

JAVA 技术在首钢管理信息系统中的应用

王 伟

(北京首钢计量自动化系统工程有限公司)

摘 要 信息化是继工业化之后世界经济的又一场革命,是当今世界经济和社会发展的趋势。如今,信息已成为生产力的重要因素,企业信息化水平已成为企业应对市场应变能力和参与市场竞争能力的重要标志。本文结合 JAVA 在企业信息化建设中的应用,向读者介绍 JSP、Servlet、JDBC、JavaBean 等 JAVA 技术的概念和区别,并介绍了目前信息化建设中较流行的应用开发模式。

关键词 JAVA B/S MVC 应用模型

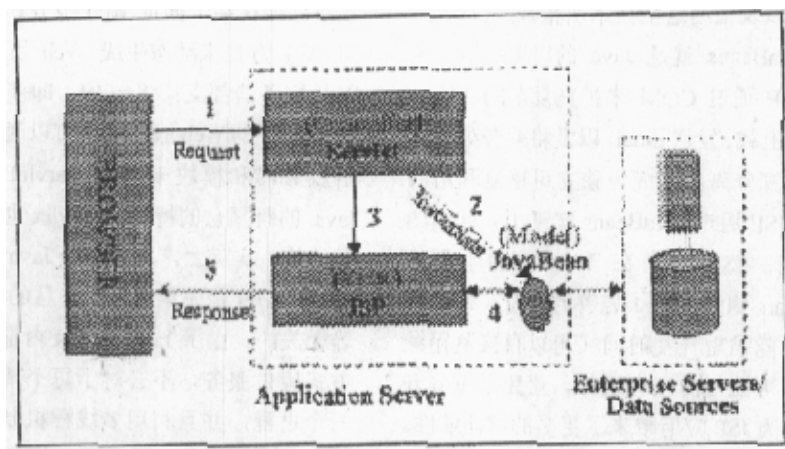
目前,首钢正在大规模开展管理信息系统建设,这必将给首钢的管理带来新的活动力和变革,使首钢加快现代化建设的进程,增强与世界钢铁诸强的对抗力和竞争力。一期工程除基础网络设施建设外,包括办公自动化、财务、调度、经贸、原料和材料 6 个子系统,其中办公自动化采用 Lotus 进行二次开发,财务系统购买重钢较成型产品,材料系统由首钢总计控室计算站和共达公司合作开发。考虑到工期短、任务重等情况以及与先进技术接轨、为首钢培养一只技术过硬的软件研发队伍的需求,决定对调度、经贸、原料和材料采用 Java 技术的 N 阶层 B/S 结构进行开发。

Java 是一种简单易用、完全面向对象、具有平台无关性且安全可靠的主要面向 Internet 的开发工具。自从 1995 年正式问世以来,Java 的快速发展已经让整个 Web 世界发生了翻天覆地的变化。随着 Java Servlet 的推出,Java 在电子商务方面开始

崭露头角,最新的 Java Servlet Page 技术的推出,更是让 Java 成为基于 Web 的应用程序的首选开发工具。而浏览器/Web 服务器/数据库的结构为企业所关心的问题提供了解决方案。嵌入在客户机和服务器之间,位于企业防火墙之前的 Web 服务器建立于基于数据库的应用程序变得相当的非常简单。预装在最终用户台式电脑中的浏览器使开发人员不再需要为应用程序编写复杂的图形用户接口了。如果这个应用程序受欢迎,企业只需简单地扩大其规模,并把工作量分布在多个 Web 服务器上即可。此外,因为浏览器可自动下载用户所需要的任何图形用户接口以及万维网主页,所以信息技术企业也不再需要参与软件分发了。

本次模式采用 JavaSoft 提供的 JSP 应用模型 2。是目前网站应用建模的业界标准之一。

模型结构图如下：



JSP 应用模型 2

服务器端系统栈结构如下：

JDBC	
Serviet	JSP
WebLogic/ISP ASP	
UNIX	
康柏主机	

服务器端系统结构图

客户端系统栈结构如下：

Applet	信息
BROWSER	
OS	
主机	

客户端系统栈结构图

下面为读者简要介绍一下在该模型中用到 JavaBeans、JDBC、Java Servlet 和 Jsp 等技术以及采用这些技术所带来的优势。

JavaBeans 就是 Java 的可重用组件技术。ASP 通过 COM 来扩充复杂的功能，如文件上载、发送 email 以及将业务处理或复杂计算分离出来成为独立可重复利用的模块。JSP 通过 JavaBeans 实现了同样的功能扩充。JSP 对于在 Web 应用中集成 JavaBean 组件提供了完善的支持。这种支持不仅能缩短开发时间（可以直接利用经测试和可性任的已有组件，避免了重复开发），也为 JSP 应用带来了更多的可伸缩性。JavaBean 组件可以用来执行复杂的计算任务，或负责与数据库的交互以及数据提取等。在实际的 JSP 开发过程中，读者将会发现，和传统的 ASP 或 PHP 页面相比，JSP 页面将会是非常简洁的，由于 JavaBean 开发起来简单，又可以利用 Java 语言的强大功能，许多动态页面处理过程实际上被封装到了 JavaBean 中。

JDBC 是用于执行 SQL 语句的 Java 应用程序接口，由一组用 Java 语言编写的类与接口组成，在 JSP 中将使用 JDBC 来访问数据库。JDBC 是一种规范，它让各数据库厂商为 Java 程序员提供标准的数据库访问类和接口，这样就使得独立于 DBMS 的 Java 应用程序的开发工具和产品成为可能。一般的 Java 开发工具都带有 JDBC-ODBC 桥驱动程序，这样，只要是能够使用 ODBC 访问的数据库系统，也就能够使用 JDBC 访问了。

Java Servlet 是 JSP 技术的基础，而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成，Servlet 这个名称大概源于 Applet，可以称之为“小服务程序”。Servlet 其实和传统的 CGI 程序和 ISAP、NSAPI 等 Web 程序开发工具的作用

是相同的，在使用 Java Servlet 以后，用户不必再使用效率低下的 CGI 方式，也不必使用只能在某个固定 Web 服务器平台运行的 API 方式来动态生成 Web 页面。许多 Web 服务器都支持 Servlet，即使不直接支持 Servlet 的 Web 服务器也可以通过附加应用服务器和模块来支持 Servlet。得益于 Java 的跨平台的特性。Servlet 也是平台无关的，实际上，只要符合 Java Servlet 规范，Servlet 是完全平台无关且的 Web 服务器无关的。由于 Java Servlet 内部是以线程方式提供服务，不必对于每个请求都启动一个进程，并且利用多线程机制可以同时为多个请求服务，因此，Java Servlet 效率非常高。

但 Java Servlet 也不是没有缺点，和传统的 CGI、ISAPI、NSAPI 方式相同，Java Servlet 是利用输出 HTML 语句来实现动态网页的，如果用 Java Servlet 来开发整个网站，动态部分和静态页面的整合过程简直就是一场恶梦。这就是为什么 SUN 还要推出的 Java Server Pages 的原因。

前面说过，Java Servlet 的最大缺点就在于没有把网站的逻辑和页面的输出分开，导致整个 Servlet 代码混乱不堪。为了解决 Java Servlet 的这种缺点，SUN 推出了 Java Servlet Pages-JSP。

按照脚本语言是服务于某一个子系统的语言这种论述，JSP 应当被看作是一种脚本语言，然而，作为一种脚本语言，JSP 又显得过于强大了，在 JSP 中几乎可以使用 Java 类。作为一种基于文本的、以显示为中心的 开发技术，JSP 提供了 Java Servlet 的所有好处，并且，当与一个 JavaBeans 类结合在一起时，提供一种使内容和显示逻辑分开的简单方式。分开内容和显示逻辑的好处是，更新页面外观的人员不必懂得 Java 代码，而更新 JavaBeans 类的人员

也不必是设计网而的行家里手就可以用带 Java Beans 类的 JSP 页面来定义 Web 模板, 以建立一个具有相似的外观的页面组成的网站。JavaBeans 类完成数据提供, 这样在模板中就没有 Java 代码, 这意味着这些模板可以由一个 HTML 编写人员来维护。当然, 也可利用 Java Servlet 来控制网站的逻辑, 通过 Java Servlet 调用 JSP 文件的方式将网站的逻辑和内容分离。

模型2中采用MVC(Model, View, Controller)结构理论, 该理论是美国施乐公司的 Smalltalk 早年提出的产物。Model 体现了应用的对象, View 体现了用户界面的表现, Controller 完成的是对用户界面的表现和应用的对象之间的控制。由于 Java Servlet 拥用对逻辑处理非常强的特征, 所以担当 Controller 的角色非常适合。Java Beans 的完整性, 独立性, 灵活性, 可再利用性等优点使其所以担当的 Model 的角色非常适合。Java 的面向对象性, 与平台的无关性, 安全性, 坚固性, 与 WWW 技术的适应性等优点使得我们采用了述应用模型。

在基于组件的 J2EE 平台充分内置了灵活性的情况下, 剩下的问题可能是如何组织应用程序以实现简单高效的应用程序

升级和维护, 以及如何让不懂程序代码的人员避开程序数据。答案就在模型、视图和控制器架构 (MVC) 的使用之中。MVC 这样的构架是一个描述重现的问题及其解决方案的设计范式, 但每次问题重现时, 解决方案都不完全相同。MVC 设计范式包括三种对象: 模型 (model) 提供应用业务逻辑 (Enterprise Beans 类); 视图 (view) 则是其在屏幕上的显示 (HTML 页面、JSP 页面、Swing GUI); 控制器则是 Servlet、JavaBeans 或 Session Beans 类, 它用于管理用户与视图发生的交互。并且, 通过将控制器和模型代码保持在视图之外, 那些不理解这些代码的人员就不能改变他们不应改变的东西。

将控制器和模型分开就可以在不影响模型的情况下改变控制器, 也可以在不影响控制器的情况下改变模型。例如, 如果应用的前端是一个 HTML 页面, HTML 专家就可以更新它。如果使用一个 JSP 页面, 将控制器的代码放到一个 JavaBeans 或 SessionBeans 类中, 或使用动作标记 (action tags), 这样, JSP 页面就仅包含 JSP 代码了。