

# 用 J2ME 构建企业移动电子商务系统

## Build Enterprises Mobile E - Commerce System with J2ME

徐翔斌 方水良 (杭州市 浙江大学玉泉校区 先进制造工程研究所 308 室 310027)

**摘要:** 本文在探讨利用 J2ME 技术将企业的电子商务活动扩展到无线移动领域的可行性和关键技术的基础上,重点介绍如何设计出用于在 J2ME 和 J2EE 系统之间通信的耦合模块和数据传输对象。并且开发了一个完整的移动电子商务系统。

**关键词:** 无线移动通信设备 J2ME J2EE MIDlet Servlet 耦合模块 数据传输对象(DTO)

### 1 引言

随着移动通信网络的发展和个人移动通信设备的普及,用手机、PDA 和其他嵌入式设备从网络中获取信息已经成为现代人生活的一部分,同时传统意义上的电子商务活动也开始向移动通信领域内扩展,为此,企业必须改造和升级现有的电子商务系统,构建自己的移动电子商务系统。利用无线互联技术,企业可以将电子商务的终端用户从企业内部的 Intranet 或广域的 Internet 中的客户端扩展到不受空间位置限制的移动用户,从而拓展了企业的需求。而作为企业的

高效利用企业现有的资源,降低改造成本,最好是将移动电子商务系统构建在现有的电子商务系统的基础上,并且对其进行扩展。笔者在为某企业的移动电子商务系统改造过程中,用 SUN 公司的 J2ME(Java 2 Platform, Micro Edition)开发平台在基本不改变系统的情况下,对原有的系统进行扩展和升级,取得了良好的效果。

### 2 企业现有的电子商务系统

和大多数企业的电子商务系统一样,该企业的电子商务系统是一个典型的 J2EE (Java 2 Platform, Enterprise Edition)系统。它能提供网上产品展示,选购等功能。J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。从结构上看,该 J2EE 的电子商务系统分为表示层,WEB 层, EJB 层,企业信息层四个层次。各层在编程过程中为相互独立的模块,它们之间通过标准的 Java API 接口函数调用和传递参数。该系统数据库采用微软的 SQLServer2000,部署在 Weblogic 服务器中,运行的操作系统为 Win2000 Server,其系统结构如图 1 所示。

从图 1 中可以看出,该系统是基于 WEB 的典型的 B/S 型 J2EE 系统,对该系统的改造和升级的关键是在不要改动其他三个层的情况下,将表示层扩展成为具有 Java 功能的手机和 PDA 等无线移动终端设备。

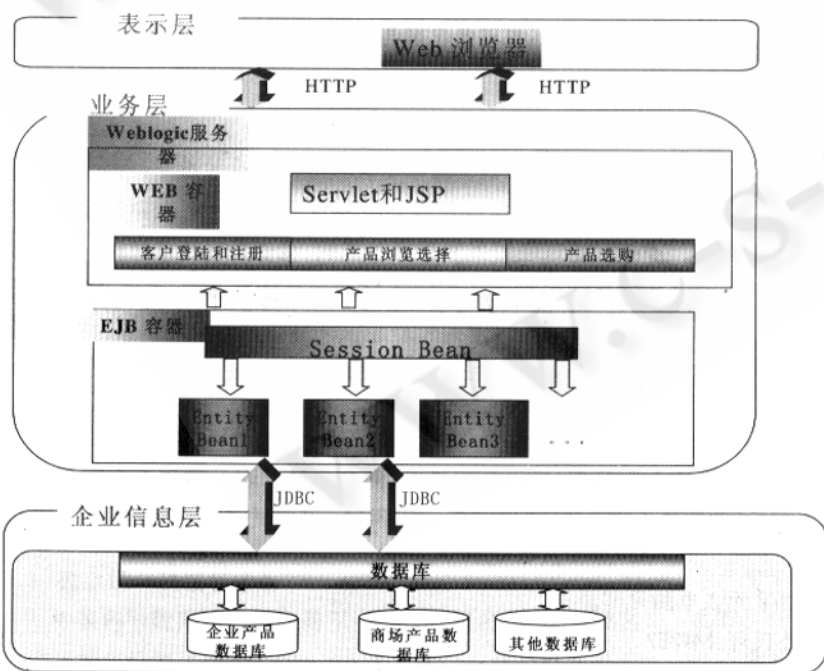


图 1 J2EE 系统结构图

管理者在出差途中,生产现场等因受限而不能接触互联网的情况下也可以通过移动通信设备来及时了解商品的销售和库存等情况(在获得相应的权限的前提下)。为了最大程度、

### 3 J2ME 简介

J2ME 是 SUN 公司在 1999 年推出的针对资源受限设备(手机, PDA)进行二次开发的 Java 开发平

台。它具有 Java 语言的简单,安全与平台无关,代码可以移植,网络传输可靠等优点,有着与 J2EE (Java 2 Platform, Enterprise Edition) 系统无缝结合的能力。目前市场上支持 J2ME 的无线移动产品有很多,用 J2ME 平台开发的游戏,工具(如字典,地图)和移动电子商务系统有着广阔的应用前景。

为了为小型的、资源有受限制的连接设备定义一个统一的 Java 开发标准,SUN 公司规定了有限连接设备配置 (CLDC),它的核心是 SUN 的 K 虚拟机 (KVM),能够在大多数像无线通信设备这样的小型设备上运行,使得编程人员能够轻松地开发出在它们上面运行的 Java 应用程序。在

进行控制,MIDlet 位于 javax.microedition.midlet 包中,它必须在所有的 MIDP 应用程序中引入。

## 4 J2ME 和 J2EE 系统的耦合

### 4.1 系统结构图

经过改造和升级后的系统结构图如图 2 所示。

从图 2 可以看出,同属于 SUN 公司的 J2ME 和 J2EE 体系具有良好的兼容性和耦合性。在系统改造和升级的过程中,充分利用了现有的资源,几乎没有更改原有的构架。J2ME 层和现有的系统之间的通行方式采用标准的 HTTP 协议,两者之间耦合的关键技术是设计出能够连接两者之间的

耦合组件和相应的数据传输对象 (Data Transfer Object, 简称 DTO),耦合组件能够对 DTO 进行传输、解析和识别,而 DTO 不仅要能够提供足够的产品和客户信息,还能够序列化自己。

### 4.2 数据传输对象 (Data Transfer Object)

由于无线网络中的 HTTP 延迟时间较长,因此应该努力减少 Midlet 使用的 HTTP 的往返传输次数。尽管互联网的浏览器可以进行多次 HTTP 往返传输来获取某个网页上不同的框架和图像,但是位于无线移动设备中 Midlet 却应该一次性的获取它的全部内容。因此我们必须对 J2ME 和 J2EE 之间传输的数据进行封装。而且 J2ME 是基于内存有限设备的,他不支持对象的序列化技术,数据的传输无法利用对象的序列化技术来进行封装而导致一个数据对象的

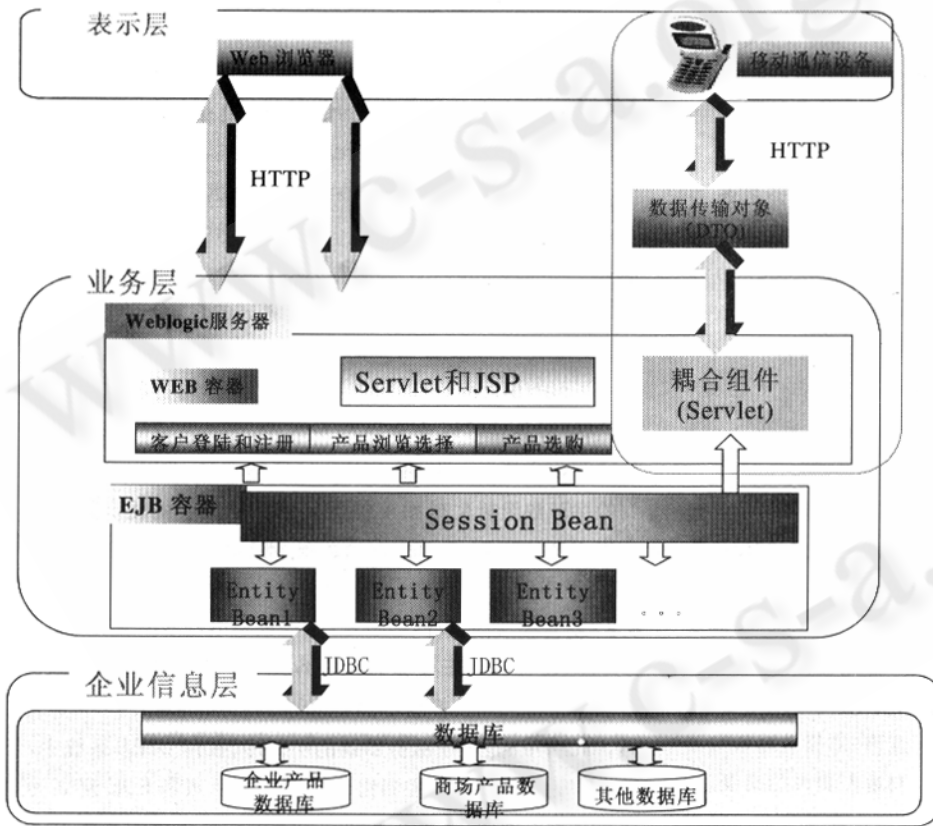


图 2 集成 J2ME 移动电子商务系统

CLDC 上的 MIDP (J2ME Mobile Information Device Profile) 类库和 API 函数许多都来自 J2SE,为了实现特定的功能,MIDP 也增加了一些新的类库。基于 MIDP 类库开发的 J2ME 应用程序称为 MIDlet,它与 J2SE 中的 Applet 十分相似。它们不能单独运行,都必须运行在特定的运行环境之中,并且由该运行环境来管理它们的生命周期。编写 MIDP 应用程序的要求就是必须扩展 MIDlet 类,这个类定义了一些接口,应用管理软件可以通过这些接口对 MIDlet

的各个字段传输分多次进行,客户无法忍受较长的等待时间。为此,笔者在 J2ME 中模拟对象序列化技术,设计了几个能够在 J2EE 和 J2ME 之间传输的数据传输类。对需要在两者之间传输的大量数据进行了封装,达到一次传输多个数据,改善 J2ME 无线通信的带宽所带来的缺陷。一个典型的数据传输对象的类如下:

```
public class Watch
{
```

```

private String watch_name;
private long watch_price;
private String watch_description;
public Person()
{
}
public Person(String watch_name, long watch_
price,String watch_description)
{
    this.watch_name= watch_name;
    this.watch_price= watch_price;
    this.watch_description= watch_description;
}
public String getWatch_name()
{
    return watch_name;
}
public void setWatch_name (String watch_name)
{
    this.watch_name= watch_name;
}
public long getWatch_price()
{
    return watch_price;
}
public void setWatch_price (long watch_price)
{
    this.watch_price= watch_price;
}
public String getWatch_description()
{
    return watch_description;
}
public void setWatch_description (String watch
_description)
{
    this.watch_description= watch_description;
}
public byte[] serialize() throws IOException
{
    ByteArrayOutputStream bout = new ByteAr-
rayOutputStream();
    DataOutputStream dout = new DataOutput-

```

```

Stream(bout);
    dout.writeUTF(watch_name);
    dout.writeLong(watch_price);
    dout.writeUTF(watch_description);
    dout.flush();
    return bout.toByteArray();
}
public void deserialize(byte[] data) throws IO-
Exception
{
    ByteArrayInputStream bin = new ByteArrayIn-
putStream(data);
    DataInputStream din = new DataInputStream
(bin);
    watch_name = din.readUTF();
    watch_price= din.readLong();
    watch_description = din.readUTF();
}
}

```

该类封装了手表的产品信息。手表的名称、价格和产品说明等信息被封装在该类的实例中,通过它身的序列化方法——`serialize()`能够将产品信息的文字描述转化成字节数组,字节数组可以安全地在网络中传输给 J2EE 系统,其反序列化方法——`deserialize()`能够将从 J2EE 系统接收到的字节数组转化成相应的产品信息的文字描述。其他几个用于产品搜索,定单处理的数据传输对象也采用类似的设计方法。

#### 4.3 耦合模块的设计

J2ME 和 J2EE 都是基于 Java2 平台的两套不同的 API 体系。两者之间的耦合和数据传输同样可以用 Servlet 进行,Servlet 是位于服务器端的小程序,在用 J2EE 系统中,它被广泛的用来作为控制组件在控制表示层和 EJB 层之间信息交换。为此,笔者在不触动原有 J2EE 系统的基础上设计了几个专门和 J2ME 通信的 Servlet,并且将它们有效的集成到 J2EE 中,充当 J2EE 和 J2ME 之间数据传输的桥梁和纽带。这些 Servlet 扩展 HttpServlet,用能够透过防火墙的 HTTP 方式来访问企业的 J2EE 系统,在 J2ME 和 J2EE 之间进行 HTTP 通信。J2ME 端的无线设备与 PC 服务器进行 TCP/IP 的 HTTP 通信方式如下代码所示:

```

HttpConnection conn;
conn = (HttpConnection) Connector.open (" ht-
tp://localhost:8080 /servlet /searchservlet");
InputStream in= conn.openInputStream();

```

```
OutputStream out= conn.openOutputStream();
```

专门用于产品搜索的 `searchservlet` 在产品搜索过程中,它接收从移动用户端发送过来的查询条件,从企业数据库中搜索出合乎条件的数据,然后将这些数据进行封装成能够被 J2ME 系统识别的数据传输对象(DTO)。DTO 自身序列化成为字节数组后被该 Servlet 安全的传输到 J2ME 系统,数据传输对象在 J2ME 系统反序列化,经过 J2ME 系统的处理后以适当的形式显示在移动用户的手机或者 PDA 上。用于订单处理的 Servlet 能够接收移动用户在手机或者 PDA 的订购信息(以 DTO 的形式),这些信息在 J2EE 系统中被处理后生成一份完整的企业订单储存在企业的数据库中,而表示移动用户的一笔交易已经完成。

## 5 原型系统

用 J2ME 平台构建的移动电子商务系统能够实现基于互联网的电子商务系统的大部分功能,它能提供移动用户登陆,注册用户信息,查询和选购商品,并且能够实现在线填写订单和支付交易的能力。

该系统的开发环境是:系统环境 win2000 server,平台支持:J2ME Connected, Limited Device Configuration 2.0 (CLDC), J2ME Mobile Information Device Profile

2.0 (MIDP); 开发工具:J2me Wireless Toolkit 2\_0\_01。

## 6 结束语

无线移动设备作为一种屏幕和内存等资源受限设备,无法像 PC 机那样有很强的数据计算和图像处理能力,键盘输入快捷性不强,在系统的改造和升级的过程中,应该考虑将数据计算和图像处理放在 J2EE 层进行,然后将计算和处理的结果经过编码通过网络传输给无线移动设备。考虑到无线移动设备屏幕受限等因素,对需要传输和显示的数据和图像进行全面分析,在有限的显示空间内尽可能地提供有效、直观、丰富的文字和图片信息。

### 参考文献

- 1 Roger Riggs, J2ME 无线设备编程[M],机械工业出版社,2001。
- 2 焦祝军、张威, J2ME 无线通信技术应用开发[M],北京希望电子出版社,2002。
- 3 sun 无线开发网址 <http://wireless.java.sun.com/>。
- 4 Calvin Austin Monica Pawlian Java2 高级编程[M],机械工业出版社,2001。