

# 基于拦截器机制的编辑器上传附件的自动清理<sup>①</sup>

陈 洁

(中华女子学院 计算机系, 北京 100101)

**摘 要:** 在线文本编辑器在 web 项目中得到广泛应用, 针对从网页文档中删除图片、音视频等对象时如何有效清除服务器上相应附件的问题, 提出一种在 Java EE 平台下使用的基于过滤器或拦截器机制的通用方案, 在每一次文档操作完成之后, 由系统自动清除服务器上无用的磁盘文件; 以 FCKeditor 编辑器的应用为例, 介绍了实现技术和软件实现方法. 附件的清理对用户是透明的, 且代码易于实现和重用, 能够有效而安全地应用于各类涉及内容发布的 web 系统中.

**关键词:** 在线文本编辑器; FCKeditor; 过滤器; 拦截器; 软件复用

## Interceptor Mechanism Based the Automatic Cleaning of Files Uploaded By the Editor

CHEN Jie

(Department of Computer, China Women's University, Beijing 100101, China)

**Abstract:** The text editor for internet has been widely used in web projects. In view of the problem how to clean up the corresponding disk files in the server when deleted the objects of the image, audio and video from the content in the editor, a general-purpose solution application on Java EE platform was put forward. Based on filters or interceptor mechanism, it could automatically clean up the attachments loaded to the server but of no use whenever the edit was over. Taking the FCKeditor application for example, the software implementation technology and method was introduced in detail. The cleaning was transparent to user. The solution is safe and reusable, and also easy to implement. It can effectively apply in various types of web projects related to content publishing.

**Key words:** the text editor for internet; FCKeditor; filter; interceptor; software reuse

在线文本编辑器, 也称为富文本编辑器, 是一种基于 Web 的可视化 HTML 编辑器, 在网页上对文档进行编辑和排版后可保存为 HTML 格式的文本, 在内容管理、博客、论坛、新闻发布、网络教学平台等与内容发布相关的 web 系统中有着广泛应用. 如, CKSource 公司开发的 FCKeditor(从 3.0 版本开始更名为 Ckeditor)就是一个功能强大的开源在线文本编辑器, 可以实现类似微软 Word 桌面文本编辑器的许多强大功能, 除编辑文本内容外, 还支持文件上传功能, 可以方便地在文档中链接其他文件、插入图片、Flash、音视频等附件对象, 在国内外许多 Web 项目或大型网站中被采用<sup>[1-3]</sup>.

但是, 随着网站使用量的增加, 以及对已有内容

的修改和删除, 特别是对文档中附件对象的更新操作, 如上传新的附件以替换原有附件或是取消链接、删除图片等, 从文档中删除或取消附件对象并不会从服务器上清除对应的磁盘文件, 由此导致服务器上留存很多“垃圾”文件.

FCKeditor 自带的文件管理器组件没有删除文件的功能, 其他基于 Web 的文件管理器, 如可以作为插件集成到 FCKeditor 或 Ckeditor 中的 KFM、corefive、phpletter 等文件管理器<sup>[4-6]</sup>, 以及 CKSource 的文件管理器 CKFinder<sup>[7]</sup>, 虽然提供了浏览服务器并删除文件的功能, 但是, 出于安全性考虑, Web 程序大多不对用户开放浏览服务器的功能, 用户的手动操作可能会删除有用的附件.

<sup>①</sup> 基金项目: 中华女子学院 2013 年度立项项目(JS2013-02009)

收稿时间: 2013-07-18; 收到修改稿时间: 2013-08-19

针对上传附件的清理问题,本文以 FCKeditor 编辑器在 Java 平台下的 web 应用为例,提出一种基于过滤器或拦截器机制自动清理附件的通用方法,可以在每一次添加、修改、删除或取消文档操作完成之后,及时清除服务器上已上传的但未使用的附件。

## 1 设计思想

### 1.1 “垃圾”附件的产生

以下几种操作会导致在服务器上产生“垃圾”附件。

① 在文档编辑过程中,更换或删除了文档中已有的附件对象。

② 在文档编辑过程中,上传了附件,但最终并未保存文档。

③ 因业务需要,删除已有文档(如,从数据库中删除了某个文档记录),但文档中使用的附件仍留存在服务器上。

### 1.2 在文档的每一次操作完成后清除附件

如果对文档的每一次添加、修改、删除或取消操作完成后,系统都能及时清除不需要的附件,使得服务器上的附件数量在文档操作前后始终保持一致状态,即文档操作前的附件数量加上操作过程中添加或删除的附件数量等于操作结束后的附件数量,即可保证服务器上不会留存无用的附件。

通过记录用户在每一次编辑过程中上传的所有附件信息,更新文档前的附件信息,保存文档时确认使用的附件信息,以及被删除文档中使用的附件信息,并对比操作前后的附件信息,可以有效而便捷地完成附件的清理。

针对 java EE 平台上的应用,FCKeditor 使用 ConnectorServlet 类(调用 dispatcher.doPost 方法)实现文件上传功能<sup>[8]</sup>,通过这个 servlet 类或被其调用的 dispatcher 类可以捕获用户在文档编辑过程中上传的所有附件信息。

使用文本编辑器进行业务操作时,如新闻发布系统中的新闻发布、修改和删除等,可以将操作请求提交给 servlet 或 struts2 框架中的 action,在 servlet 或 action 中调用业务层,再由业务层调用 DAO 层的数据库操作类完成文档在数据库中的添加、修改和删除操作,写入文档的附件信息可以利用正则表达式提取出来。

附件管理是每一个使用在线文本编辑器进行业务操作且需要上传附件的 web 应用都需要的功能,利用

Java EE 技术中的过滤器或 struts2 框架下的拦截器机制,通过相关配置建立附件管理与业务逻辑的联系,可以方便地实现附件管理,且代码具有良好的可移植性和可复用性。

## 2 实现技术

### 2.1 使用过滤器截取上传的附件信息

FCKeditor 自带的文件管理器组件通过 ConnectorServlet 类实现文件上传功能,为避免修改原有组件,可以为 ConnectorServlet 设置一个过滤器(Filter),在附件上传成功之后,截取上传的附件信息。

过滤器的基本功能是对 Servlet 容器调用 Servlet 的过程进行拦截,截取发送给目标资源的请求或/和来自目标资源的响应,并在 doFilter 方法中完成过滤任务<sup>[9]</sup>。因此,使用过滤器可以查看、提取或操作在客户机和服务器之间交换的数据。

过滤器在 web.xml 文件中进行注册并设置它所能拦截的目标资源,添加和删除过滤器不需要改动任何应用程序代码。web 应用程序启动时,web 服务器会根据 web.xml 文件中的配置信息自动创建 Filter 实例对象,当客户机请求目标资源时,容器就会调用与这个目标资源相关联的过滤器的 doFilter 方法。

### 2.2 使用拦截器截取文档内容及其附件信息

在 Java EE 平台下,使用 struts2 框架构建 web 程序时,业务请求是提交给 action 处理的,因此,可以对 action 设置拦截器(Interceptor),在添加、修改、删除等相关业务逻辑完成之后截取文档内容,并提取附件信息。

与过滤器一样,struts 框架的拦截器也是动态配置的,只需要在 struts.xml 文件中注册自定义的拦截器实现类,并在相关 action 的配置中引用拦截器,当 struts 框架工作时,拦截器即可生效。

拦截器可以动态地拦截发送到指定 Action 的请求,通过拦截器机制,开发者可以在 Action 执行的前后插入某些代码,通过这种方式,就可以把多个 action 中需要重复执行的动作放在拦截器中定义,从而提供更好的代码复用性<sup>[10]</sup>。此外,拦截器中的方法会在目标方法调用之前或之后由系统“自动”执行。

### 2.3 使用正则表达式提取文档中的附件信息

在线文本编辑器产生的文档以 XHTML 格式存储,所有的附件对象都是用标准的 HTML 标记标识,如,链接对象用 <a> 标记标识,href 属性值指定资源路径及

名称; 图片对象用<img>标记标识, flash 与音视频对象用<embed>标记标识, src 属性值指定资源路径及名称. FCKeditor 编辑器界面及其生成的 HTML 代码如图 1 和图 2 所示.



图 1 FCKeditor 编辑器

```
<a target="_blank" href="/forum/userfiles/file/ref.txt">链接文本</a><br />
插入图片
```

图 2 编辑器生成的 HTML 代码

正则表达式是字符串处理的利器, 利用正则表达式可以从保存为 HTML 格式的文档中提取所有附件信息.

### 3 软件实现

为实现附件的自动清理功能, 需要在 web 程序中定义以下几个对象.

#### 3.1 定义 FileManager 类

该类用于管理附件, 有两个静态方法:

① findFile 方法, 使用正则表达式提取文档中的所有附件信息, 方法声明为:

```
public static List<String> findFile(String content)
```

所使用的正则表达式:

```
<(?:embed|img|a).*?(?:src|href)="(.+?)".*?>
```

“(+)”括号中匹配的内容就是文档中插入的附件信息.

② delFile 方法, 删除文档中的附件, 方法声明为:

```
public static void delFile(List<String> fileList1,
List<String> fileList2)
```

该方法有 2 个 List 类型的参数, 存放附件信息. 对于添加操作, fileList1 中存放本次编辑操作中上传的所有附件信息, fileList2 中存放最终在文档中用到的所有附件信息. 对于修改操作, fileList1 中存放修改前文档中的附件信息及修改过程中上传的附件信

息, fileList2 中存放修改后的文档中用到的所有附件信息. 对于删除操作, fileList1 中存放被删除文档中的所有附件信息, fileList2 为空. delFile 的算法如图 3 所示.

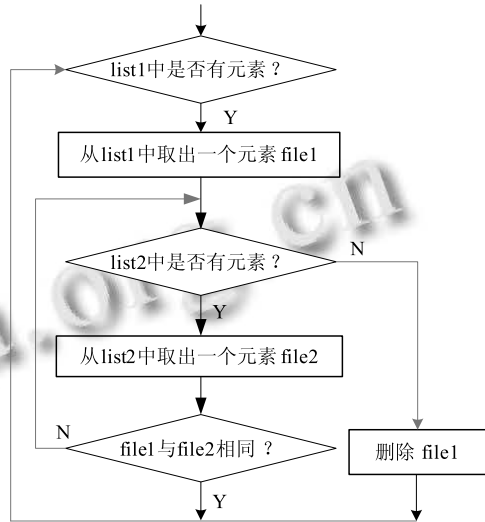


图 3 delFile 方法的实现流程

#### 3.2 定义过滤器类 FileFilter

该类用于截取上传成功的附件信息. 过滤器的目标资源为 ConnectorServlet, 该 servlet 实现了文件上传功能, 并生成一段 javascript 文本作为响应返回客户端, 以下是上传成功后生成的 javascript 文本:

```
<script type="text/javascript">
// 省略其他代码
window.parent.OnUploadCompleted(0,
'/forum/userfiles/image/enter.png');
</script>
```

其中的 OnUploadCompleted 方法包含若干参数, 第 1 个参数为状态码: 0-表示上传成功且无任何警告信息; 201 表示上传成功, 但有文件重命名的警告; 其他值均表示有错误, 上传失败. 第 2 个参数为上传的附件在服务器上的存储路径及文件名. 同样可以使用正则表达式从上述文本中提取状态码和上传的附件信息, 从而过滤出所有上传成功的附件信息.

FileFilter 过滤器的功能是截取上传成功的附件信息, 因此过滤器需要拦截来自 ConnectorServlet 的响应, 在过滤器的 doFilter 方法中利用自定义的响应包装器可以获取目标资源的响应结果<sup>[11]</sup>, doFilter 方法的实现流程如图 4 所示.

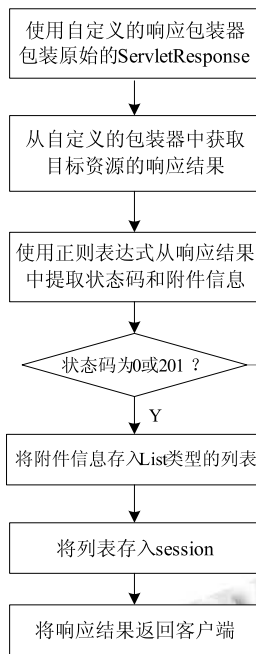


图 4 doFilter 方法的实现流程

### 3.3 定义 3 个拦截器类

为了管理已上传到服务器上的但在文档中未使用的附件, 针对用户的添加、修改、删除和取消操作, 需要定义以下 3 个拦截器, 处理文档中的附件信息。

① OkInterceptor 拦截器在完成文档的添加、修改和删除等业务逻辑(即, 将文档内容存入数据库或从数据库中删除)之后, 处理不再使用的附件。

② PreUpdateInterceptor 拦截器在完成查找文档的业务逻辑之后, 获取文档中的附件信息。

③ CancelInterceptor 拦截器 在用户放弃已编辑的文档跳转到其他页面或直接关闭浏览器窗口时, 处理编辑过程中上传的所有附件。

对于用户的取消操作, 必须在页面的 onunload 事件中提交处理请求。由于 web 程序只需要完成附件的清理工作, 因此, CancelInterceptor 拦截器所拦截的目标 action 本身不执行任何操作, 其返回结果也无需映射到任何物理视图, 所以这个 action 是以 ajax 方式请求的, 并在配置文件中将结果类型指定为“json”。

与过滤器类似, 拦截器可以拦截传给 action 的请求或从 action 返回的处理结果。OkInterceptor 和 PreUpdateInterceptor 都是要拦截相应 action 的处理结果, 即在添加、修改、删除和查找等业务逻辑执行成功后再处理文档中的附件。

以新闻发布系统为例, 当用户执行修改操作时, 管理过程如图 5 所示。

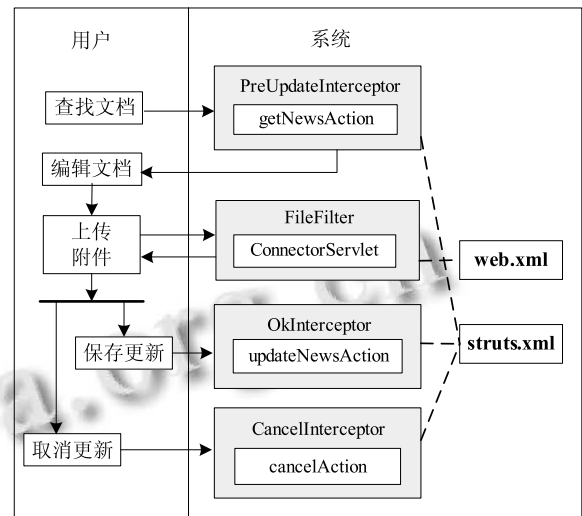


图 5 用户执行修改操作时的系统管理过程

对应的 struts.xml 配置如下:

```

<!-- 配置自定义的 3 个拦截器 -->
<!-- 配置 3 个拦截器栈 stackForOk, stackForPre, stackForCancel, 分别引用 3 个拦截器 okInterceptor, preUpdateInterceptor, cancelInterceptor -->
<!-- 配置 addNewsAction、updateNewsAction、delNewsAction, 并引用 stackForOk -->
<action name="*NewsAction" class="news.action.NewsAction" method="{1}News">
    <result>...</result>
    <interceptor-ref name="stackForOk"/>
</action>
<!-- 配置 getNewsAction, 并引用 stackForPre -->
<!-- 配置 cancelAction, 结果类型为“json”, 并引用 stackForCancel -->
<action name="cancelAction">
    <result type="json" />
    <interceptor-ref name="stackForCancel" />
</action>
    
```

如果在同一个 web 应用中, 有多个业务逻辑需要使用文本编辑器, 例如, 网络教学平台中的课件制作、课程论坛、在线答疑等模块都要用到文本编辑器, 由于相关业务请求是分别提交给不同的 action 类处理的, 此时可以为这些 action 类定义一个父类, 在父类中设

置一个 `String` 类型的属性,用于存储文档内容,这样拦截器不仅可以作用于不同的 `action` 类,而且还可以直接应用于不同的 `web` 项目。

#### 4 结语

本文针对在线文本编辑器 `FCKeditor` 在 `java EE` 平台上的应用,提出一种使用过滤器或拦截器机制对上传附件自动清理的通用方法,并对该方法的设计思想、实现技术及软件实现进行了详细介绍,该方法的主要优点是:

(1) 利用 `servlet` 过滤器和 `struts` 拦截器实现类具有模块化设计及声明式配置方式的特点,使代码具有很好的重用性和可移植性。

(2) 不需要改动编辑器组件代码,也不会污染业务逻辑组件,易于实现。

(3) 附件的清理工作由系统自动完成,对用户是透明的,更为安全可靠。

本文主要以 `FCKeditor` 编辑器自带的文件管理器的应用为例,作为 `FCKeditor` 升级版的 `CKEditor` 中不再自带文件管理器,要实现文件上传功能,可以在 `CKEditor` 中集成 `FCKeditor` 的文件管理器<sup>[12]</sup>,或者使用 `CKFinder` 插件,或者使用其他任何一种文件上传组件,也就是说,如果使用其他文本编辑器或是其他文件管理器插件,只要能获得返回客户端的响应信息,过滤器的方法仍然是有效的(由于不同的组件返回的响应信息不同,需要修改提取附件的方式)。

另外,如果不使用 `struts` 框架,而是使用 `servlet` 处理请求和返回响应,可以将拦截器的功能移植到过滤器中。

(上接第 229 页)

#### 4 结语

本文讨论了构建 `PMS` 业务支撑平台的目的和意义,对平台的总体架构、设计思路和关键技术路线进行了详细介绍,目前平台对 `PMS` 业务应用系统的开发、运行、部署和管理进行了全面的支撑。与不基于平台构建的业务应用系统相比,平台在提高业务应用系统技术架构的一致性、充分共享和整合研发资源、提高软件开发质量和服务水平、降低建设与运维成本等方面效果非常显著,并可满足 `PMS` 一级部署、高性能数据处理等未来新的业务应用需求。

本文提出的附件管理方法已在自主开发的网络教学平台中得到有效的应用。

#### 参考文献

- 1 于梅英,姜波,张珂.基于 Java Web 的 HTML 在线文本编辑器解决方案——以 `FckEditor` 在线编辑器为例.软件导刊,2011,(2):36-37.
- 2 CKSource Docs. [http://docs.cksource.com/FCKeditor\\_2.x/Developers\\_Guide](http://docs.cksource.com/FCKeditor_2.x/Developers_Guide). 2008-06-05.
- 3 mamboChina.net. [http://www.mambochina.net/option,com\\_remository/Itemid,88/func,fileinfo/id\\_197.html](http://www.mambochina.net/option,com_remository/Itemid,88/func,fileinfo/id_197.html). 2007-06-12.
- 4 KFM FileManager. <http://kfm.verens.com>. 2008-03-23.
- 5 CoreFive FileManager. <http://labs.corefive.com/Projects/FileManager/>.
- 6 Phpletter FileManager. <http://www.phpletter.com/>.
- 7 CKFinder-Ajax File Manager. <http://ckfinder.com>.
- 8 FCKeditor.Java Integration. <http://java.fckeditor.net>. 2010-01-04.
- 9 Hall M, Brown L, Chalkin Y. Servlet 与 JSP 核心编程(第 2 卷).北京:清华大学出版社,2009:161-175.
- 10 李刚.Struts2 权威指南.北京:电子工业出版社,2007:207-216.
- 11 孙卫琴.Tomcat 与 Java Web 开发技术详解.北京:电子工业出版社,2009:518-525.
- 12 ckeditor 用 fckeditor 的文件管理器实现图片上传. <http://www.laokboke.net/2011/12/09/ckeditor-with-fckeditor-file-manager>. 2011-12-09.

#### 参考文献

- 1 徐向东,党倩,景中昌.省级电网生产管理系统设计应用与效益分析.电力信息化,2012,10(12):6-10.
- 2 董昕.电力企业统一生产管理系统基础架构.计算机应用与软件,2009,26(11):189-192.
- 3 吴俊敏,李黄海,黄刘生,鲍春健,王文韬.高可用应用开发平台的设计与实现.计算机工程,2006,24(12):283-285.
- 4 菅志清,祝顺吾.电力企业应用集成平台项目风险控制与管理.电力信息化,2012,10(8):73-76.
- 5 吴淑玮,闫训超.信息系统数据交换技术分析.计算机系统应用,2012,21(4):18-22.