开发基于 Web Services 的 Java 卡系统的研究

A Research Based on Web Services to develop Java－Card system

梁俊斌　陆　松　苏德富　（广西大学 计算机与电子信息学院 530004）

摘要：Web Services 是新一代 Internet 分布式应用框架，代表了下一代网络计算和企业应用的必然趋势，有着巨大的应用需求和市场潜力。Java 卡是智能卡领域的主流方向，在现实生活中将发挥越来越重要的作用。因此，将 Java 卡系统的开发建立在 Web Services 之上能使其应用更为广泛。

关键词：Web Services　Java 卡　.Net

1　概述

智能卡在银行、通信、交通等领域发挥着越来越重要的作用，如 IC 电话卡，银行 IC 卡以及手机 SIM 卡等等。它具有大容量、稳定性高、易携带、安全性好等优点[2]。随着智能卡的应用范围不断扩展，传统的智能卡技术暴露出了不少问题：(1)不同开发商的指令集一般各不相同，开发人员的培训费用很高。(2)开发流程复杂，模块化程度不高，重复开发很普遍。(3) 不同厂商的卡片之间兼容性不好。(4) 卡片用途单一，无法在一张卡片上实现多个功能。由此 SUN 公司提出了 Java 卡的概念，目前已经推出了 Java Card 2.2 规范[3]。为了让 Java 卡的应用更加广泛和安全，广西大学计算机与电子信息学院负责的广西大学“校园一卡通”项目就将 Java 卡系统的开发和 Web Services 结合起来，并取得了很好的效果。

2　Web Services

2.1　Web Services 的特点

Web Services 具有以下特点：

(1) Web Services 提供的服务不一定要存在于 Web 上，它可以位于任何网络中，不仅是外部的 Internet 网络，还可以是内联网等。

(2) Web Services 实现平台的细节和业务调用程序无关，可以形成松散耦合的组件系统。

(3) Web Services 是自描述的。

(4) Web Services 是可查找的。

(5) Web Services 是可以互操作的。

(6) Web Services 具有普遍性。

(7) Web Services 具有良好的封装性。

(8) Web Services 使用的是标准的协议，如 WSDL。

(9) Web Services 具有可集成能力。

2.2　Web Services 的体系结构

2.2.1　Web Services 模型

面向服务的 Web Services 体系架构中共有 3 种角色：服务提供者、服务请求者和服务注册代理。简单来说：(1)服务提供者是 Web 服务的拥有者，它为其他服务和用户提供服务功能，在实现服务之后可以发布服务，并响应对其服务的请求；(2)服务请求者是 Web 服务功能的使用者，它可以利用 Web 服务注册代理查找所需的服务，并且向 Web 服务提供者发送请求以获得服务；(3)服务注册代理是把服务请求者与合适的服务提供者绑定在一起，它能够注册已经发布的服务提供者以及所提供的服务，并且提供服务的检索。这 3 种角色是根据逻辑关系划分的，在实际应用中它们可以交叉和互换。

图 1　Web Services 模型

组成 Web Services 体系架构的组件必须具有 3 种角色中的一种或多种，这些不同的角色之间通过发布(Publish)、查找(Find)和绑定(Bind)3 种操作提供完整的 Web Services 功能。操作是通过不同角色的交互来实现的，具体来说：(1)发布操作：服务提供者通过发布操作向服务注册代理注册自己的功能和访问接口；(2)查找操作：服务请求者通过查找操作向服务注册代理查找特定的服务；(3)绑定操作：服务请求者通过绑定操作使用服务提供者所提供的服务。

2.2.2　Web Services 协议栈

为了实现 Web Services 体系架构中的不同操作和交互，需要有一系列分层的协议规范来提供实现。Web Services 体系结构的基本原则之一就是使用通用的标准和技术

(包括服务描述、服务消息通信以及数据格式等)开发与平台和编程语言无关的 Web Services,从而能够充分利用现有资源,实现分布式开发和重用性。Web Services 的协议栈就充分体现了开放和标准的原则。

表 1　Web Services 协议栈

Tool	Layer	Business Issues		
WSFL	Service Flow	Security	Management	Quality of Service
Static→UDDI	Service Discovery			
Direct→UDDI	Service Publication			
WSDL	Service Description			
SOAP	XML-Based Messaging			
HTTP,FTP,SMTP	Transport			
Ipv4,Ipv6	Internet			

其中开始的 5 层是目前开发的 Web 服务的相关标准协议,包括服务调用协议 SOAP、服务描述协议 WSDL、服务发现/集成协议 UDDI 和服务工作流描述语言 WSFL;最后两层是已经定义好的并且广泛使用的传输层和网络层协议的标准;垂直的协议代表协议栈各层必须满足的高层需求,是各个协议层的公用机制,这些机制一般由外部的正交机制完成。

2.2.3 Web Services 相关标准和技术

(1) XML(可扩展标记语言)。XML 不是 Web Services 一个单独的协议层,但它是 Web Services 的核心技术,它为 Web Services 提供了统一的数据格式。消息、服务描述和工作流描述等不同层次协议,都采用 XML 作为定义语言。

(2) SOAP(简单对象访问协议)。SOAP 是用于交换 XML 编码信息的轻量级协议,它主要包括 SOAP 封装(Envelop)、SOAP 编码规则和 SOAP 远程过程调用(RPC)3 个主要方面。SOAP 完全继承了 XML 的开放性和描述可扩展性,而且用 XML 进行消息编码,SOAP 和 XML 是不同企业系统之间跨语言、跨平台的很好的解决方案。

(3) WSDL(Web 服务描述语言)。WSDL 是用来描述网络服务或端点的 XML 语言,它用于定义 Web Services 以及调用的方式。WSDL 文档可用于动态发布 Web Services、查找已发布的 Web Services 并且绑定 Web Services。

(4) UDDI(Web 服务注册规范)。UDDI 提供了在 Web 上描述并发现商业服务的框架,是面向 Web 服务的信息注册中心的实现标准和规范。UDDI 通过服务注册,以及使用 SOAP 访问这些注册信息的约定来实现上述目标[1][4]。

3　Java 卡

3.1　Java 卡简介

Java 卡是一种可以运行 Java 程序的接触式微处理器智能卡,具有动态处理数据和管理文件的功能。它的基本思想就是在智能卡的 ROM 中保持一个 Java 虚拟机 JVM,在这个 JVM 上运行 Java 小应用程序(Java Applet)[2]。它的优点在于[6]:

(1) 支持包括 WINDOWS、UNIX、Solaris 等在内的多种开发平台或集成开发环境。

(2) 遵循通行的智能卡标准,基于对象的 API 简化了卡内 Applet 与终端或后台服务之间的通信,易于开发。

(3) 可以根据需要删除或添加 Java 卡上的 Applet,多个完成不同功能的 Applet 能够运行在同一张卡片上,实现"一卡通"。

(4) Java 拥有一系列其他编程语言所不具备的安全特性,其安全程度远远超过了传统的预编译代码。

3.2　硬件通信

Java 卡遵循 ISO7816 规范,它与终端的通信必需采用 APDU 格式。APDU 即应用协议数据单元,是终端与 Java Card 沟通的格式与协定。Java 卡和终端之间通过 485 串口电缆连接。

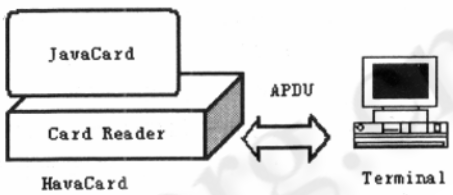

图 2　JavaCard 运作模式

由终端传送给 Java Card 的 APDU 称为 Command APDU,主要用来下达指令以及传输资料给 Java Card。下表即 Command APDU 的格式,其中 CLA 用于识别 applet,INS 表示下达给 applet 的指令,P1、P2 为指令参数,Lc 为发送数据长度,Le 为接收数据长度[6]。

表 2　Command APDU 格式

Mandatory Header				Optional Body		
CLA(1)	INS(1)	P1(1)	P2(1)	Lc(1)	Data Fields(不定长度)	Le(1)

由 Java Card 端回传给终端的 APDU 则称为 Response APDU。下表即 Response APDU 的格式,其中 SW1 和 SW2 是执行状态参数。

表 3　Response APDU 格式

Optional Body	Mandatory Trailer	
Data Fields	SW1	SW2

APDU 的具体内容可以自行确定，也就是说可以制定自己的指令集。

3.3 安全性和事务处理

作为一个智能卡系统，安全性是个重要的问题，因此客户端需要有各种安全性检查。首先是卡片上的数据必需加密，根据 SUN 公司的 Java 卡安全白皮书[7]，javacard.security 和 javacardx.cryptos 包中提供了当前通用的各种加密算法和技术，包括对称/公开密钥算法、数字签名和验证、报文摘要、随机数据产生和 PIN 码管理等等，在支持 DES、RSA 等加密算法的同时也允许用户将自己独有的安全算法和应用程序下载到卡内执行。我们的系统利用 javacard.security 和 javacardx.cryptos 包中提供的方法对卡内关键数据进行 3－DES 加密，此外还提供 PIN 码验证以及对操作合法性验证和权限检查。

事务是分布式系统的重要组件，良好的事务处理有助于增强数据的完整性，防止网络失效、服务器停机和其他意想不到的问题可能对业务系统造成的严重破坏。我们的系统主要利用 Web Services 来实现事务管理，因为 Web Services 中包含有主要的 Java 事务 API，提供了一组简单的面向事务命令。

3.4 Applet 的生成和装载

Java Card 是 Java 平台中最小的子集。根据 Sun 公司公布的 Java 卡规范，目前与 Java 相关的只有 4 个包，包括 java.lang、javacard.framework、javacard.security 和 javacardx.crypto。同时 Java 卡 Applet 不能支持诸如对象克隆、多线程和动态类装载等 Java 语言固有的特性，也不支持 float、double、long、char 等数据类型。

编写 Applet 后将其编译为 .class 文件，然后利用 Sun 公司的 Java Card 开发工具包[8]中提供的转换器（Converter）将 .class 文件转换为可以被装载到 Java 卡上的 .cap 文件，最后使用装载器（Installer）装载到卡上。具体流程如图 3 所示

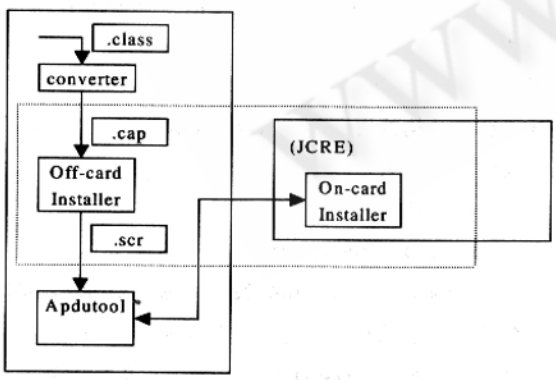

图 3　Java 卡 Applet 的生成和装载

4 开发实例

图 4 是广西大学“校园一卡通”项目的系统结构图。该项目包括门禁、考勤、教师及研究生实验室管理、图书馆管理、学生学籍管理、学生缴费管理等模块，各个模块由学校的不同部门管理，相关数据存储在各部门的服务器之中，它们之间可能需要跨越防火墙进行通信，而且为了和过去已经采用的别的平台开发的成熟的系统进行集成，采用基于 Web Services 的 Java 卡系统在安全性、经济性、适用性和重用性等方面是完全可行的。

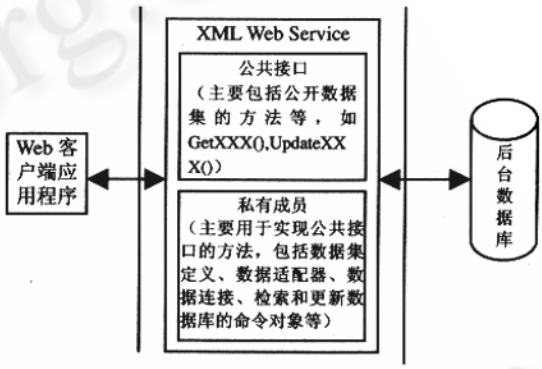

图 4　系统结构图

卡的开发在第三节中已经介绍，接下来是对卡通过卡座传来的信息的处理及对项目各种具体应用的管理，即后台管理系统的开发。我们所采用的开发工具为：在 Windows2000 Server 平台下 Microsoft Visual Studio .NET 和 SQL Server2000。系统的后台管理由三个逻辑层组成：数据层、业务对象层和用户界面层。数据层是 SQL Server 中的数据库；业务对象层处理如何访问数据以及如何将数据分发到客户端；用户界面层针对不同的用户采用了 Web 和传统的 Windows 应用程序两种方式。具体代码由于篇幅的缘故我们不再给出。

5 总结

Web Services 是一个实用的技术，有巨大的应用需求和市场潜力，同时也是在 Internet 上进行分布式计算的基本构造块，其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。Java 卡统一了智能卡的编程接口和编程语言，为智能卡的更大范围的使用提供了基础，真正使智能卡行业成为一个统一标准的产业。到目前为止，代表着全世界 90% 智能卡制造能力的制造商们均支持 Java 卡的使用，其中包括 MORTOROLA 公司、PHILIPS 公司、和 SIEMENS 公司等业界巨头。同时两大信用卡国际组织 VISA 和 MASTER

（下转第 63 页）

(上接第 15 页)

CARD 也都是 Java 卡的有力支持者,Java 卡是智能卡研究发展的主流方向。因此,基于 Web Services 的 Java 卡系统是非常有发展前景的。

参考文献

1 李安渝 等著,Web Services 技术与实现,国防工业出版社,2003。

2 Zhiqun Chen, Understanding Java Card 2.0, www.javaworld.com/javaworld/jw-03-1998/jw-03-javadev.html, 2002-12-05

3 包盛杰, Java 卡的 Applet, http://www-900.ibm.com/developerWorks/cn/java/jcard/index3.shtml, 2002-11-30

4 柴晓路 著,Web 服务架构与开放互操作技术,清华大学出版社,2002。

5 包盛杰,Java 卡概述, http://www-900.ibm.com/developerWorks/cn/java/jcard/index2.shtml, 2002-11-30