

# 一种多 Portlet 互操作框架<sup>①</sup>

刘高嵩 卢 维 龙 军 (中南大学 信息科学与工程学院 湖南 长沙 410075)

**摘 要:** 企业信息 Portal 作为企业信息集成的工具,主要是基于 Portlet 在表现层进行集成。这种集成实质上就是 Portlet 之间数据交互问题。在分析现有数据交互方法基础上,遵循 JSR-168 和 WSRP 规范,通过扩展 Portlet 实现接口,提出了多 Portlet 数据交互框架,并基于 XML 技术封装交互信息,实现 Portlet 之间信息交互。

**关键词:** 门户; 企业信息集成; 数据交互; Portlet 规范; WSRP 规范

## Framework of Multi-Portlet Interaction

LIU Gao-Song, LU Wei, LONG Jun

(Institute of Information Science and Engineering, Central South University, Changsha 410075, China)

**Abstract:** Portals are a means integrating entergration information through a portlet, which is a presentation component. It is the essence of data interaction between portlets. This paper proposes an approach for extending the interface of a portlet and implementing a portlet cooperation by supporting the communication between portlets. The approach is based on an established portlet Specification, a WSRP Specification, and an analysis of existing data exchange methods. This approach allows the exchange message of structured XML between portlets to complete the work together.

**Keywords:** portal; enterprise information integration; data interactive; portlet specification; WSRP specification

## 1 引言

企业信息集成是信息技术的研究热点。企业信息门户为企业应用集成提供了重要的途径, Portal 作为企业信息门户,集成各种应用,为信息整合提供个性化交互统一平台。Portal 的核心技术是 Portlet 组件, Portlet 是可复用的 Web 组件,它由 Portlet 容器进行管理,接收来自 Portal 容器的请求并进行处理,产生信息片断,然后 Portal 服务器聚集片断形成 Portal 页面<sup>[1]</sup>。JSR-168 和 WSRP 规范是 Portal 的两个主要标准。

企业信息集成是将企业内部或企业之间不同企业应用系统的数据和业务处理整合起来,实现跨越应用系统边界集成,同时还要保留原有应用系统,实质上

就是实现多个应用系统间互通互联<sup>[2]</sup>。Portal 作为一种企业信息集成的工具,主要是利用 Portlet 集成来实现企业信息集成。这种集成实质是通过 Portlet 之间进行数据交互通信来实现。但 JSR-168 并没有明确定义数据如何在多 Portlet 间传递的互操作。

目前 JSR-168 支持使用 PortletSession 这种简单的通信方式,即向 PortletSession 中添加参数 parameter 并将该参数的 Scope 属性设置为 APPLICATION-SCOPE,使得所有 Portlet 都能读写该 parameter,从而通过共享 PortletSession 实现通信<sup>[3]</sup>。当前主流门户产品在 Portlet 交互方面大多采用类似方法实现数据传递,如 Oracle portal 通过页面参数传递的方法进行 Portlet 交互协作。首先,在

<sup>①</sup> 基金项目:国家自然科学基金(60873081, M0921005, U0835003);高等学校博士学科点专项科研基金(20090162110072)

收稿时间:2010-01-15;收到修改稿时间:2010-03-18

Portal 页面建立页参数相当于页面的全局变量,然后在 Portlet 中设置 Portlet 参数,并与页面参数建立对应的关联关系,Portlet 的参数传递到页面的全局参数是通过事件来传递<sup>[4]</sup>。

通过 PortletSession 实现数据交互,实质是一种参数值传递,值传递方法可以很好的解决同一 Portal 页面的交互协作问题,但是该方法存在以下不足:首先当页面内容庞杂时,大量属性信息修改容易引发逻辑错误导致程序混乱;其次这种协作方式仅仅适用于同一系统中的 Portlet 之间传递简单的交互数据;最后这种方式并不支持与部署在远程服务器上的 Portlet 之间的通信。这种应用相当有限,无法从根本上满足当前门户在应用向集成化和协作化方向发展的要求。因此研究 Portlet 间的数据交互很有必要。

## 2 多Portlet互操作框架设计

Portlet 交互协作是指多个 Portlet 之间进行协作完成某项业务,即用户的一个请求数据能够得到一组 Portlet 响应<sup>[5]</sup>。许多应用需要大量 Portlet 进行交互,例如在一个应用系统里面,有 3 个 Portlet 分别是 A、B 和 C。其中 A 显示有关账户的概要信息,B 显示在过去 20 天内有关账户发生的所有事务信息,而 C 显示某一特定账户的详细信息。这是一个简单的 Portlet 交互协作应用场景。为实现 Portlet 交互需要解决如下问题:

①支持跨越应用系统边界的通信;②与远程 Portlet 进行通信并支持异构系统;③在 Portlet 之间支持 1:N 通信,而且能够传送复杂数据;④必须考虑系统可重用性,可移植性以及对已存组件的集成问题。

参考当前 Portal 中 Portlet 的交互协作的设计思路,依赖 Portlet 和 WSRP 规范的核心特征,扩展 Portlet 实现接口,提出多 Portlet 数据交互框架,基于 XML 封装技术在 Portlet 之间进行复杂数据交互,实现 Portlet 间信息交互,达到集成目的,同时保证系统的灵活性和扩展性。

### 2.1 扩展标准 Portlet

为增强 Portlet 之间交互协作,充分利用 Portlet 规范,对标准 Portlet 进行扩展:增加了输出端口和输入端口的概念、一个 Message 类和扩展公共接口 GenerateMarkup。输出端口表示数据源 Portlet 要传送的数据;输入端口表示目标 Portlet 准确接收数据的验证数据结构;Message 是一个简单的 javabean,它的作用是对源 Portlet 传送的数据进行封装,表示数

据输出端口结构;GenerateMarkup 接口提供一个 embXMLasNote(Message msg)方法,该方法是将 Message 对象使用 XML 来表示,返回值是一个字符串。

自定义 Portlet 通过继承 Portlet 规范中的 GenericPortlet 抽象类并实现 GenerateMarkup 接口。当该 Portlet 接收到请求后,进行响应处理,如果需要提供交互数据,则将交互信息以 Message 形式表示,再将 Message 以形参方式传递给 embXMLasNote 方法产生字符串,然后将字符串以注释的形式嵌到由 GenericPortlet 接口的 render 方法形成 html 片段。

数据输出端口结构包括 name、message\_type、broadcast、generatemarkupnested、message 和 to\_portlets 六个数据项组成,各数据项的具体意义由图 1 所示。为了保证 Message 能够被 Portlet 准确地接收,仅仅靠输出端口中的 name 和 to-portlets 是不够的,为此提出了数据输入端口。数据输入端口由 name、from-portlets、message-type 和 action 四个数据项组成,各数据项具体含义如图 2 所示。数据输入端口以 XML 文件存在,其主要作用是匹配输出端口中的 name 和 to-portlets,确保 Portlet 的 action 能够准确接收数据。

由上可知数据源输出端口通常嵌套在 Portal 页面的 html 标签中,这意味着数据源 Portlet 的输出端口能够在每个显示的页面中进行修改。图 1 显示输出端口的 XML 描述,将以注释的形式内嵌在 HTML。标记 <name> 和 <to-Portlets> 可以确定相互关联的 Portlet;标记 <messages> 中包含该 Portlet 要传送的数据;标记 <message> 定义传送数据的具体类型。图 1 显示两个 Portlet 之间数据交互关系,其中 org.ssh.personData 为 org.ssh.showPersonData 提供用户详细信息。

```

<port>
<name>org.ssh.personData</name><!--产生该信息的 portlet 完整类名-->
<message-type>org.ssh.model.Person</message-type><!--发送的完整数据类型-->
<broadcast>true</broadcast><!--进行广播 -->
<generatemarkupnested>true</generatemarkupnested><!--表示信息是否以 xml 形式嵌入到该 portlet 产生的 html 片段中-->
<to-portlets><!--表示源数据发送到目标 Portlets, 包含 to-portlet 子标记-->
<to-portlet>org.ssh.showPersonData</to-portlet><!-- 源数据发送到目标 Portlet-->
</to-portlets>
<messages><!--传送的数据内容, 包含 message 子标记-->
<message type=" org.ssh.model.Person"><!--传送具体数据类型-->

```

图 1 输出端口 xml 描述

数据输入端口同样用 XML 进行描述并保存在独立描述文件中, 可以通过 URL 进行访问。图 2 显示简单输入端口 XML 描述文件。输入端口除了包含端口名字和类型外, 还包括了 action 属性, 用于确定这个输入端口由哪个 action 接受数据, 该 action 标记依赖于数据接收器实现的具体分发机制。图 2 表示由 org.ssh.personData 传送过来的数据被 org.ssh.showPersonData 的 showData 方法接受。

```
<?xml version="1.0" encoding="utf-8"?>
<port>
<name>org.ssh.showPersonData</name><!--输出端口对应 portlet 的完整
类名-->
<message-type>org.ssh.model.Person</message-type><!--指定源 Portlet
发送的完整数据类型-->
<from-portlet>org.ssh.personData</from-portlet><!--表明发送数据的源
portlet 的名称-->
<action>showData</action><!--处理数据的具体方法-->
</port>
```

图 2 输入端口 xml 描述

自定义 Portlet 通过继承 Portlet 规范中的 GenericPortlet 抽象类并且实现 GenerateMarkup 接口的方式提供数据输出端口, 可以通过标记 <name>与<to-portlets>在 Portlet 间建立 1:N 数据交互关系, 标记<messages>进行封装传送的数据, 利用标记<message >确定交互数据具体类型, 通过这种方式可以支持 Portlet 间的数据交互。输入端口利用<name>与<from-portlet>标记确保数据可以被正确的 Portlet 接收起验证作用。输出和输入端口采用 XML 描述是由于 XML 可以在不破坏原有信息内容的同时, 可以自由定义、扩充、修正标记和文档结构, 以满足不断变化的应用需求, 实现异构系统间的信息表示、交换和集成。

2.2 多 Portlet 互操作框架

Portal 服务器是 J2EE Web 服务器, 它允许将 Web 应用程序部署在 Servlet 容器, 该容器为 Web 应用程序的每个组件定义并维护运行环境, 组件共享用户会话状态和全局应用状态。Portal 可以看作一个特殊 web 应用, 它由 Servlet 容器, Portal 应用服务程序, Portal web 应用和 Web 应用四部分构成。为了

支持扩展的 Portlet 组件之间交互, 在 Portal 应用服务程序中增添了四个额外组件: 数据过滤服务组件、Portal Dispatch Servlet、注册器 Servlet 和数据缓存存如图 3 所示。

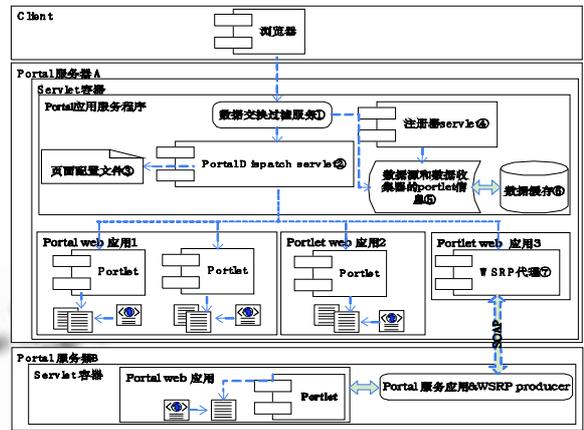


图 3 数据交互服务体系框架结构

由 2.1 及图 3 可知 Portlet 可以分为三种类型: 充当数据源的 Portlet、目标 Portlet 用于接收数据以及代理 Portlet 用于代理远程 Portlet。所以该 Portal 应用服务程序处理的数据结构主要是输入和输出端口。具有数据交互功能的扩展 Portlet 跟其他 Portlet 一样也是由 Portlet 容器进行管理。新增组件功能如下:

**Portal Dispatch Servlet 组件:** 接受页面请求, 然后根据页面配置文件, 将请求转发到 Portlet 容器由具体的 Portlet 进行处理, 将 Portal 页面返回给数据过滤服务组件。

**数据交换过滤服务组件:** 负责对返回的 Portal 页面进行分析和转换工作。主要是抽取源 Portlet 产生的片段, 形成一个独立数据输出端口 XML 文件存放在应用服务程序的 temp 文件夹下, 该 temp 文件能够被不同系统边界中的 Portlet 应用程序访问并利用 XML 解析器对该临时文件进行解析; 在原来数据输出端口位置将用一个 URL 和一个应用菜单生成逻辑所代替, 其中 URL 指向数据接收器 Portlet 的地址, 而插入的应用逻辑主要是根据<to-portlets>标记产生一个可供选择的导航菜单, 通过用户交互的方式, 触发解析数据输出端口描述文件与数据输入端口描述文件进行匹配, 若匹配成功则将数据传送到用户选择的 Portlet 处理。数据过滤服务组件是新增组件中最重要也是最

复杂的一部分,其结构如图4所示。

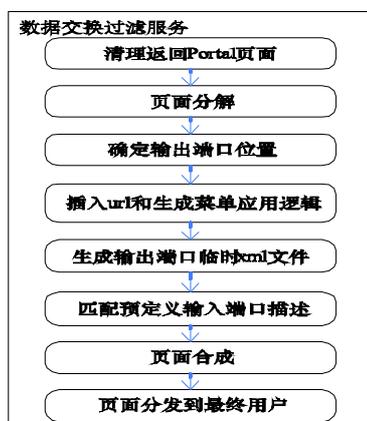


图4 数据交换过滤服务组件结构

注册器 Servlet 组件:负责登记第一次被请求的 Portlet 信息,以及将数据过滤服务组件对页面分析和转换后的信息进行登记。

数据缓存:主要是用于缓存已经使用过的 Portlet 信息以及 Portlet 之间的数据交互关系。

带有扩展服务器组件的 Portal 请求响应处理主要步骤:①客户端浏览器向 Portal 服务器发送一个 page 请求,经验证通过后将请求交给数据交换过滤服务组件;②Portal web 应用服务程序中数据交换过滤服务暂时接收该 Page 请求,然后传送到 Portal Dispatch Servlet;③Portal Dispatch Servlet 读取页面配置文件,判断请求是否包含与组成门户网站网页的 Portlet 有关动作;④如果存在与某 Portlet 相关动作,Dispatch Servlet 将该请求分发到 Portlet 容器调用相应的 Portlet 处理请求;⑤Portal 通过 Portlet 容器调用 Portlet,产生该请求的响应数据,若有数据交互的操作,则将使用 GenerateMarkup 中的 embXMLasNote 方法将交互数据嵌入到内容片段;否则直接产生内容片段;⑥Portal 通过 frame 将产生的结果聚集于门户网站的网页,然后将网页返回到数据交换过滤组件;⑦数据交换过滤组件将对网页进行分析和转换操作并判断此次请求页面中的 Portlet 是否为第一次请求;分析网页中是否存在数据输出口注释,如果存在则将该端口抽取形成 xml 临时文件存放在 temp 文件夹下;并在输出端口的位置用 URL 和

一个处理逻辑代替,其作用已在上面提过不再赘述;

⑧如果 portlet 是第一次请求则将该 Portlet 由注册器 Servlet 进行登记;⑨将分析和转换操作的结果和注册器的内容存入数据缓存;⑩数据交换过滤组件将处理后的网页重新组装返回客户端。

具有数据交互功能的 Portlet 注册数据源与数据接收器的处理时序如图5所示。浏览器将页面请求发送到 Portal 服务器;Portal 服务器组装 Portlet 产生的片断形成页面;将该页面返回给过滤组件,过滤器抽取 Portlet 的信息和输入输出端口,输出端口由 URL 代替,并确定使用该数据的 Portlet;Registry 注册 Portlets、ports 和 URL,并将这些信息存入 Cache;最后 Filter 将处理后的页面返回给浏览器运行。

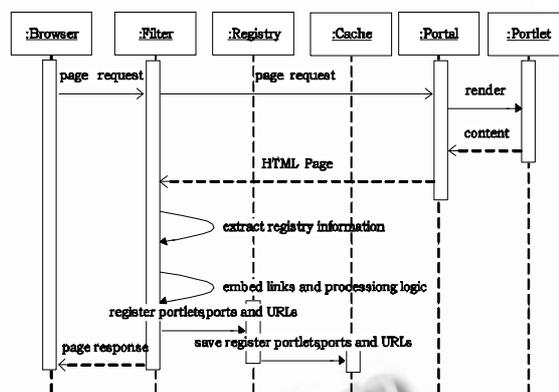


图5 注册数据源与数据接收器

为了能集成远程 Portlet,扩展 Portlet 组件同时也遵循 WSRP 规范。远程 Portlet 通过代理 Portlet 充当 WSRP 消费者进行集成。它们既可以作为数据源也可以当作数据接收器。此时对于远程 Portlet 需增加额外数据项该 Portlet 的地址,它和输出端口描述都将嵌入到 Portlet 生成的片断,并通过 WSRP 协议中的 getMarkup 方法获得。将数据发送到远程 Portlet 利用 SOAP 协议发送给 Portlet 的 ActionURL,通过 WSRP 协议中的 performBlockinginteraction 方法调用结果[6]。

### 3 分析

首先自定义 Portlet 通过继承 Portlet 规范中的 GenericPortlet 抽象类并且实现 GenerateMarkup

接口的方式提供数据输出端口,利用 Message 对象封装,并将其作为参数传递给 `embXMLasNote` 方法,最后将输出端口以注释形式写入片断中。Portal 将所有片断组装成一个完整的页面,返回给数据过滤服务组件,数据过滤服务组件将识别嵌入在片断中的注释并进行抽取形成一个独立的 xml 文件存放在 portal 服务器中的 temp 文件夹下,这样不论是否为同一系统中的 Portlet 均可访问这些文件,数据接收器可以解析该 temp 下的 xml 文件并与自身预先定义的输入端口文件匹配确定交互数据是否由该接收器接收,这样就可以实现跨越系统边界的通信交互。

其次数据输出端口的数据结构可以表示在 Portlet 之间进行 1:N 通信,而且能够传送复杂数据。由于数据输出端口是基于 XML 技术表示,可以利用 XML 在不破坏原有信息内容的同时,自由定义、扩充、修正标记和文档结构,以满足不断变化的应用需求,实现异构系统间的信息表示、交换和集成不断变化。

最后在 Portal 应用服务程序中提供的 4 个组件部署在 Servlet 容器,与 Portlet 容器基本上是独立的,这样有利于降低与 Portlet 容器的耦合,可以实现系统的灵活性、扩展性和重用性。由于对 Portlet 扩展的实现接口是在 Portlet 规范和 WSRP 规范基础上进行扩充,因此可以对标准 Portlet 以及远程 Portlet 进行集成。

#### 4 结束语

企业信息集成是当今信息技术的研究热点,企业

信息门户是企业信息集成的重要工具,利用 Portal 进行信息集成的实质就是 Portlet 间的信息交互问题。本文分析了目前主流 Portlet 交互的常用方法以及它的局限性,遵循 JSR-168 和 WSRP 规范,并依赖这些规范的核心特征,通过实现 GenerateMarkup 接口,扩展 Portlet 的实现,提出输入和输出端口的概念以及多 Portlet 数据交互框架,并基于 XML 封装交互信息,实现 Portlet 间信息交互,协同完成业务工作,达到企业信息集成目的。该方法优点是扩展 Portlet 组件实现集成方式简单,可以跨越系统边界传送复杂的数据,支持远程 Portlet 通信,具有很好的灵活性、扩展性和重用性。

#### 参考文献

- 1 Java Community Process. The Java Portlet Specification(JSR168)[2009-9-12].<http://jcp.org/en/jsr/detail?id=168,version1.0>.
- 2 五云庆,杨丹.基于门户技术的 EAI 实现方法研究.计算机工程与设计,2007,28(6):1418-1419.
- 3 陈毓林.Portlet 协作框架研究.北京:中国科学院软件研究所,2005:19-20.
- 4 马将,聂瑞华等.基于 JetSpeed 的多 portlet 流程协作机制研究与应用.计算机系统应用,2009,18(4):70-71.
- 5 马增辉,解建仓等.一种多 Portlet 之间交互方法的研究.计算机工程,2007,33(13):242-243.
- 6 OASIS Web Services for Remote Portlets (WSRP) Specification [2009-9-12].<http://www.oasisopen.org>.