基于 JetSpeed 的多 portlet 流程协作机制研究与实现①

Research and Implementation of Portlets Process Cooperative Mechanism Based on Jetspeed

马　将[1]　聂瑞华[1,2]　罗辉琼[2]　黄序鑫[1]　林怀恭[1]

(1 华南师范大学 计算机学院 广东 广州 510631; 2 华南师范大学 网络中心 广东 广州 510631)

摘　要： 当前门户的功能定位已经从传统的信息集成转向应用集成。门户环境中应用间的进一步集成实际上表现为 Portlet 协作问题。在分析现有协作方法的基础上，遵循 jsr286 规范，提出了可扩展的多 Portlet 的流程协作框架，并结合开源门户产品 JetSpeed2.0 进行开发，从而解决了数字校园建设中利用门户实现应用交互协作和个性化信息服务的关键问题。

关键字： 门户 portlet 协作 JSR286 shark

1　引言

门户是以 portal 为核心，集成各种应用系统，为信息整合提供个性化交互的统一平台。Portal 的核心技术是 Portlet 组件，Portlet 是一个可复用的 Web 组件，它由 portal server 中的 Portlet 容器进行管理，接收 Portal 容器传来的请求进行处理，然后向用户显示动态的内容，其中同一 portlet 的多个实例为每个用户显示不同的数据。多个 portlet 产生的内容聚集在一起形成 Portal 页面。

Portlet 协作是指多个 portlet 通过某种协作机制共同完成某种业务的过程。从某种意义上讲，门户环境下的应用集成问题实际上是 Portlet 之间协作或者交互问题[1]，这也是解决信息孤岛的关键问题。

2003 年 JCP (Java Community Process) 发布的 Java Portlet 规范 V1.0 规范 (简称 JSR-168)定义了 Portlet 容器和可用于与用户特定 Portlet 代码进行交互的标准接口(Portlet API)．所有遵循 JSR-168 标准的 Portlet 可以在任何符合这一标准的 Portal 服务器上运行[2]。但 JSR-168 里面并没有明确定义数据如何在多 portlet 间传递的互操作或者协作。这就无法从根本上满足当前门户在应用向集成化、协作化方向上的要求。刚刚发布的 JSR-286 规范引入了事件、共享呈现参数等不同的通信机制，为多 Portlet 应用程序实现应用协作提供了便利[3]。

Jetspeed 2.0 是 Apache 软件基金会的开源门户项目。由于它采用 Spring Framework 的组件架构，依照 MVC 模型来构建，功能完备，且完全兼容 Java Portlet API 标准，并易于扩展[4]。在遵循 Java Portlet 两个规范的基础上，提出了一种多 portlet 流程协作框架，并在 Jetspeed 平台上通过 shark 工作流引擎实现。

2　当前主流门户的portlet协作机制

目前主流门户产品在 portlet 协作方面大多是在 JSR-168 的基础上通过某种机制实现值传递，比如 Oracle portal 通过页面参数传递的方法进行 portlet 协作.首先，在 portal 页面建立页参数(相当于页面的全局参数)，然后在 porlet 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。

① 基金项目:华南师范大学“十一五”、“211”公共服务平台建设项目
收稿时间:2008-10-14

在 WebSphere 的 Portlet 协作中，通过 Property Broker 用来维护与 Portlet 相关的属性，为 Portlet 协作提供服务接口。在 Broker 内部 Websphere 利用 wiring tool 在输出属性和所有与之匹配的输入属性之间建立连接。当源 Portlet 的属性值发生变化时就调用所有与之相连接的输入属性所在的目标 Portlet 来协作完成一项业务流程。这种操作可以通过输入属性更改 setProperties()方法调用，还可通过 changedProperties()方法提供任何数量的输出属性更改[5]。

其实这些方法与 JSR—286 提出的共享呈现参数相似，都是一种参数值的传递，值传递的方法能很好的解决各个用户 portal 内尤其是在一个 Portal 页面内的 Portlet 协作问题，但在页面内容庞杂时，大量的属性信息容易引发逻辑错误，并且这种协作只是简单的值传递，应用相当有限，且这种机制不支持异步协作，不支持协作的并发，循环以及跳转等，因此必须加以扩展，以适应门户中多 porltlet 的协作与衔接。

3　多portlet的流程协作框架

门户中的各种应用不仅仅是独立的聚合在一起，更重要的是要能灵活地相互协作，真正消除信息孤岛，提高工作效率[6]。多 portlet 的流程协作就是在各个 portlet 应用独立的基础上，进行扩展，使 portlets 能够按照可变更的流程实现灵活的协作，是应用集成的高层次体现。

JSR－286 规范对事件的支持是 portlet 流程协作框架提出的基础。流程协作要包含协作的基本步骤，这些基本步骤应该包含：①流程的初始和激活；②允许协作流程从一个 portlet 提供的端口上调用、请求/响应(request/response)其他 portlet 的操作；③协作流程的中断控制；④协作流程包含对事件的激活与应答处理；⑤允许数据在 portlet 间传递；⑥抛出错误或异常处理；⑦整个协作流程的终止。

基于流程的 portlet 协作除了要实现流程协作，实现流程的自定义，应用的扩展以及对用户的透明，还必须考虑到门户中 portlet 组件的交互性和重用性、门户之间的持久性逻辑和状态信息的能力[7]。图 1 是一个整体流程协作框架，其中的流程配置器用于协作流程的定义，使协作流程可以灵活配置，协作流程缓存可以提高协作流程的存取效率，管理及监控工具使协作流程可控，协作流程引擎负责流程的初始化和启动工作。在中间参数的传递中整个协作流程完成流转。

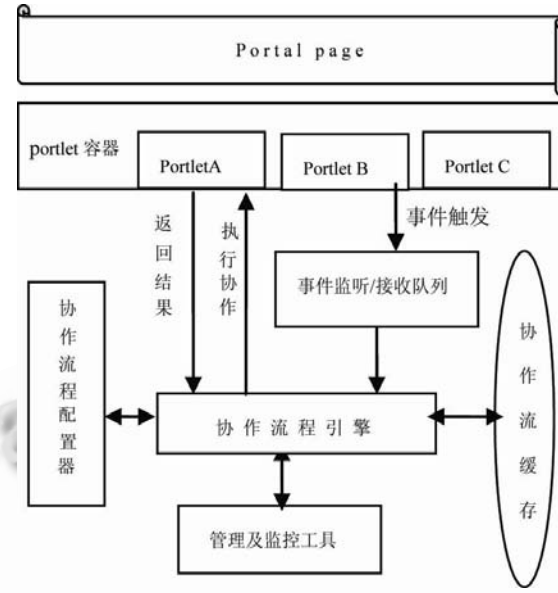

图 1　portlet 流程协作框架

根据门户的实际应用需求，将流程协作进行了细化，下面按照结构化的 portlet 流程协作和用户交互的 portlet 流程协作分别予以讨论。

3.1 结构化的 portlet 流程协作

结构化的 portlet 流程协作最大的特点是流程按照结构顺序自动执行，这种 portelt 的交互对用户来说是透明的，流程要跨 portlet 应用，应用或服务之间可能由于无法理解相同的操作集合而不能够彼此进行协作，可以使用接口映射或对象映射来解决。

在实现上，使用调用方法的方式启动，以便在流程结束后返回输出消息。结构化的 portlet 流程协作通常运行时间较短，在数据库中不保存运行时值，只是暂时保存在缓存里，另外不包括中断性流程。

在 JSR 168 Portlet 的开发中，开发者通常继承抽象类 javax.portlet.GenericPortlet 来实现自己的 Portlet 逻辑代码[8]。同 JSR 168 相比，JSR 286 不仅在 portlet.xml 里增加了对时间的声明、定义等，还在 GenericPortlet 增加了 javax.portlet. EventPortlet 接口的实现，从而增加了事件处理的功能。

为了增强对结构化流程的支持，首先要根据 portelt 的协作需求在 GenericPortlet 类中的 processaction 方法里增加结构化的逻辑，里面包含相应的结构化任务。这些结构化任务应包括：①Sequence

任务——定义一个有序的事件序列；②Switch 任务——根据条件选择某一事件；③While 任务——定义循环执行，直至满足某一个条件的一组任务；④Flow 任务——定义一组应并行执行的事件。

结构化定义完成以后，将由事件启动协作流程的开始。Event 并不直接指向某个目标 portlet,而是指向协作流程引擎，协作流程由配置器定义，可以新增、修改协作流程。在框架中配合流程引擎和配置工具，将 portlets 之间的协同工作自动化。协作流程引擎主要负责流程的初始化和启动工作，将协作路由带下一个活动或者输出节点，并监视 portlet 协作流程的执行，记录参数值的变化，并做出合理的反应如终端操作或抛出异常。一个协作流程被启动后，将按照预先定义的流程逐个调用目标 portlet,目标 portlet 根据协作参数执行方法并反馈到页面或者返回结果为流程的下一步执行提供参数。协作引擎可以根据参数值对协作流程做进一步细化和分流。

3.2 用户交互的 portlet 流程协作

结构化的流程协作主要在门户后台自动处理运行，整个流程由开发人员定义，用户无法直接参与交互，而用户交互的 portlet 流程协作则是用户之间通过 portlet 的流程协作实现多用户交互，比如上报或委派用户任务，申请贷款，入学注册，公共课程课程选修，完成论文后的学位申请等，所以必须充分考虑协作的异步性，即协作方并不一定同时处于活动状态，力求使这种协作既是基于流程，又能动态、异步交互。

另外业务流程必须是可中断的，协作过程可能要接受协作方用户的输入或者其他操作，因此在协作未完成时必须持久地保存运行时值，正常情况下协作会运行很长的时间[9]，协作过程中人员、流程和信息连接到一起，属性信息以及协作上下文应保存在各门户的 portal sever 上而不是在登录用户的客户端。

4 Portlets流程协作的实现

协作流程引擎采用开源的 Shark 工作流引擎并利用 Java 技术实现。Enhydra Shark(简称 Shark) 是一个著名的开源工作流引擎。它完全基于 WFMC(Workflow Management Coalition) 和OMG (Object ManagementGroup)标准，模块独立性强，并且非常可配置。以 JaWE(一个开源建模工具) 生成过程定义文件,控制工作流引擎中的任务的产生。用扩展后符合 jsr286 规范的开源 Jetspeed 技术完成表示层与应用层的搭建。Shark 和 jetspeed 都运行在 tomcat 应用服务器上。

通过实现事件监听接口和相应 adapter 建立 portlet 应用与 shark 工作流引擎之间的连接。需要强调的是，Shark 只定义与协作流程相关的逻辑，而具体的功能则有其所调用的 portlet 应用来实现并最终通过 render 方法展现，与 shark 无关。下面给出过程建模后 xpdl 文件的部分代码,这其中包含了协作流程的过程定义，并能够被 shark 工作流引擎解释。

```
<xpdl:Participants>   //对用户的描述
  <xpdl:Participant Id="test" Name="Test">
  <xpdl:ParticipantType Type="student"/>
</xpdl:Participants>
<xpdl:Applications>   //对 portlet 应用程序的描述，其中 ID 是唯一标识
    <xpdl:Application Id="portletapp1">
    <xpdl:FormalParameters>  //对 portlet 应用程序中参数以及属性的描述
      ……
  </xpdl:Application> </xpdl:Applications>
  <xpdl:WorkflowProcesses>   //协作流程的过程描述，包含初始化参数等
    <xpdl:WorkflowProcess  AccessLevel="public" Id="1" Name=" cooper_process1">
      <xpdl:Activities>   //活动
       </xpdl:Activity>
          <xpdl:Transitions>   //过程内的协作流程转移
              </xpdl:Transition>
      </xpdl:WorkflowProcess>
      </xpdl:WorkflowProcesses>
```

之后还必须进行用户映射和 portlet 应用映射并进行实例化等，使其既支持结构化的协作又可以支持多用户交互的 portlet 协作。

协作流程在执行期间不可避免的会产生异常和错误，这时就需要通过确定行为来让协作流程引擎处理。Shark 支持动态工作流机制，能通过 Dynamicredesign 修改其自身来支持更复杂的工作流环境或组织的异常处理。Exception Handling 允许处理未预见的事件，同时动态的重新定义可能导致

某些实例进入无效状态，此时用异常处理可以恢复这种状况或强制终止该协作流程的执行。为了将流程的状态回滚，回到跟进入作用域前一样，需要将该作用域内已执行部分采用其它行为进行撤销。通过 Spring 的 Declarative Transaction 机制，Jetspeed 很轻易实现了细颗粒度的事务管理，用户可以很容易配置需要管理事务的方法。

基于 JetSpeed 平台的多 portlet 流程协作机制已经在华南师范大学校园信息门户中得以实现，并且解决了多 Portlet 在传递数据的同时进行流程协作的问题，取得了显著的应用效果，但是还必须对流程的深度和并发度进行控制进而保证整个门户系统的稳定性。

5 结束语

本文研究了 portlet 流程协作机制，并在 Jetspeed 和 shark 工作流引擎的基础上进行二次开发得以实现。门户中集成了大量的 portlet 应用，但这些 portlet 应用并不应该是孤立的，它必须具备根据不同的用户角色以及业务环境的改变而做出相应反应的能力。通过多 portlet 流程协作机制可以使它们协调一致的工作，并可以按照一定的业务需求进行组合，真正实现应用集成，从而缩短工作周期，提高工作效率。

参考文献

1 宋靖宇,魏峻,万淑超.门户环境中基于语义数据协作应用集成方法.软件学报, 2007,18(7):1705－1714.

2 JSR 168-Portlet Specification 1.0.http://jcp.org/ en/jsr/detail?id=168

3 JSR 286-Portlet Specification 2.0.http://jcp.org/en/ jsr/detail?id=286.

4 Jetspeed 2 enterpriseportal.http//portals.apache.org／jetspeed-2.

5 Amber RC, Wu YPC. Developing JSR168 compliant cooperative portlets.http://www.ibm.com/developerworks/websphere/library/techarticles/0412_roy/0412_roy.htm.

6 Jones C.门户体系架构技术探讨. IBM 软件集团.http://www.xyf.com.cn/ibm/2003/portal.doc,2002.

7 聂瑞华,等.数字校园规划与建设方案.华南师范大学数字校园建设办公室, 2006.

8 刘美荣,刘建,马殿富.支持 Portlet 互操作的容器.计算机集成制造系统, 2007,13(6):1241－1248.

9 shark 开源项目 http://download.us.forge.objectweb. org/shark/shark-1.0-1.src.zip.