文章编号:1000-5013(2013)06-0640-05 **doi**:10.11830/ISSN.1000-5013.2013.06.0640

结合 HTML5 和 JSF2 技术定制复合组件

曾伟升,余金山

(华侨大学 计算机科学与计算学院,福建 厦门 361021)

摘要: 提出一种通过引入 JSF2 框架支持,利用 JSF2 的新技术与 HTML5 相结合开发复合组件的方法.以一组具有拖放功能的 drag 和 drop 复合组件的实现为例,说明 JSF2 和 HTML5 技术相结合的工作原理和实现机制.通过所提供的定制复合组件方法,使开发者可以高效地利用 JSF2 和 HTML5,实现易于修改和扩展的用户界面组件;开发人员根据自己需要的功能可以自行构建 JSF2 组件,定制可复用的 HTML5 表现组件,提高 Web 应用程序的开发效率和质量.

关键词: JSF2; HTML5; 复合组件; Web

中图分类号: TP 311 **文献标志码**: A

超文本标记语言(hypertext markup language,HTML)的最新版本 HTML5 首次提出将桌面应用程序的强大功能,通过拖放、画布、视频和音频等设施引入到浏览器中,并增加了一些反映 Web 应用的新元素、事件属性和更多的 API,使 Web 的实现更丰富,更具灵活性和互动性[1].但是,目前浏览器对 HTML5 的兼容性并不统一.JSF2(java server faces 2)是为了解决开发 Web 应用程序客户端难度大、效率低的问题而由开放的国际组织 JCP(java community process)[2]提出的一项开发框架.JSF2 完全采用 Facelet 模板技术[3]、Managed Beans 技术、混合组件[4]技术、加入了内置的 Ajax 支持[5-6],还提供丰富的底层构建组件和 UI 组件等有用的新特性.其缺点是还没有很好的集成开发环境支持其可视化.本文提出结合 HTML5 和 JSF2 技术定制复合组件的一种方法,引入 JSF2 框架支持的 Facelets 模板技术,利用 JSF2 混合组件技术定义复合组件的接口[7-8].

1 整体设计

以具有拖放功能的 JSF2 复合组件〈h5:drag〉和〈h5:drop〉的实现为例,说明 JSF2 和 HTML5 的复合组件构建机制及其实现.

图 1 为 drag 复合组件的整体结构.整个组件包括 HTML5 支持部分和 JSF2 支持部分.HTML5 支持部分主要负责基于事件的拖放机制实现,包括 drag 拖拉源和 drop 放置源实现;而 JSF2 支持部分则将拖放组件封装成 JSF2 组件,具体包括模板视图,资源托管和内置 Ajax 支持等.

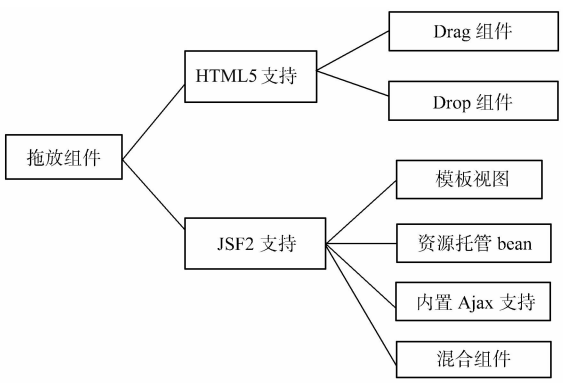

图 1 drag 复合组件整体结构

Fig. 1 Whole structure of the drag's composite component

2 组件的定义与实现

2.1 使用 JSF2 对组件的支持

引入 JSF2 框架,为复合组件提供所需的基础架构支持,其配置代码如下:

〈servlet〉

收稿日期: 2012-05-12

通信作者: 余金山(1952-),男,教授,主要从事软件工程与网络计算和人工智能的研究.E-mail:yjs@hqu.edu.cn.

```
    〈servlet-name〉Faces Servlet〈/servlet-name〉
    〈servlet-class〉javax. faces. webapp. FacesServlet〈/servlet-class〉
    〈load-on-startup〉1〈/load-on-startup〉
  〈/servlet〉
  〈servlet-mapping〉
    〈servlet-name〉Faces Servlet〈/servlet-name〉
    〈url-pattern〉/faces/ * 〈/url-pattern〉
    〈url-pattern〉 * . faces〈/url-pattern〉
  〈/servlet-mapping〉
```

JSF2 综合了 Facelets 模板、资源处理和一个简单的命名约定来实现复合组件. 资源处理是通过打包机制来得到的. 资源通常被打包放置在 Web 应用程序的根目录下或 classpath 下，所以需创建一个 resource 资源库，用于消除对配置的需要，消除 Tag 标签文件.

下面在资源库中创建 Facelets 标记文件，用于定义复合组件接口和属性[9-10].

2.2　定义组件

引入 facelets 组件包，其代码如下：

```
〈ui:composition xmlns="http://www. w3. org/1999/xhtml"
    xmlns:ui=http://java. sun. com/jsf/facelets
```

下面使用组件技术定义组件的接口和属性，定义 drag 组件的接口和属性代码如下：

```
〈composite:interface〉//定义 drag 接口
    〈composite:attribute name="ondragstart"/〉//拖放开始的事件属性
〈/composite:interface〉
```

其中 ondragstart 是 drag 组件的属性，它在拖放开始时将被拖放元素的 textContext 传递给 dataTransfer 对象，用于后面的处理.

定义 drop 组件的接口和属性代码如下：

```
〈composite:interface〉//定义 drop 接口
    〈composite:attribute name="ondragenter"/〉//拖放元素进入目标元素的事件属性
    〈composite:attribute name="ondragover"/〉 //拖放元素在目标元素上移动的事件属性
    〈composite:attribute name="ondragleave"/〉//完成拖放的事件属性
    〈composite:attribute name="ondrop"/〉 //拖放过程的提示事件属性
    〈composite:attribute name="render"/〉//通知放置事件属性
    〈composite:attribute name="payload"/〉 //拖放过程的负载事件属性
    〈composite:attribute name="payloadType"/〉//负载类型事件属性
    〈composite:insertChildren /〉
〈/composite:interface〉
```

其中：ondragenter，ondragover，ondragleave 分别表示拖放过程中被拖放的元素开始进入、正在进入和离开目标元素范围的事件；ondrop 用于拖放过程的提示；render 用于通知放置；payload 为拖放过程的负载；payloadType 为负载类型.

2.3　组件的实现

设计 drag. xhtml，drop. xhtm 和 JavaScript 文件(drop. js)来实现〈h5:drag〉和〈h5:drop〉组件.

首先引入 HTML5 文档类型/名称空间 DOCTYPE，其代码为〈! DOCTYPE html〉，以便于组件对 XHTML 语法和 HTML5 新的标签和新的特性的引用.

拖拉源 〈h5:drag〉 组件的实现代码如下：

```
〈composite:implementation〉
    〈div draggable="true"//设置是否允许施放属性
        //通过 JSF2 表达式语言访问组件的界面内定义的属性
```

```
        ondragstart="#{cc.attrs.ondragstart}">
        <composite:insertChildren />
    </div>
    </composite:implementation>
```

〈h5:drag〉组件要创建一个可拖拉的DIV,首先需要对可拖曳的节点进行声明,给拖曳的节点添加draggable属性并设值为true.ondragstart是创建者在交互界面中指定的值,通过cc.attrs来引用接口中定义的属性.最后使用〈composite:insertChildren〉将任意标记插入〈h5:drag〉标签体中.

放置目标〈h5:drop〉组件的实现代码如下:

```
<composite:implementation>
    //导入组件相应的JavaScript
    <h:outputScript library="html5" name="drop.js" target="head" />
    <div id="#{cc.id}" ondragenter="#{cc.attrs.ondragenter}"
        ondrop="#{cc.attrs.ondrop}"
        ondragover="#{cc.attrs.ondragover}"
        ondragleave="#{cc.attrs.ondragleave}">
    </div>
    <script> html5.jsf.init("#{cc.id}",//组件的JavaScript方法
        "#{cc.attrs.payloadType}",
        "#{cc.attrs.render}"); </script>
    </composite:implementation>
```

〈h5:drop〉组件像〈h5:drag〉一样创建一个DIV,然后将〈h5:drop〉标签体中的标记插入此DIV中.当创建了〈h5:drop〉组件后,调用html5.jsf.init()开始执行.该函数通过添加drag-enter和drag-over事件处理程序初始化组件的DIV,这些处理函数在〈h5:drop.js〉中实现.〈h5:drop.js〉的代码如下:

```
if(! html5)
var html5 = {}
if (! html5.jsf) {
    html5.jsf = {
        init : function(ccid, payloadType, renderIds) {//组件的JavaScript方法
        ⋮
        dropzone.addEventListener("drop", function(event) {//拖放组件监听事件
          event.preventDefault(); //取消浏览器的默认响应
        jsf.ajax.request(dropzone.payloadInput, event, {//JSF2内置Ajax处理方法
          html5.jsf.init(ccid, payloadType, renderIds);
        ⋮
        }, false);}
```

〈h5:drop.js〉中的JavaScript阻止浏览器对drag-enter和drag-over事件进行默认响应,阻止浏览器拒绝放置,使其无条件接受所有放置.

2.4 资源托管Bean

组件通过JSF2传递拖放负载到DragDrop.setPayload().基于该负载,DragDrop.setPayload()方法创建一个新的项,并将其添加到保存的程序中.这里还为负载属性包含一个getter方法,因为JSF2需要setter和getter来处理输入值.其实现代码为

```
@Named
@SessionScoped
public class DragDrop implements Serializable {
    @Inject private RSSFeed rssFeed;
```

```
    public DragDrop() {
    }
  //负载的 get 和 set 方法
    public String getPayload() {
    ⋮
    }
    public void setPayload(String payload) {
    ⋮
    }
```

代码中的@Named 注释与@SessionScoped 一起实例化托管 Bean；@Inject 注释提供对程序托管的 Bean 的访问. JSF2 应用程序通常需要在 Java 代码中访问这些托管的 Bean. 这里可以使用 JSF2 和 JSP Expression Language API 来访问那些托管的 Bean，或者仅使用@Inject 注释也可.

2.5　添加 JSF2 内置 Ajax 并发送一个负载到服务器

在 JSF2 应用程序中，用户操作的数据大多数存储在服务器上. 在这里，〈h5：drag〉和〈h5：drop〉组件支持附加一些数据（负载）到一个拖放指令中. 页面创建者指定一个 Bean 属性充当这个负载，其代码为

```
〈h5:drop id="dropzone" payload="#{dragDrop.payload}"
          render="@this"
    ondragover="dragover(event)"…〉
〈/h5:drop〉
```

在代码中将拖放负载连接到一个 Bean 属性：#{dragDrop. payload}，并且通知放置目标显示@this 即放置目标本身. 把一个不可见输入文本添加到放置目标，当生成一个放置时，在 Ajax 请求之前设置不可见输入的值，然后发起一个 Ajax 调用. 在 Ajax 调用过程中，JSF2 使用 jsf. ajax. request() 函数，并调用负载 Bean 属性的 setter 方法.

在〈h5：drop〉组件中，作为不可见文本输入的值的代码实现如下：

```
〈h:inputText id="payload" value="#{cc.attrs.payload}" style="display: none"/〉
```

不可见文本输入被指定为 Ajax 请求的源：

```
jsf.ajax.request(dropzone.payloadInput, event, {…});
```

由于不可见文本输入被指定为 Ajax 请求的源，JSF2 在服务器端处理输入，即它将调用与属性 setter 方法相关的文本输入 DragDrop. setPayload().

2.6　使用拖拉源和放置目标

〈h5：drag〉和〈h5：drop〉分别代表 HTML5 拖拉源和放置目标，要使用该复合组件首先需要声明一个名称空间并使用标记. 其代码为

```
xmlns:h5="http://java.sun.com/jsf/composite/html5"
```

这样使用复合组件的功能就很简单，就像〈h：inputText〉在服务器端与客户端传送数据的功能一样. 在应用程序中，可以像这样使用拖拉源：

```
〈h5:drag ondragstart="dragStart(event)"〉
〈/h5:drag〉
```

当一个拖拉启动时，指定一个 JSF2 调用的 JavaScript 函数，浏览器传递一个事件到此 JavaScript 函数，事件目标是〈h5：drag〉组件. 该函数从〈h5：drag〉元素深入到 anchor 元素，获取链接的文本和引用. 然后，将该信息嵌入到一个与 text 数据传输类型相关的字符串中. 则当一个放置被接受时，HTML5 拖放系统将此字符串转换成放置目标.

使用放置目标如下：

```
〈h5:drop id="dropzone" payload="#{dragDrop.payload}" render="@this"
          ondragover="dragover(event)"
          ondragenter="dragenter(event)"
```

```
ondragleave="dragleave(event)"
ondrop="drop(event)"></h5:drop>
```

与拖拉源一样,页面创建者可以将一些组件或 HTML 元素放在〈h5:drop〉组件中.页面创建者也可以指定一个 Bean 属性作为拖放负载,且当在一个放置之后执行的 Ajax 调用从服务器返回时,指定 JSF2 显示哪个组件.页面创建者也可以选择使用 JavaScript 代码对拖放指令进行干预.

3 结束语

文中给出如何使用封装了 HTML5 拖放功能的 JSF2 实现复合组件的一种方法.复合组件是对 JSF2 开发人员工具箱功能的增强,这些组件能重复利用,而不需要编写 Java 代码或者执行任何配置.文中还说明了现有组件的重用,如 JSF2 的内置〈h:inputText〉元素,减少在实现自己的复合组件时的工作量.当开发出一些复合组件之后,就可以将它们组合在一个 JAR 文件中,在开发其他项目的时候只需将这些组件打包成 JAR 文件即可直接引用.

参考文献:

[1] 唐灿.下一代 Web 界面前端技术综述[J].重庆工商大学学报:自然科学版,2009,26(4):350-356.

[2] GEARY D. JSF2 简介,流线化 Web 应用程序开发[EB/OL].[2009-6-15]. http://www.ibm.com/developerworks/cn/java/j-jsf2fu1/index.html.

[3] 彭邦伦.构建 Facelets 的 Web 应用程序[J].电脑编程技巧与维护,2011,38(1):55-58.

[4] 余金山.利用 XML, Tamino 和 CORBA 的软构件管理与检索技术[J].华侨大学学报:自然科学版,2008,29(4):518-521.

[5] 彭胜,刘卫国.基于 JSF 的 Ajax 组件开发与应用[D].长沙:中南大学,2009:11-16.

[6] 刘红坤.基于 Ajax 和 PHP 数据分页的实现[J]. 计算机系统应用,2012,21(2):218-220.

[7] 余金山,刘志伟.一种基于内置合约检查和可配置接口的软构件测试技术[J]. 计算机应用研究,2011,28(5):1756-1760.

[8] 毛澄映,卢炎生.构件软件测试技术研究进展[J].计算机研究与发展,2006,43(8):1375-1378.

[9] 余金山,周武斌.MDA 模型转换的 OCL 扩展[J].小型微型计算机系统,2012,33(3):548-551.

[10] 陈杜英,刘韶涛.面向复用的软构件信息系统的设计与实现[J].华侨大学学报:自然科学版,2012,33(3):270-274.

Combining HTML5 and JSF2 Technology to Develop Composite Components

ZENG Wei-sheng, YU Jin-shan

(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

Abstract: A method to make customized composite components through HTML5 and JSF2 technology, including the framework supporting by JSF2 was proposed. The working principle and the realization mechanism of composite components based on HTML5 and JSF2 are detailed analyzed, through the example that implements a group composite components with drag and drop function of drag and drop. By the method of making customized composite components, developers can use the JSF2 and HTML5 efficiently, realize the user's interface components easy to modify and expand, and use it to make customized components of JSF2 on their own needs, realize the reusable presentation components of HTML5, hence improve the development efficiency and quality of WEB applications.

Keywords: JSF2; HTML5; composite component; Web

(**责任编辑**:黄晓楠 **英文审校**:吴逢铁)