

# 基于 J2EE 框架的 Java 智能卡系统的研究

张玉华, 古丽拉·阿东别克

(新疆大学 信息科学与工程学院, 新疆 乌鲁木齐 830046)

**摘 要:** Java Card 技术是目前智能卡领域的主流方向, J2EE 架构技术是当今非常热门的企业级软件开发技术, 将 Java 智能卡技术与 J2EE 技术结合起来, 开发功能强大的集成企业应用, 应该说是 Java 智能卡开发技术的最好归宿。文中概述了 Java 卡及相关技术, 介绍了如何通过 OpenCard 框架将 Java Card 技术与 J2EE 架构结合起来, 对 Java 卡技术进行了总结并提出了展望。

**关键词:** Java 卡; J2EE; JavaCard 远程方法调用; OpenCard 框架; APDU

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1673-629X(2008)04-0182-03

## Study on Java Card System Based on J2EE Framework

ZHANG Yu-hua, GULILA · Adongbieke

(Xinjiang Univ. of Information Science and Engineering, Urumqi 830046, China)

**Abstract:** Java Card technology is the mainstream direction in the field of the Smart Card at present. J2EE technology is very popular in today's enterprise-level software development technology. Combine Java Card technology with J2EE technology to develop powerful enterprise application integration is the best end-result of Java Card technology development. This paper first summarizes Java Card and related technology; and introduces that how to combined Java Card technology with J2EE framework by OpenCard Framework; summarizes and advances on Java Card technology.

**Key words:** Java Card; J2EE; JCRMI; OpenCard framework; APDU

### 1 Java 卡概述

Java 卡是一种可以运行 Java 小应用程序(Applet)的智能卡,它充分利用了 Java“一次编写,随处运行”的能力,使 Java 也能在智能卡和其他存储容量相对匮乏的设备上得以应用<sup>[1]</sup>。卡上有 Java 虚拟机(Java Virtual Machine, JVM)和 Java 卡运行环境(Java Card Runtime Environment, JCRE),在卡中运行的程序叫 Applet,Applet 可以动态装载到 Java 卡上。Java 卡的 API 规范为智能卡制定了一个 Java 语言的特殊子集,目前的版本为 2.2<sup>[1~3]</sup>。

Java 卡的问世,一方面是为了深化 Java 的应用层次,另一方面也是为了将 Java 平台的特性以及 Java 卡的优点带到智能卡上<sup>[4,5]</sup>。

(1)平台独立:Java 卡 Applet 能够在不同卡片的 JCRE 上执行,即通过 JVM 的机制来达到跨平台的能力;

(2)一卡多用:在同一个 Java 卡中能够存放多个 Java 卡 Applet,并且也能够下载新的 Applet,从而达到“一卡多用”的目的;

(3)复用:可以根据需要删除 Java 卡上的应用或增加新的应用,而无需更换新的智能卡,这样大大增强智能卡的灵活性;

(4)与现有智能卡兼容:Java 卡能与国际标准 ISO 7816(智能卡标准)以及工业界标准(如 Europay/Master Card/Visa, EMV)相容;

(5)应用开发简单快速:开发人员无需了解复杂的智能卡硬件和智能卡专用的技术,就可以进行智能卡应用的开发,从而大大减少开发时间和降低开发难度;

(6)开发环境和开发人员丰富:开发人员可以任意选择他们所熟悉和喜欢的开发工具,所有几乎当今所有流行的 Java 开发环境都可以被用来进行 Java 卡的开发。由于任何 Java 开发人员都可以变为 Java 卡开发人员,为智能卡的发展提供了强有力的保证。

### 2 Java Card 应用程序的体系结构

一个典型的 Java Card 应用程序不是孤立的,而是

收稿日期:2007-07-07

作者简介:张玉华(1982-),女,新疆乌鲁木齐人,硕士研究生,研究方向为计算机应用、智能卡应用;古丽拉·阿东别克,硕士,教授,研究方向为计算机应用、民文信息处理。

包含卡片端、读取端和后端元素,如图 1 所示,现在更详细讲述每个元素。

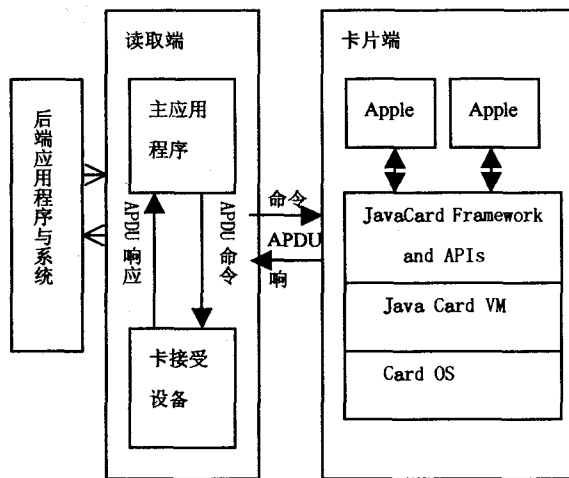


图 1 Java Card 应用程序的体系结构

(1)后端应用程序和系统:后端应用程序提供了支持卡上 Java 小应用程序的服务。例如,可以提供到安全系统和卡上的证书的连接,提供强大的安全性。

(2)读取端主应用程序:主应用程序存在于一个例如 PC 机或者终端、电子付款终端、手机或者一个安全子系统中。它用来处理用户、Java Card 小应用程序和供应商的后端应用程序之间的通讯。

(3)读取端卡片接受设备:卡片接受设备(CAD)是处于主应用程序和 Java Card 设备之间的接口设备,为卡片提供电力,以及与之进行电子或者射频通信。

(4)卡片端小应用程序和环境:Java Card 平台是一个多应用程序环境。在图 1 中可以看到,卡片上可能存在一个或多个 Java Card 小应用程序,还有支持软件——卡片的操作系统和 Java Card 运行时环境(JCRE)。

(5)与 Java Card 小应用程序通讯(访问智能卡):消息传送模型,其核心就是应用程序协议数据单元(Application Protocol Data Unit, APDU)<sup>[4]</sup>。卡片端接收任何 CAD 发送进来的 APDU 命令并且传送到相应的小应用程序中。小应用程序处理 APDU 命令,然后返回一个响应 APDU。

APDU 是读取端(主应用程序)与卡片端(小应用程序)进行交互的格式与协定。它分为 APDU 命令和 APDU 响应,具体的格式见表 1 和表 2<sup>[4]</sup>。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 为接收数据的最大长度; SW1 和 SW2 是执行状态参数,反应了 APDU 命令的执行结果,例如: SW1 + SW2 = 0X9000 表示正常处理返回。

表 1 APDU 命令格式

| APDU 命令          |     |    |    |                |           |    |
|------------------|-----|----|----|----------------|-----------|----|
| Header(required) |     |    |    | Body(optional) |           |    |
| CLA              | INS | P1 | P2 | Lc             | DataField | Le |

表 2 APDU 响应格式

| APDU 响应        |  |                   |
|----------------|--|-------------------|
| Body(Optional) |  | Trailer(required) |
| Data Field     |  | SW1 SW2           |

### 3 OpenCard 框架和 JCRMI 模型简介

OpenCard 框架(Open Card Framework, OCF),是一套基于 Java 的应用程序编程接口,把一些来自不同的供应商的智能卡与读卡器交互的底层通信过程隐藏起来,OCF 的主要目的是用来开发与智能卡进行交互的智能卡外部主应用程序的,这些主应用程序可能运行在 PC 机、网络计算机、ATM 机或者 PDA 移动设备上。通过使用 OCF,程序员可以把精力投入到智能卡外部应用的逻辑实现上去,而不需要关心繁琐复杂的底层通信过程<sup>[5]</sup>。图 1 中的主应用程序就是基于 OpenCard 框架开发的。

当编写一个基于 OCF 的主应用程序时,基本上要把它分离成两个部分:

- \* 和终端或者读取器交互的主应用程序对象(初始化 OCF,等待卡片插入并且终止 OCF),并且够显露高级的卡片访问方法,例如 get Balance()。

- \* 一个实现实际的低级通道管理和 APDU 输入/输出的小应用程序代理。当把 APDU 细节从应用程序中隐藏起来的时候,这个代理(Proxy)设计模式允许你显露一个面向对象接口。

此外,图 1 中与 Java Card 小应用程序通讯还可以采用另一种模型——JavaCard RMI(JavaCard Remote Method Invocation, JCRMI)<sup>[4]</sup>,严格地说它是 J2SE RMI 分布对象模型的微型版,它是一个以对象为中心的模型,根据该模型 APDU 硬件通信的过程将被抽象化,取而代之的是程序员直接处理对象,简化了编程。在 RMI 模型中,一个服务器应用程序创建并生成可访问的远程对象,并且一个客户应用程序获得到远程对象的远程引用,然后调用它们的远程方法。在 JCRMI 中,Java 智能卡小应用程序是服务器,主应用程序是客户端,也就是说,JCRMI 提供了一个基于 APDU 消息传递模型的分布式对象模型机制,通过这个机制服务器和客户端通信,来回传送方法信息、参数和返回值。

### 4 Java 卡与 J2EE 应用的集成构架

通过上面所讲的 OpenCard 框架和 JCRMI 模型,

可以有机地将 Java 智能卡应用与 J2EE 应用结合起来。如图 2 所示是 Java 智能卡应用与 J2EE 应用通过 OpenCard 框架集成的示意图。从图可以看出 OpenCard 主应用程序在 Java 智能卡小应用程序与 J2EE 前端应用程序之间起到了桥接的作用。

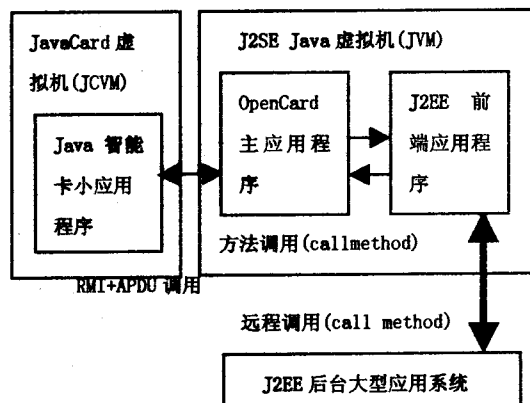


图 2 Java 卡与 J2EE 应用的集成构架

由图 2 可以看出, J2EE 前端应用程序与 OpenCard 主应用程序在同一个虚拟机运行空间下可以通过方法调用直接通信; OpenCard 主应用程序与智能卡中小应用程序的通信过程实际就是交换 APDU 命令序列的过程, 若使用 RMI, 还存在一些远程方法调用的过程; J2EE 前端应用程序主要通过 JNDI (Java Naming and Directory Interface) 调用来完成于 J2EE 后台应用的通信过程。

具体过程是: 当 J2EE 前端应用程序调用 OpenCard 主应用程序中的方法, 试图访问智能卡中的数据信息时, OpenCard 主应用程序会把相应的调用请求转发给智能卡中的小应用程序, 如果该方法调用不要求有任何返回值, 则 OpenCard 直接返回, 否则, OpenCard 将等待小应用程序的返回信息, 并且将返回信息传递给 J2EE 前端应用程序, 最终传递给 J2EE 后台大型应用系统<sup>[5]</sup>。这样就完成了 Java 智能卡应用与 J2EE 应用的集成。

文中以网上书店一个简单的例子验证了 Java 智能卡与 J2EE 集成的可行性。其中客户层开发过程中要考虑的重要问题是 Java 卡上的 Applet 的制作和装载, 读卡器和终端机的通信; 中间层则使用开源的 JBOSS 应用服务器, 提供 EJB 部署、EJB 容器、安全性和事务性等服务; EIS 层采用 MySQL 数据库, 使用 JDBC 连接。整套系统的开发环境为 JDK + JCDK + Eclipse。整个系统的软件结构分为 7 个主要部分, 分别为智能卡小应用程序模块、OpenCard 主应用程序模块、客户端应用软件模块、J2EE 前端应用模块、J2EE 后台大型应用系统模块、数据存取模块与持久化模块。

由于本系统设计的目的仅仅在于验证可行性, 所以仅用 SUN JCDK 提供的智能卡模拟运行环境工具 CREF 对智能卡小应用程序模块部分的执行进行了模拟, 而没有使用真正的 Java 智能卡。

## 5 总结和展望

Java 卡统一了智能卡的编程接口和编程语言, 为智能卡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。使用 Java 卡技术的智能卡是携带数字个人信息和计算能力的最便携和安全的方法, 它是一个今天数字世界中非常强大的并且必要的技术。

近几年, Java 卡技术在国外基本上已经成为当前智能卡的标准, 占了市场的主流。国内 Java 卡技术起步较晚, 但发展迅速, 是智能卡必然的发展方向<sup>[6]</sup>。在信用卡应用方面, Java 卡也日益蓬勃发展。为了保证信用卡交易的安全性, Visa、MasterCard 以及日本最大的信用卡机构 JCB 都在发行以 Java 卡技术为基础的智能卡。国内的银行业也不甘落后, Sun 公司与中国银联宣布合作建立“多应用智能卡联合实验室”, 加快了国内银行卡向智能卡迁移的步伐。现在全球已经有 10 亿多张 Java 卡正在各地使用, 而 Sun 已经发起了一项名为“Java Card S”的计划, 希望将 Java 卡目前的市场逐步扩大到更多的智能卡领域。

实际运用中, Java 卡的运用还有一些瓶颈, 首先就是 Java 卡的成本问题, 由于国内发展较慢, 运用还比较落后, 没有大批量发卡, 造卡成本较高, 反过来又严重制约了普及<sup>[7]</sup>, 我国第二代身份证就是由于成本原因没有使用 Java 卡; 其次是下载问题, 应用模块的增加都需要相应智能卡的动态下载来实现, 如何便捷经济地搭建下载平台也是制约其发展的一个问题。但总的来说, Java 卡是现技术水平下智能卡的一个发展方向, 国内各方面对其发展还是抱有了很大的热情, 另外, 随着企业级多层分布式应用程序开发架构 J2EE 的普遍应用, 将两种技术结合起来, 开发功能强大的集成企业应用, 应该说是 Java 智能卡开发技术的最好归宿。

## 参考文献:

- [1] JavaCard2.2.1 Virtual Machine Specification[EB/OL]. 2003-10-29. URL: <http://java.sun.com/products/javacard/specs.html>.
- [2] JavaCard2.2.1 Application Programming Interface Specification[EB/OL]. 2003-10-29. URL: <http://java.sun.com/>

(下转第 188 页)

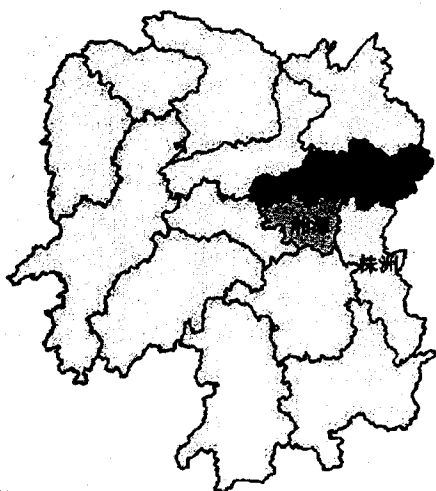


图 2 基于 Ajax 的地图发布效果

的效率,方便了用户在客户端进行图形操作及可视化查询,为系统最终实现网上办公、降低工作负荷、提高工作效率奠定了良好的图形技术基础。然而这种方法

还可以进一步完善,如在如何更合理地提供网络地图的互操作功能等方面还可以进一步改进。

#### 参考文献:

- [1] 游文杰. JavaScript 函数与事件应用[J]. 计算机应用, 2001 (S1):119-120.
- [2] Crane D, Pascarello E, James D. Ajax 实战[M]. 北京:人民邮电出版社, 2006.
- [3] Asleson R, Schutta N T. Ajax 基础教程[M]. 北京:人民邮电出版社, 2006.
- [4] 王兴玲. SVG 与矢量地图的 Web 发布技术[J]. 计算机工程与应用, 2002(10):1-4.
- [5] 杨超伟, 李琦. Web 空间信息发布研究[J]. 北京大学学报:自然科学版, 2001, 37:413-419.
- [6] 王志兵, 李满春, 李响, 等. 基于 IMS 的 WebGIS 应用开发[J]. 计算机应用研究, 2001, 18:120-121.
- [7] 马小虎. 多媒体数据压缩标准及实现[M]. 北京:清华大学出版社, 1997.

(上接第 123 页)

任务队列的阈值阈长和主辅关系的解除时机来控制负载均衡。经过测试,在发生大量并发任务,即有重载时,服务器和终端设备通信次数约 120 次/ms 与单服务器下最大通信次数约 80 次/ms 相比,提高约 50%。这表明,该模型的负载均衡算法可行,系统额外开销小,能有效改善动态任务的分配和调度,增强了动态负载均衡的自适应能力。下一步的工作是对一些具体算法进行优化和进行详细的性能测试。

#### 参考文献:

- [1] DI Serio A, Ibanez M B. Distributed load balancing for molecular dynamics simulations[C]// New York: 16th Annual International Symposium on High Performance Computing Systems and Applications. [s.l.]: IEEE, 2002:284-289.
- [2] Kameda H, Fathy El-Z S, Ry U I, et al. A performance comparison of dynamic vs. static load balancing policies in a mainframe - personal computer network model[C]// New

York: Decision and Control. New York: IEEE, 2000:1415-1420.

- [3] Zomaya A Y, Yee-Hwei Teh. Observations on using genetic algorithms for dynamic load balancing[J]. IEEE Trans parallel and Distributed systems, 2001, 12(9):899-911.
- [4] Xing Y, Zdonic S, Jeong-Hyon. Dynamic load distribution in the Borealis stream processor[C]// The 21st International Conference on Data Engineering. Tokyo, Japan: ICDE, 2005: 791-802.
- [5] 谢季坚, 刘承平. 模糊数学方法及其应用[M]. 第 2 版. 武汉:华中科技大学出版社, 2000.
- [6] Naor Z, Levy H. LATS: a load - adaptive threshold scheme for tracking mobile users[J]. IEEE/ACM Transactions on Networking (TON), 1999, 7(6):808-817.
- [7] Horinan, Wattanachai, Benjapolakul. The performance analysis for fuzzy inference system - based adaptive soft handoff thresholds[C]//2004 IEEE Region 10 conference. TEN-CON: IEEE Computer society, 2004.

(上接第 184 页)

products/javacard/specs.html.

- [3] JavaCard2. 2.1 Runtime Environment(JCRE) Specification [EB/OL]. 2003-10-29. URL: <http://java.sun.com/products/javacard/specs.html>.
- [4] Java Card 应用程序开发三部曲[EB/OL]. 2005-11. URL: <http://www.zasp.net/>.

- [5] 林胜利, 路宗强, 王坤茹. Java 智能卡开发关键技术与实例[M]. 北京:中国铁道出版社, 2006.
- [6] 游代安, 何久田, 蒋遂平, 等. Java 卡应用的设计与实现[J]. 计算机工程与应用, 2006, 42:229-231.
- [7] 傅俊, 许柳威. Java Card 技术运用于校园一卡通的探讨[J]. 现代计算机, 2006(7):87-90.