则) 以及描述远程过程调用和响应的约定 (SOAP RPC 表示)<sup>[3]</sup>。

(2) WSDL。WSDL 定义了一种基于 XML 规范的用于描述 Web 服务的语言<sup>[4]</sup>,它将 Web 服务描述为一组对消息进行操作的网络端点。一个 WSDL 服务描述包含对一组操作和消息的一个抽象定义、绑定到这些操作和消息的一个具体协议和这个绑定的一个网络端点规范。在 WSDL 中,作为一个网络端点的集合,Web 服务的端点及消息的抽象定义与它们具体的网络实现和数据格式绑定是分离的,这样就可以重用这些抽象定义<sup>[5]</sup>。

(3) UDDI。UDDI 提供一种发布和查找服务描述的方法。UDDI 是 Web Services 的消息注册规范,以便被需要该服务的用户发现和使用<sup>[6]</sup>。UDDI 规范描述了 Web Services 的概念,同时也定义了一种编程接口。通过 UDDI 提供的标准接口,系统可以发布自己的 Web Services 供其它系统查询、调用;也可以查询特定服务的描述消息,并动态绑定到该服务上。通过 UDDI, Web Services 可以实现消息的一次注册,随处访问。

(4) WSFL。由 IBM 提出,使用 WSDL 和 WSEL 来描述服务接口和它们的协议绑定。WSEL 是用来描述非操作性的服务特征 (如服务质量等) 的一种语言<sup>[9]</sup>。

## 2.3 基于 Web Services 系统集成的特征

与传统的分布式应用集成模型相比,Web Services 与平台和语言无关。面向 Web Services 的应用集成运用松散耦合的关系来组织应用之间的互联,实现应用之间信息的实时共享。Web Services 体系有如下优势:

(1) 简单性。由于业界的支持,面向 Web Services 的集成容易实现。作为实现 Web Services 的两大主流平台 J2EE 和 .NET 框架都提供了开发和部署 Web Services 的工具、实例和丰富的参考资料。这些支持极大地简化了 Web Services 的开发。目前有许多应用 Web Services 开发的工具,比如微软的 SOAP Toolkit、IBM 的 WSDK 等等。相对于使用 DCOM、CORBA 等技术,构建应用系统的 Web Services 更方便,开发周期要短得多。同时,Web Services 利用广泛使用的 HTTP、SOAP 等协议和 XML 技术,节约了为支持新标准的开销。

(2) 开放性。Web Services 是基于诸如 UDDI、SOAP、HTTP 等开放标准的。基于现存的开放标准可以消除政府潜在的为了支持新出现的 Web 技术的投入的需要。并且任何支持这些技术的设备都可以拥有和访问 Web Services。

(3) 灵活性。面向 Web Services 的应用集成克服了传统集成的紧密耦合性,它建立在发布服务的应用程序和使用服务的应用程序之间的松散耦合。当有新业务 Web 服务出现时,它只需注册到 UDDI 服务注册中心即可。Web Services 引擎会发现它并根据需要调用它。当有新的请求产生时,只需在集成规则库中添加新规则即可,因此,对 Web Services 的增加、修改、更新等操作易于进行,管理方便。

### 3 基于 Web Services 的电子政务系统集成方案

根据上面的讨论, Web Services 具有良好的集成性质, 是连接各类应用的桥梁, 采用的是松散的耦合方式, 即任何应用都可以通过对应的接口连接到系统中来, 方式灵活, 简单快速, 是高伸缩性、健壮性集成平台的良好选择。本文基于 Web Services 的特点, 提出了一种基于 Web Services 的电子政务系统集成方案。该集成方案如图 3 所示。

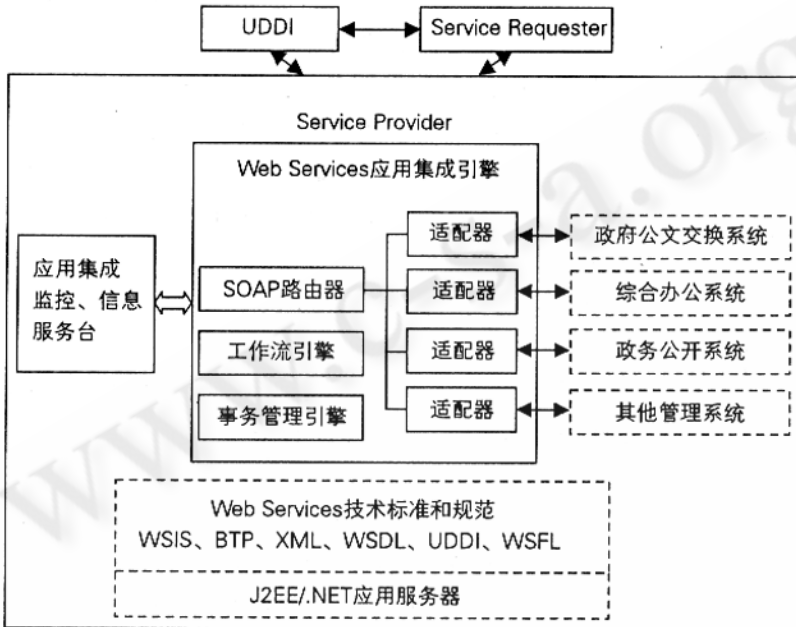


图 3 基于 Web Services 的电子政务系统集成方案

在该集成方案中, 政务系统既可以是已有的政务系统, 也可以是新开发的 Web Services 系统。如果是已有的政务系统, 需要先将此系统封装成 Web Services 组件, 其方法是: 首先生成描述该系统功能和调用方法的 WSDL 文件; 然后生成服务器端基于 SOAP 的服务框架, 并在此基础上开发适用于已有系统的适配器; 最后将服务描述文件通过 UDDI API 发布到 UDDI 注册服务器中。下面重点介绍该方案的几个关键环节。

#### 3.1 SOAP 路由器

SOAP 路由器是实现客户调用 Web Services 的关键部件, 用来实现 SOAP 消息的传递。客户调用 Web Services 的过程如下:

- (1) 客户用 WSDL 描述需要访问的服务, 用 SOAP 消息向注册中心发出查询请求;
- (2) 注册中心将该服务的 WSDL 描述返回客户;

(3) 客户用得到的 WSDL 描述生成 SOAP 请求消息, 绑定服务提供者;

(4) SOAP 请求被作为一条 HTTP POST 请求发出, 交由 SOAP 路由器处理;

(5) SOAP 路由器分析 HTTP 头找出某个 Web Services 适配器的位置, 将该请求传送到所请求的适配器;

(6) 适配器将消息传递到相关政务系统, 政务系统处理请求并将结果返回给适配器;

(7) 适配器将得到的结果打包成 SOAP 消息, 返回给 SOAP 路由器, SOAP 消息再返回到客户。

#### 3.2 Web Services 适配器

Web Services 适配器是集成引擎的核心, 它是政务系统与 SOAP 服务器进行信息交互的纽带, 由下面四部分组成:

(1) 接口, 针对不同的政务系统, 适配器提供不同的接口。这是政务系统可以调用 Web Services 集成平台的唯一途径;

(2) 连接控制逻辑, 用于建立安全的与后端服务器的通信连接;

(3) 数据转换器, 用于验证数据有效性、生成 WSDL, 实现在 SOAP 数据格式与政务系统数据格式之间的转换;

(4) 消息路由器, 实现在 SOAP 路由器与适配器之间的消息传递, 将 SOAP 消息过滤后路由到正确的目的地。

#### 3.3 应用集成、监控、信息服务台

应用集成服务台用于实时应用集成服务的配置和管理及政务系统解决方案的动态配置, 包括 workflow 管理、应用资源管理服务、统一界面服务等; 监控服务台用于对服务的运行进行管理和监控, 包括服务对象管理器、动态监控及安全管理等; 信息服务台用于为应用服务中心和监控中心提供数据和模型服务, 实现信息的安全生命周期管理和维护, 包括共享信息管理、共享模型管理、数据操作管理等。

#### 3.4 工作流引擎

工作流引擎是为了政府结构化或非结构化业务流程/活动提供执行环境的软件服务, 提供按照流程定义来执行流程的功能。WSFL 将业务过程中的每一个步骤都定义为一个活动, 将业务过程的一次执行看成是

各活动按照指定顺序的激发。

### 3.5 事务管理引擎

基于二阶段提交协议<sup>[7]</sup>,确定事务协调和参与者,由协调者控制整个事务的提交和失败后的事务回滚。关于 Web Services 事务处理,目前主要有 OASIS 提出的 BTP,以及 IBM、微软和 BEA 联合提出的 WS - Transaction 等标准。

## 4 基于 Web Services 的电子政务系统集成示例

如上所述,基于 Web Services 的电子政务系统集成方案是基于 Web Services 技术,集成引擎通过各类接口将应用封装成 Web Services 组件后发布到 UDDI 注册中心,并通过接口调用相应的应用。下面以用户的单点登录作为示例,来具体介绍 Web Services 技术在电子政务系统中的具体应用。

要实现多系统的单点登录,必须保证这些系统中的用户数据(包括用户名、密码等信息)保持一致。此时,应以某一拥有用户数据比较完整的系统(假设为 A 系统)中的数据为中心,通过 Web Services 服务,保证其他系统(假设为 B 系统)的用户和 A 系统的用户同步。

首先,确定在两个系统之间传递的用户数据都包含哪些信息,并以 XML 格式表现,如:

```
<UserInfo >
<UserID > admin </UserID >
<Password > xxxxxxxxxx </Password >
<UserName > 系统管理员 </UserName >
<CmpylID > 20 </CmpylID >
<Department > xxxxx </Department >
<Email > xxxx@xxx.gov.cn </Email >
<OfficePhone > xxxx - xxxxxxx </OfficePhone >
<mobile > xxxxxxxxxxxx </mobile >
</UserInfo >
```

然后,编写 Web Services 服务组件,该服务组件主要提供以下三种方法:

- (1) addUser(object userXml);
- (2) updateUser(object userXml);
- (3) deleteUser(object userXml)。

这三种方法都是根据接收到的 XML 信息,来更新 B 系统中的用户数据。然后将 Web Services 服务组件在 B 系统的应用服务器上部署,并将生成的 WSDL 文件提供给调用服务的系统。

最后,在 A 系统中调用用户同步服务组件。在 A 系统中,我们可以根据 WSDL 文件获得服务组件的访问地址以及提供的方法,然后在合适的地方访问服务,调用用户同步方法,如对用户的增删改操作以及禁用、启用等操作。

## 5 结束语

本文基于 Web Services 的特点,提出了一种基于 Web Services 的电子政务系统集成方案,并分析该方案各个环节的关键技术实现问题。该集成方案已经在实际的项目中实施,表现出了良好的可扩展性与健壮性。Web Services 是一种面向服务的体系结构,具有简洁性、开放性、灵活性等特点,但 Web Services 在执行效率、安全性等方面仍存在的问题。这些问题仍需要做进一步的工作。近年来,IBM、微软等 Web 服务体系架构的领导者陆续提出了一系列技术规范,这些技术规范将会推动 Web Services 的进一步发展完善。

### 参考文献

- 1 Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., & Ferris, C., et al. (2004). Web Services activity. W3C Working Group Note. Available from <http://www.w3.org/TR/ws-arch/>.
- 2 Vogels W. Web Services Are Not Distributed Object [J]. IEEE Internet Computing, 2003, 7(6): 59 - 66.
- 3 SOAP Version 1.2, W3C Recommendation 24 [S], 2003.
- 4 Web Services Description Language (WSDL) Version 2.0 [S]. W3C Working Draft, 2004.
- 5 Tapang CC. Web Services Description Language (WSDL) Explained, Infotects [Z]. MSDN Library Archive, 2001.
- 6 UDDI Version 3.0.1, UDDI Spec Technical Committee Specification [S], 2003.
- 7 钟远明、奚建清,分布实时数据库系统中事务处理的研究 [J], 计算机应用研究, 2002. 19 (2): 75 - 78.
- 8 柴晓路、梁宇奇, Web Services 技术、架构和应用 [M], 北京: 电子工业出版社, 2003.
- 9 Dr Frank Leymann, IBM Software Group. Web Services Flow Language (WSFL) Version 1.0, 2001.