

一种多 Portlet 之间交互方法的研究

马增辉¹, 解建仓¹, 张永进², 罗军刚¹

(1. 西安理工大学水利水电工程学院, 西安 710048; 2. 西安理工大学工商管理学院, 西安 710048)

摘要: J2EE 企业应用中, Portal 作为一种信息和资源整合模式, 能够有效地整合信息资源。Portlet 是 Portal Web 应用的重要组件。在实际的应用中, 多 Portlet 交互共同完成一项任务已经相当普遍, 而在现有的 Portlet 标准 JSR-168 中, 没有定义多 Portlet 交互的方法和实现, 难以满足 Portal 的应用需求。该文在 JSR-168 标准的基础上, 通过扩展 Portlet 实现接口、增加 Portlet 交互描述等方法生成 AAFPortlet 来实现多 Portlet 之间的交互, 并将此方法在开源的 eXo 中实现, 应用于实际系统, 效果显著。

关键词: Portal; 多 Portlet; Portlet 交互; 方法

Study on Approach of Interactive Multi-Portlet

MA Zenghui¹, XIE Jiancang¹, ZHANG Yongjin², LUO Jungang¹

(1. Institute of Water Resource and Hydro-electric Engineering, Xi'an University of Technology, Xi'an 710048;

2. Institute of Business Administration, Xi'an University of Technology, Xi'an 710048)

【Abstract】 As a kind of information and resources integration paradigm, Portal can effectively integrate information resource in the J2EE enterprise application. Portlet is the important component of Portal. In the actual application, it is quite ubiquitous that multi-Portlet together complete a task. However, the current Portlet criterion (JSR-168), doesn't define approach and realization of multi-Portlet interaction, which doesn't satisfy the demands of Portal application. By extending implement interface of Portlet, increasing interactive describe of Portlet and so on, AAFPortlet is produced to implement multi-Portlet interaction, based on JSR-168 standard. It is implemented in eXo and used in the practical application system. The effect is remarkable.

【Key words】 Portal; multi-portlet; interactive portlet; approach

企业信息门户(enterprise information portal, EIP)已经成为企业发布各种信息, 整合企业资源的重要工具。Portal 是一种基于 Web 的应用程序, 具有个性化界面定制, 单点登录(Single Sign-on, SSO), 以及信息聚合等服务功能, 通常在整个系统中为用户提供单一的访问入口、个性化界面定制和统一的安全管理, 提高了系统的表现能力。Portal 模型的核心技术是 Portlet 概念的引入, Portlet 是一个 Web 组件, 它是由 portal server 中的 Portlet 容器进行管理, 通过用户的请求产生动态片断(fragment)内容, 然后返回给用户展现内容。SUN 和 IBM 领导制定了 Portlet 1.0 规范^[1](简称 JSR-168), 提供了一个用户请求/响应的标准体系结构, 遵循 JSR-168 标准的 Portlet 可以在任何符合这个标准的 Portal 服务器上运行。

在 JSR-168 标准中, 并没有定义多个 Portlet 之间的交互问题, 而由于在实际的应用中大量出现多 Portlet 交互问题, 例如在水利行业实时水雨情门户中, 在条件查询 Portlet 发出查询请求后, 需要表格显示的 Portlet, GIS Portlet 以及过程线 Portlet 多个 Portlet 对条件查询请求作出响应, 已达到多种表现形式展现给用户的目的。因此, 提出一种对 JSR-168 标准 Portlet 扩展的模型结构, 实现多 Portlet 之间的交互。

1 JSR-168 及标准 Portlet 请求/响应模式

JSR-168 是基于 J2EE 1.3 规范, 这个规范全面地描述了基于 Java 技术的 Portlet, 定义了 Portlet 的 API 标准。

JSR-168 标准在 request 中定义了 processAction, render 两个方法, 当 Portlet 容器调用 Portlet 的 processAction 时,

request 形成 action URLs 向 portal 发送请求, Portal 容器将调用目标 Portlet 的 processAction 方法, 然后执行 render 方法, 与此同时其它 Portlet 将执行 render 方法。Render 方法会设置标题属性并执行 doDispatch 方法, doDispatch 根据 Portlet 状态模式分别处理 request 请求, doView 处理视图模式, doEdit 处理 EDIT 编辑模式, doHelp 处理帮助模式。如果 Portlet 的 Window 状态为 MINIMIZED, 则 render 不会执行任何一个方法。

客户端在 Portal 页面上对一个 Portlet 窗口进行操作并提交请求后, Portal 服务器分析请求, 获得需要激活的 Portlet, 然后把请求转交给 Portlet 容器处理, Portlet 容器调用被激活的 Portlet 的 processAction 和 Render 方法处理客户端请求, 输出新的信息片段。由于 Portal 页面上有多个 Portlet 窗口, 对应着一组 Portlet, 这些 Portlet 也要执行 Render 方法输出信息片段, 但是并没有处理客户端的请求, Render 方法输出的仍然是旧的信息片段。一组 Portlet 在执行 render 方法时, Portlet 并行执行, 这与 Portlet 容器的实现有关。Portal 服务器把 Portlet 输出的信息片断组成一个 Portal 页面返回到客户端, 完成对客户端的响应。门户系统中 Portlet 的请求/响应模式如图 1 所示。

基金项目: 国家“863”计划基金资助项目(2002AA113150, 2005AA113150); 国家自然科学基金资助项目(50279041)

作者简介: 马增辉(1977-), 男, 博士研究生, 主研方向: 区域经济与水资源管理, 水利信息化; 解建仓, 教授、博士生导师; 张永进, 副教授; 罗军刚, 博士研究生

收稿日期: 2007-02-10 **E-mail:** mzh@xaut.edu.cn

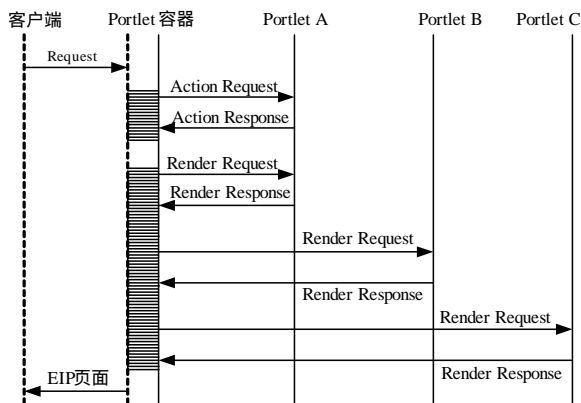


图 1 Portlet 的请求/响应模式

2 多 Portlet 之间交互的设计

Portlet 交互,指多个 Portlet 进行协同完成某项业务,用户的一个请求能够获得一组 Portlet 的响应。许多应用需要大量的 Portlet 进行交互,例如在水利应用中,需要通过一个统一的查询条件将时间,网站编码等信息作为参数传递给其它 Portlet(GISPortlet, 过程线 Portlet 和表格 Portlet),然后 Portlet 在进行数据源绑定和信息查询。这就是一个简单的 Portlet 交互的应用场景。

Portlet 交互需要完成以下功能,

- (1)Portlet 参数需要在多个 Portlet 间灵活的传递。
- (2)具有灵活性和可扩展性。
- (3)具有统一的 Portlet 开发模式和简化开发过程。

多 Portlet 之间交互的设计思路是通过 Portlet 接口、Portlet 注册以及 Portlet 的请求/响应模式的扩展,调整 Portlet 的运行调用过程,达到多 Portlet 之间交互的目的。与此同时,增加系统的灵活性和可扩展性,统一开发模式,简化开发过程。

2.1 继承 GenericPortlet 的 AAFPortlet 扩展接口设计

给出 AAFPortlet 扩展 JSR-168 标准中的 GenericPortlet 接口 UML 设计图,如图 2 所示。

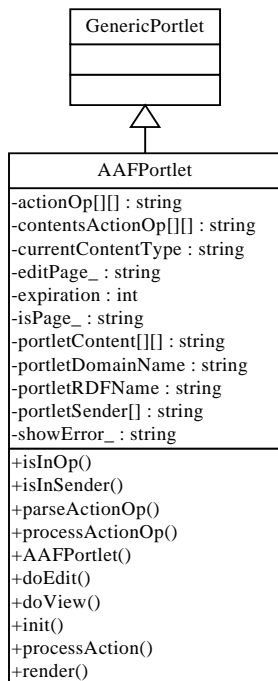


图 2 AAFPortlet 接口设计

通过扩展在标准的 JSR-168 标准中的接口,包含属性和

方法的扩展,实现 Portlet 的交互,具体如下:

(1)扩展属性

- 1)定义 actionOp[]属性:存储 action 的操作内容和结构。
- 2)定义 contextOp[]属性:存储 action 操作的上下文信息。
- 3)定义 PortletSender[]属性:存储 Portlet 的请求者信息。
- 4)定义 PortletContent[]属性:存储 Portlet 的上下文信息。

(2)扩展方法

- 1)isInOp():判断是否有该操作。
- 2)isInSender():判断是否和 Portlet 进行交互。
- 3)parseActionOp():执行具体的 Portlet 交互操作,包括配置属性读取、匹配。
- 4)AAFPortlet():AAFPortlet 构造方法。

其中对 GenericPortlet 的初始化方法进行重载设计和实现。在 Portlet 注册时 portlet 容器通过调用 Init()读取注册文件信息,提取 PortletSender、actionOp、contentsOp、PortletContent 等属性,存储到 Portlet 上下文中。一般来说,对基于同一个抽象 Portlet 的每一个具体的 Portlet 的初始化都在这个方法中进行。如果你选择重载这个方法,至少应该通过 super.init(PortletConfig)来调用它的父方法。

2.2 Portlet 注册扩展设计

通常在 Portal 应用系统,Portlet 部署在 portal server 上都必须写 Portlet.xml 注册文件,注册文件中包含 description、Portlet-name、display-name、Portlet-class、expiration-cache、Portlet-info 等 JSR-168 标准的条目,拟将扩展 default-view、sender、ActionOp 等属性条目,引入 Portlet 交互描述,达到 Portlet 间交互目的。其中 Sender 属性表示 Portlet 受到那个 Portlet 请求时,执行相应的 action 操作,ActionOp,表示有哪些具体的 action 操作请求。Default-view 表示 Portlet 初始页面设置。

以下以查询条件 Portlet 为例,

```

<Portlet>
  <description xml:lang="EN">查询条件</description>
  <Portlet-name>Query</Portlet-name>
  <display-name xml:lang="EN">查询条件</display-name>
  <Portlet-class>cn.edu.xaut.aaf.Portletbase.AAFPortlet</Portlet-
class>
  <init-param>
    <name>default-view</name>
    <value>/WEB-INF/templates/commonquery/commonquery.jsp<
/value>
  </init-param>
  <init-param>
    <name>sender</name> //请求者
    <value>Query;NEWQuery;MonitorQuery;XXZH_CZZHFA<
/value>
  </init-param>
  <init-param>
    <name>ActionOp</name> //实例操作接口扩展
    <value>Query; GisKeyStcdList.query@exchange; GisKeyStcdList,
query</value>
  </init-param>
  <init-param>
    <name>Contents</name> //Portlet 的上下文信息
    <value>ssyq_t_page;/WEB-INF/templates/water_info/shishiYL_T.jsp<
/value>
  </init-param>
</Portlet>

```

```

<name>ContentsActionOp</name>// 操作上下文
<value>ssyq_t_page|exchange;shishiYL_T.Query </value>
</init-param>
<expiration-cache>-1</expiration-cache>
<supports>
  <mime-type>text/html</mime-type>
</supports>
<Portlet-info>
  <title>查询时间</title>
  <short-title>查询时间</short-title>
  <keywords>查询时间</keywords>
</Portlet-info>
</Portlet>

```

2.3 AAFPortlet 的请求/响应及运行模式

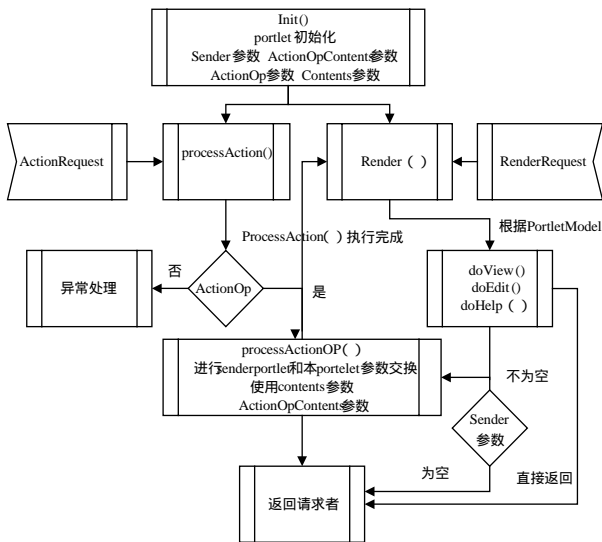


图3 AAFPortlet的请求/响应及运行模式

在标准 Portlet 请求/响应模式基础之上,加入 Portlet 间交互的操作步骤,提高 Portlet 的交互能力。AAFPortlet 的请求响应模式如图 3 所示。

3 实例

基于 AAFPortlet 的多 Portlet 之间交互已经在 eXo Portal Server 中实现,在此技术之上实现了陕西省水利厅防汛抗旱决策服务平台。在一个 portal 页面中:(1)测站组合方案和基本查询两个 Portlet,提供两种途径选择断面站号,在已选站 Portlet 显示测站选择结果;(2)查询条件 Portlet 提供时间条件;(3)在已选站、查询条件两个 Portlet 共同交互下,在表格、图形、GIS 服务 3 种不同表现形式 Portlet 中展示相同数据结果。说明了多 Portlet 交互的具体实现。

4 结论

基于 AAFPortlet 的 Portlet 交互方法,已经在开源的 eXoPlatform 得到了应用,并且解决了水利行业中的多 Portlet 交互的实际问题,为 Portlet 交互的实现提供了一种方法,取得了显著的应用效果。

参考文献

- 1 Sun Microsystems Inc., IBM Corporation. Java Portlet Specification[Z]. 2003-08. <http://www.jsp.org>.
- 2 陈毓林, 许舒人, 宋靖宇. 一个 Portlet 协作框架的分析与设计[J]. 计算机工程, 2006, 32(11): 86-88.
- 3 Song Jingyu, Wei Jun. Extending Interactive Web Services for Improving Presentation Level Integration in Web Portals[J]. Comput. Sci. & Tech., 2006, 21(4): 620-628.
- 4 Sun Microsystems. Java 2 Platform, Enterprise Edition Specification[Z]. 2001-08. <http://www.sun.com>.
- 5 Sun Microsystems. Java? Servlet Specification[Z]. 2001-09. <http://www.sun.com>.

(上接第 241 页)

从图 7 和图 8 中可以大致观察到如下的规律:

(1)当网格子节点的数目比较少时,基于 MA-GMA 的网格监控系统的总响应时间高于 GMA 架构。但当网格子节点数增多时,基于 MA-GMA 的监控系统的总响应时间则大大低于 GMA 架构。

(2)随着有效期的增加,网格监控的响应时间大体上呈下降趋势。这是由于缓存机制起到了作用,有效期的增加导致了缓存命中率的提高。

(3)当有效期分别为 30s 和 60s 时,响应时间的变化不明显。这说明有效期增加到一定程度后,对缓存的命中率和整体性能的影响变得不是很明显。

由于试验是在 Internet 上进行,网络的忙闲程度具有很强的随机性,试验结果存在一定的误差,但大体规律仍符合预期目标。

5 结论

本文在 GMA 架构的基础上,提出了基于移动代理的网格监控体系架构(MA-GMA),从几个不同的侧面解决了目前网格监控系统中存在的一些问题。针对现有系统的不足,将 MDS 中的缓存机制与移动代理技术应用于网格监控,为网格监控的研究提供了新的思路。同时还提出了设计与实现的优

化方案,并进行了试验验证。

虽然 MA-GMA 架构能够解决目前网格监控系统中存在的部分问题,但仍有许多值得改进之处,比如对网格监控的分层机制讨论不够深入,对整个网格监控的安全和授权机制没有过多考虑等。

参考文献

- 1 Foster I, Kesselman C. The Grid2: Blueprint for a New Computing Infrastructure[M]. Morgan Kaufmann, 2004.
- 2 Tierney B, Aydt R, Gunter D, et al. A Grid Monitoring Architecture[Z]. <http://www.didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-2.pdf>
- 3 Papaioannou T. On the Structuring of Distributed Systems: The Argument for Mobility[D]. Loughborough University, 2000.
- 4 Guo Shangfen, Zhang Wei, Ma Dan, et al. Grid Mobile Service: Using Mobile Software Agents in Grid Mobile Service[C]//Proceedings of International Conference on Machine Learning and Cybernetics. 2004: 178-182.
- 5 赵宏立, 庞小红, 吴智铭. 基因块编码的并行遗传算法及其在 TSP 中的应用[J]. 上海交通大学学报, 2004, 38(10): 213-217.