ISSN 1002－4956
CN11－2034/T

实 验 技 术 与 管 理
Experimental Technology and Management

第 28 卷 第 4 期 2011 年 4 月
Vol. 28 No. 4 Apr. 2011

一种轻量级企业应用 Web2.0 开发框架——Beehive＋ExtJs＋Json

来天平，欧阳荣彬，王素美，彭一明

（北京大学 计算中心，北京 100871）

摘 要：企业级应用正在积极向 web2.0 转型。在介绍了 Beehive、Extjs、Json 技术后，依据 MVC 模型，对 Beehive＋ExtJs＋Json 的 Web2.0 的开发框架的表示层、控制层、模型层的内容和实现分别进行阐述，同时，总结了本框架的一些特性。

关键词：Web2.0；AJAX；开发框架；ExtJs；Json

中图分类号：TP311.52 **文献标志码**：B **文章编号**：1002-4956(2011)04-0296-03

Development framework of a lightweight enterprise application Web2.0: Beehive＋ExtJs＋Json

Lai Tianping, Ouyang Rongbin, Wang Sumei, Peng Yiming

(Computer Center, Peking University, Beijing 100871, China)

Abstract: Abstract Enterprise applications are actively restructuring to Web2.0. First, this paper introduces the Beehive, Extjs, Json technologies, and then based on MVC model, explains the content and implementation of the Beehive ＋ ExtJs ＋ Json of Web2.0 development framework layer, control layer, model layer. At the same time, some of the characteristics of this framework are summed up.

Key words: WEB2.0; AJAX; development framework; ExtJs; Json

1 背景

随着 Web2.0 概念的提出，企业级应用也正积极地从 Web1.0 到 Web2.0 的方向转型[1-2]。在 Web2.0 中实现方法中，Ajax 技术具有重要的地位。Ajax 是 Web2.0 的核心技术之一。SUN 公司提供了 J2EE 开发规范。据此规范，目前有许多丰富的开发框架，如 DWR、DOJO、Struts、Spring 等，但究竟如何避开复杂纷繁的技术，将 Ajax 技术方便快捷地应用于 Web2.0 开发还存在诸多难点。对于大多数企业应用，轻量级开发基本可以满足，基于此，本文提出了利用多种技术组合的一种轻量级的开发框架。

2 技术简介

2.1 Beehive

Beehive 是一个 Apache 项目，用于简化 J2EE 以及 Web 服务编程，其支持 JPF(java page flow)、控件技术、基于 Java Specification Request(JSR)181 的 Web 服务三大功能[3-5]。

JPF 技术形成控件器层。JPF 基于 Struts 构建，但省略了其诸多繁琐的工作(主要是省略了部署配置文件的管理)。控件是一些封装在 EJB 或消息驱动 bean 中的业务逻辑组件，为所有的资源集(数据库、外部系统等)提供了一组通用的接口。控件通过提供一种允许访问各种类型的资源的统一客户机模型，从而降低了作为一个 J2EE 资源的客户机的复杂性。JSR 181 是基于 JSR 171 注释标准的 Web 服务所使用的一种元数据标准。Apache Beehive 使用 JSR 181 来定义一组注释，可以使用它们将任何 Java 类展示为 Web 服务。

在 MVC 编程模型中(见图 1)，JPF 技术形成控件器层，NetUI 标记库则会形成视图层。模型层使用 Java 控件构建。

2.2 ExtJs

首先简单介绍 Ajax 技术。

收稿日期：2010-12-27

作者简介：来天平(1977—)，男，山西晋城，硕士，工程师，研究方向：计算机软件开发与应用.

E-mail：lai@pku.edu.cn

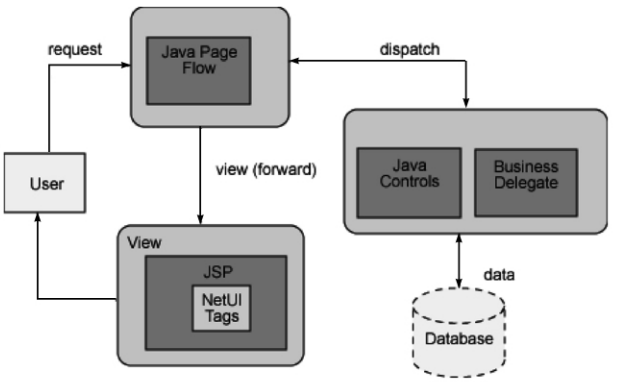

图 1 MVC 模型及 Beehive 技术

Ajax 并不是一种新的技术，而是几种早已出现的技术的综合应用，是一种新的互联网应用设计思想和实现方式，涉及 HTML、XML、DOM、CSS 以及 JavaScript、Json 技术。其核心是 XMLHttpRequest 对象。XMLHttpRequest 使 JavaScript 脚本能够在后台发送 HTTP 请求并获取和处理服务器的响应的内容。通过 XMLHttpRequest 对象提供的 open、send 方法，可以向服务器发送请求，open 方法可以将请求设置为异步方式，置于服务器处理请求的过程中。客户端不需要等待，不影响客户端用户的其他操作，当接收到服务区响应后，JavaScript 脚本通过 XMLHttpRequest 对象的 responseText 或者 responseXML 获取响应内容。

ExtJs 是基于 Ajax 技术的具体的开发框架，本质上是一系列的 JavaScrpit 包，包括了例如 Form、Grid、Window、Combobox 等大量的 UI 组件模型，方便地提供了丰富的客户界面。

2.3 Json

Json(javaScript object notation) 是一种轻量级的数据交换格式，易于人阅读和编写，同时也易于机器解析和生成，它是 JavaScript(standard ECMA-262 3rd edition -december 1999)的一个子集。对于应用 Ajax 的 Web 2.0 网站来说，Json 是目前最灵活的轻量级方案。Json 的基本规则很简单：对象是一个无序的“名称/值”集合。一个对象以“{”(左括号)开始，“}”(右括号)结束。

3 框架内容

本文提出的框架核心思想是标准的 MVC 模型。如图 2 所示。

表现层由 ExtJs 实现，控制层利用 Beehive 的 Controller，而模型层包括 beehive 的 Business Controls 和 DB Controls。用户通 Extjs 的 UI 组件从浏览器发出 Request 请求至 Controller，Controller 根据控制请求调用 Business Control 处理业务。其中和数据库的连接及数据交互通过 Beehive 的 DB control 实现。业务处理完毕后，Controller 在 Response 中写 Json 数据，Extjs 负责解析数据将最终结果在浏览器中呈现给用户。

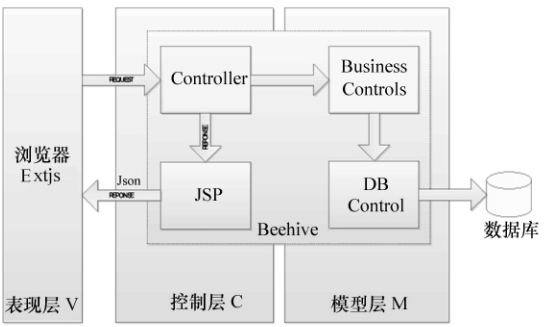

图 2 MVC 模型

下面对模型中各个部分详细说明。

3.1 表现层

表现层采用 ExtJs 实现，原 Beehive 中 Jsp 的页面展示功能全部由 ExtJs 来代替。Jsp 页面中只需定义引入的 js 源路径及 ExtJs 所需的 Div 的定义。举例如下：

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
//引入 ExtJs 编辑的 js。其包括了所有的页面展示及与服务器的数据交互
<script type="text/javascript" src="${pageContext.request.contextPath}/FileUploadField.js"></script>
//定义 ExtJs 的 UI 组件所需的 div
<div id="FileUploadField_div" style="width:100%;height:100%;"></div>
```

在浏览器中，js 代码展示业务处理界面。用户的操作请求通过 ExtJs 的相关 UI 组件(例如 form 组件)发送至服务器的控制层，同时 ExtJs 负责解析从服务器传回的数据，然后利用浏览器展示给用户。全部的操作响应及界面展示全部有 js 实现。

3.2 控制层

控制层采用 Beehive 中的 Controller 实现。Controller 本质上是一个 Java 类，其定义如下：

```
@Jpf.Controller
public class Controller extends PageFlowController{
//@Jpf.Controller 是 NetUI 的标签。Controller 中包含了 Action，每个 Action 是一个 JAVA 方法。其定义如下：
@Jpf.Action(
```

```
        forwards = {
            @Jpf.Forward(name="someName", path="somePath.jsp", [...other properties...])
        }
    )
    public Forward someMethod()
    {
        PrintWriter writer = getResponse().getWriter();
        writer.println(jsonObject);//在 Response 中写 json 数据
        Return null
    }
}
```

以上定义了 Action SomeMethod。Action 的主要作用是供表示层的 JavaScrpit 调用，根据页面提供的值，在进行业务逻辑判断后，通过 response 对象向客户端输出数据。由于在本框架中利用 ExtJs 作为页面展示，所以页面的返回元素均为 null。

3.3 数据层

数据层采用 Beehive 中的 JDBC 控件实现。其本质上也是一个 Java 类。示例如下：

```
@ControlExtension
    @JdbcControl.ConnectionDataSource(jndiName = ConstantCom.JNDINAME)//定义 jdbc 所需的 jndi
    public interface AdmissionComDB extends JdbcControl {
    static final long serialVersionUID = 1L;
    @SQL(statement="SELECT name FROM customer WHERE custid={customerID}")
    public String getCustomerName(int customerID);
    }
```

示例中定义了一个访问数据的方法 getCustomerName，实质是会向数据库传输 sql 语句：SELECT name FROM customer WHERE custid = ?，而将数据库处理结果作为方法 getCustomerName 的返回值。控件可以支持 JAVABEAN、HASHMAP、HASHMAP[]等多种类型。

3.4 业务层

业务层采用 Beehive 中的自定义控件技术实现。本质上是一组接口与实现，包括：

(1) control public interface：定义一组 public 的方法(operations)和事件(Events)。这些方法实现对 Resource 的处理。

(2) control bean class：提供实现 implementation class 的属性访问 JavaBean 类。

(3) control implementation class：对于 interface 的具体实现。

其流程如图 3 所示。

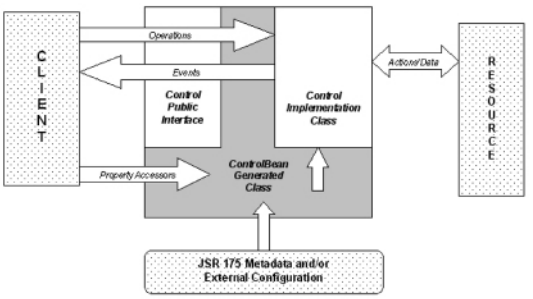

图 3 控件访问流程图

Java 类示例如下：

接口定义：

```
@ControlInterface
    public interface JmsMessageControl
    {
      ...public String handleSome()
    }
```

实现定义：

```
@ControlImplementation(isTransient=true)
public class JmsMessageControlImpl implements JmsMessageControl
{
    public String handleSome(){}
}
```

在业务层中，可以通过对具体的方法定义特殊属性实现事务处理。业务层一般由控制层直接调用。调用方法和普通的 Java 类一致。

3.5 数据交互

Json 作为一种数据交互格式，主要在本框架中处理数据交互。在前文中提到，在控制层中，将返回浏览器的数据以 Json 格式写入 Response。Json 与 XML 类似，都具有很好的可读性、扩展性，相当丰富的编码解码工具，但 Json 可以存储 JavaScrpit 复合对象，在企业级应用中，依托 Extjs 实现表示层，Json 串具有相当大的优势。

4 框架优势与不足分析

4.1 框架的优点

(1) 轻量级。无论是 Beehive，还是 Json 理念都是摒弃复杂的技术而实现，“从轻处理”业务逻辑的实现，一方面使开发者比较容易上手，另一方面服务器负载也不高。

(下转第 310 页)

参考文献(References)

[1] 郭佳荣.校园一卡通系统的现状与发展趋势[J].河北省科学院学报,2009,26(2):46-48.

[2] 郭玉娇,李志华,罗士波.一卡通的应用现状及前景[J].化工高等教育,2009(1):96-99.

[3] 高进.从一卡到无卡的高校校园一卡通金融支付展望——以东南大学为例[J].江西青年职业学院学报,2009,19(3):38-41.

[4] 修进玲,樊铁成.基于一卡通的校园电子商务研究[J].航海教育研究,2007(4):105-107.

[5] 曾晓莉,段凡丁,王龙业.基于校园网的网上支付系统设计[J].计算机时代,2006(10):35-37.

[6] 张玉键,张月琳.依托校园一卡通构建网上交易系统[J].中山大学学报:自然科学版,2009(3):108-111.

[7] 李斐,潘峰.基于网上支付模式安全性的研析[J].内江科技,2006,27(8):91-92.

[8] 王永全.网络交易安全性与可靠性技术探讨[J].电信科学,2006,22(8):60-62.

[9] 何新科.电子商务安全支付技术及其应用[J].无锡职业技术学院学报,2006,5(2):89-91.

(上接第 298 页)

(2) Web2.0。由于采用了 Extjs 作为用户的界面实现,用户操作的体验几乎可以和 CS 模式媲美,异步交互在框架中体现得淋漓尽致,浏览器和服务器的数据传输比传统的 Web1.0 更加简洁、高效。

(3) 扩展性强。框架中的 3 个组成部分可以用其他的框架或者模型代替。比如 Beehive 用 Struts、Spring 代替;Extjs 可以用 Jquery 等代替,不会影响总体 MVC 核心思想。

4.2 框架在实际运行中出现的问题

(1) 由于页面用 Extjs 实现,其在浏览器中生成了大量的 DOM 对象,其显示的效率和浏览器解析 JavaScrpit 的效率有关。实践中表明,CHROME 浏览器解析速度最快,其次是 FIREFOX。在即将退出的 FIREFOX4 中,已经采用了新的 JavaScrpit 解析器,试用后,感觉解析速度提高很多。

(2) 调试从 Java 转为 JavaScrpit 。大量的编程工作会集中在 Extjs 的 JavaScrpit 中。

(3) 不同于普通的 web 开发框架,本框架对网络要求较高,所以比较适用于基于局域网的应用系统,比如高校、企业的信息化建设。

(4) 框架中采用的 Extjs,特别适用于企业管理信息系统。对于只是门户展示等非管理性的网页,其功能优势就不是很明显。

5 框架应用

北京大学的电子校务建设中采用了此框架得到了良好的效果。比如,“北京大学学生综合信息管理系统”涉及学生、教师、教务部、实验室与设备部、总务部、研究生院和其他院系等多种用户,其操作页面采用 Extjs 实现(界面略)。业务操作的具体实现采用 Beehive。用户与后台的数据交互,比如发起操作请求及获得的数据响应利用 Json 承载。由于北京大学信息化建设比较早,早期的信息系统一般都采用 C/S 模式构建。业务部门在使用学生系统中,操作的简洁性、交互性、集成性都基本达到了 C/S 模式下的水平。

6 结束语

由于篇幅限制,无法详细举例说明框架的应用方法。在北京大学电子校务建设中,学生管理、人事管理等综合信息管理系统都已经采用了此框架。实践证明,Beehive + Extjs + Json 的开发框架对于开发 web2.0 下的企业级应用具有相当的优势,也获得了用户的积极肯定。

参考文献(References)

[1] Farrell J, Nezlek G S. Rich internet applications the next stage of application development[C]. Information Technology Interfaces, 2007:413-418.

[2] Mesbah A, van DEURSEN A. An architectural sytle for ajax[C]// Software Architecture, WICSA 07. The Working IEEE/IFIP Conference on Jan, 20007:9-19.

[3] Reenskaug, Trygve. The original MVC reports Trygve Reenskaug Home Page[EB//OL]. [2007-07]. http://heim.ifi.uio.no/~trygver/2007/MVC_Originals.pdf.

[4] Fowler, Martin. Patterns of Enterprise Application Architecture [EB/OL]. (2002) (s. l.): Addison-Wesley Professional. 978-0321127426.

[5] Alessandro Lacava. Speed Up Your AJAX-based Apps with JSON [EB/OL]. (2006-05).