View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Xiamen University Institutional Repository

基于 Flex 的企业级应用的实现

张思音，周文杉
（厦门大学软件学院，厦门 361005）

摘 要：系统采用了目前流行的 RIA (Rich Internet Applications) 富互联网应用作为企业级 Web 开发的新型的解决方案。采用更加友好、丰富的界面和强大的数据异步交互功能，大大提升了客户的网站体验。用网站本身留住客户，强化网站的粘性，在同等业务功能和服务的情况下，RIA 网站显然具有明显的优势。

关键词：Flex；RIA；胖客户端；丰富界面

The Realization of Enterprise Applications Based on Flex

ZHANG Siyin , ZHOU Wenshan
(Software school of Xiamen University , Xiamen 361005)

Abstract：The system uses the current popular RIA (Rich Internet Applications) rich Internet applications, as a new enterprise-level Web development solutions. Adopted a more friendly, rich interface, and powerful interactive features asynchronous data, greatly enhancing the customer's site experience, with the site itself to retain customers, strengthen the site's stickiness, business functions and services in the same situation, RIA website clearly has a distinct advantage.

Key words：Flex ; RIA ; Rich Client ; abundant interface

1 技术基础

本系统采用了当下流行的 RIA (Rich Internet Applications) 富互联网应用，作为企业级 Web 开发的新型的解决方案。采用更加友好、丰富的界面，和强大的数据异步交互功能，大大提升了客户的网站体验，用网站本身留住客户，强化网站的粘性，在同等业务功能和服务的情况下，RIA 网站显然具有明显的优势。Flex 是 RIA 的一种实现，也是当前比较流行的 RIA 技术。采用了 FSH (Flex + Spring + Hibernate) 代替以往的的轻量级 J2EE 解决方案 SSH，并在此基础上进行了调整和扩展。作为新型的企业级解决方案，Flex 的企业 Web 方案现在正在起步阶段，与当前成熟的 Spring + Hibernate 进行结合和扩展，在保证了企业系统稳定性和安全性的前提下，从客户角度进行了一次重大的变革。

2 设计思路

传统企业级 Web 应用流行采用的是 SSH (Struts + Spring + Hibernate) 轻量级开发框架，也在最近几年的发展中逐渐成熟和稳定，前台表现层采用 JSP 文档进行展示。这种传统的瘦客户模式在数据响应速度及简单文档展现方式等方面都不能满足 Web 浏览者更高的、全方位的体验要求了。

具体的设计思路是将表现层替换成胖客户端模式，采用了 Flex 技术——当前最火热的 RIA 实现框架，作为表现层的解决方案，而后台服务器端仍然采用了 Spring+Hibernate 框架组合，这样的整体企业 Web 框架称之为 FSH (Flex + Spring + Hibernate)。

FSH 框架主要需要解决的是 Flex 与 Spring 间的合作方式，Spring 和 Hibernate 的合作方式具体请参照 SSH 框架，这里就不再赘述了。采用了 BlazeDS 技术来连接 Flex 与 Spring 直接的联系，BlazeDS 是一个基于服务器的 Java 远程控制 (remoting) 和 Web 消息传递 (messaging) 技术。Flex 端采用 Remote Object 技术通过 BlazeDS 调用服务端 Spring 管理的 "Service Bean"，进而异步调用其业务方法，通过回调函数返回数据并展示在页面上。而 Flex 端采用了 Fabrication 框架，Fabrication 是 PureMVC Framework 一个加强版本，对 PureMVC 中比较繁琐的开发部分简化，有效地将逻辑代码与界面设计分离，并提供了相应功能对开发带来了极大便利。这样使得开发团队中的开发人员能够专注于自己的职责，使得美工人员和开发人员能完全地分离开来，进一步提高了项目的开发效率。

总的来说，这种新型的企业级 Web 开发框架 FSH 的设计思路是在前者的开发之上采用新型技术进行组装，并给客户带来了全新的 Web 体验。新技术的加入和框架的重新设计组合不仅加快了项目的开发进程，而且带来了更好项目质量。

3 实现过程

3.1 界面

采用 Flex 开发，界面是一个最大的优势，因此下面将通过较多的截图来展现 Flex 强大的界面表现能力。

首先介绍普通用户操作功能界面，利用 Flex 的各种控件实现的物流管理系统的交接单页面。Flex 不在是基于页面的系统，它是基于窗体，容易做出具有丰富交互功能的应用程序，如图 1 所示。

用户可以在直接进入主页，不登录的情况下使用本运费查询功能，查询相关运费价格，以确定是否需要登录系统下交接单，如图 2 所示。

使用 Flex 技术，可以实现动画、短片的在线播放，在展

收稿日期：2010-04-20

© 1994-2014 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

现工作流程、企业理念方面具有突出优势。如图 3 所示。

图 1 下订单界面

图 2 运费查询

图 3 寄件流程短片播放

其次介绍管理员的操作功能界面，运用 Flex 控件，可将左侧权限动态地拖拉至右侧，实现权限增加的功能，这也是其友好性的体现，如图 4 所示。

图 4 角色权限管理

3.2 业务逻辑的实现

在 Flex 处理部分业务逻辑。Flex 是 RIA 的一种实现，界面被分解成许多独立的模块，这些模块都会对收到的信息做出相应的反应，有些模块会和服务器端进行交互，有些则只需要在客户端进行处理。其通过在客户机上单独运行一个客户端应用程序（Flex 以 swf 格式在 Flash player 中运行），能够处理界面的部分消息，提高了用户的体验和响应速度。

由于客户端有一个独立于服务器的应用程序，服务器端主要是通过异步的数据传输方式来负责数据的保存和更新，同时，一些复杂的逻辑也将在这里实现。

4 源代码

4.1 前台业务逻辑处理

脚本类 orderMediator 主要是负责响应下订单页面，数据合法性检查、物品天添加窗体的创建和销毁、数据的缓存以及最后将数据传递给服务器端处理。其核心代码如下：

```
/**
 * reactToAddCargoBtnClick () 方法
 * 按下" 添加货物" 按钮产生的操作
 * @author 张思音
 * @version 2.0, 2009/07/20
 */
public function reactToAddCargoBtnClick ( event: Event) :void {
add_cargos_win = AddCargos (
PopUpManager.createPopUp ( viewComponent as Order,AddCargos,true)) ;
add_cargos_win.cargoTypeCom.dataProvider = types;
PopUpManager.centerPopUp (add_cargos_win) ;
add_cargos_win.title = " 添加货物" ;
add_cargos_win.addEventListener ( " addCargo_click" ,add_update) ;
}
/**
 * add_update () 方法
 * 添加货物完，更新列表
 */
public function add_update (event:Event) :void {
var cargo:CargoInputVO = new CargoInputVO () ;
add_cargos_win.removeEventListener ( " addCargo_click" ,add_update) ;
var cargoVolume:Number = add_cargos_win.cargoNoNum.value
* add_cargos_win.cargoLengthNum.value * add_cargos_win.cargoWidthNum.value
* add_cargos_win.cargoHeightNum.value;
cargo.cargoname = add_cargos_win.cargoNameTxt.text;
cargo.cargotype = add_cargos_win.cargoTypeCom.selectedItem.ctypename;
cargo.id = add_cargos_win.cargoTypeCom.selectedItem.id;
cargo.num = add_cargos_win.cargoNoNum.value;
cargo.weight = cargo.num * add_cargos_win.cargoWeightNum.value;
Alert.show (cargoVolume+"") ;
if (cargoVolume < 0.0001) {
cargo.volume = 0.0001;
```


© 1994-2014 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
                }
                else {
                cargo.volume = Number (volumeFormatter.for-
mat (cargoVolume)) ;
                }
                cargos.addItem (cargo) ;
                totalV += cargo.volume;
                totalW += cargo.weight;
                setPrice () ;
            }
            /**
             * check () 方法
             * 验证订单信息是否合法
             */
             public function check () :Boolean {
            var timeToSend:Date = sendTimeDate.selectedDate;
            if (snVal.validate () .type == ValidationResultEvent.
INVALID) {
                sendNameTxt.setFocus () ;
                return false;
                } else if (rnVal.validate () .type == Valida-
tionResultEvent.INVALID) {
                    receiveNameTxt.setFocus () ;
                    return false;
                } else
                    return true;
            }
        /**
        * 获取路线 ID
        */
        public function setRouteId (event:ResultEvent) :void {
        var routeIdList:ArrayCollection = new ArrayCollection () ;
            routeIdList = event.result as ArrayCollection;
            routeId = routeIdList.getItemAt (0) as Number;
            rService.removeEventListener (ResultEvent.RESULT,
setRouteId) ;
            if (0 == routeId) {
            routeErrormsg.text = " 对不起，不存在该路线！" ;
                }
            else {
                routeErrormsg.text = "" ;
                pService.addEventListener ( ResultEvent.RE-
SULT,setDprice) ;
                pService.getScopePriceByOrder ( receiveArea-
Com.selectedItem.id, totalV, totalW) ;
            }
        }
```

4.2 与服务器进行数据交互

Flex 与 Java 端间的数据传输可以采用自定义的类的形式如图 5 所示，VO 就是自己定义的类，其 Java 端的数据类型必须和 flex 端一一对应，以下是一个实例：

Flex 端的一个 InputVO：

```
package com.xmetc.xmu.coolit.lms.client.inputvo
{
 import mx.collections.ArrayCollection;
 [ RemoteClass ( alias = " com.xmetc.xmu.coolit.lms.server.
inputvo.ActorInputVO")]
```

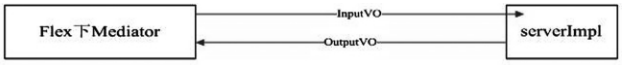

图 5 Flex 与 Java 通信

```
 [Bindable]
 /**
  * ActorInputVO 类：
  * 客户端封装角色各属性的类
  * @author 周文杉
  * @version 2.0, 2009/07/17
  */
 public class ActorInputVO
 {
          public var id:Number;
          public var code:Number;
          public var actorname:String;
             public var Authorizes:ArrayCollection;
 }
}
```

Java 端必须有个与其一一对应的 VO：

```
package com.xmetc.xmu.coolit.lms.server.inputvo;
/**
 * ActorInputVO 类：
* 一个封装 pojo 包中角色各属性的类
* @author 周文杉
 * @version 2.0, 2009/07/17
 */
public class ActorInputVO {
 private Long id;
 private long code;
 private String actorname;
 private List <AuthorizeInputVO > authorizes = new Ar-
rayList<AuthorizeInputVO> () ;
  省略 set 和 get 方法
 …
}
```

5 结语

物流系统的逻辑相对复杂，因此前期的设计非常重要，只有理清整个系统的流程，Flex 才能真正开始发挥作用。同时，熟悉 Flex 常用控件后，开发效率会大大提高。

参考文献

[1] 杨占坡，杨铭，翁颖. Flex 3 RIA 开发详解与精深实践. 北京：清华大学出版社，2009.

[2] Craig Larman ：UML 和模式应用. 北京：机械工业出版社，2006.

[3] Cay S. Horstmann Gary Cornell. Java 核心技术卷 2：高级特性. 北京：机械工业出版社，2009.

作者简介

张思音，女，本科在读，研究方向：软件工程。

周文杉，男，本科在读，研究方向：软件工程。

© 1994-2014 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net