第47卷第3期 大连理工大学学报 Vol. 47, No. 3
2007年5月 Journal of Dalian University of Technology May 2007

文章编号: 1000-8608(2007)03-0444-04

门户集成 Struts应用的 Portlet化模型研究

那日萨*, 郭宇鹏, 贾运蕾

(大连理工大学 管理学院, 辽宁 大连 116024)

摘要: 在最少改变编码的前提下,将现有运行良好的应用系统移植到门户中是门户集成所面临的首要问题. 针对门户构建中基于 Struts框架的 Web应用集成,分析了 Struts框架和 Portlet运行机制及二者的关联性,提出了一个Struts应用的 Portlet化模型,并实现该模型. 实例证明遵循 Struts标准的应用均可以利用该模型实现为 Portlet,作为组件集成到门户中,从而有效地提高了现有系统的可重用性.

关键词: 门户集成; Portlet; Struts; MVC
中图分类号: TP311.54 文献标识码: A

0 引 言

门户(portal)旨在把各种应用系统、数据资源和互联网资源集成起来,并根据用户的使用特点和角色,形成个性化的应用界面,它目前已经成为应用集成的有效解决方案之一[1]. Portlet作为门户的核心部件,是一个多层次的、面向用户的应用程序[2],同时也是一种应用集成技术[3]. 由于能够很好地满足门户中用户的个性化需求, Portlet受到了开发者和应用者的青睐.

但在通过门户集成应用系统的过程中,如何最大限度地利用好已有资源,在改变最少编码的前提下将现有系统移植入门户中,仍是一个亟待解决的问题. 目前, Portlet已成为门户集成的解决方案,且已被很多知名的软件供应商实施[4]. 一些学者也从门户个性化定制、Portlet的可重用性以及 Portlet在门户中的协同工作等方面进行了相关研究[5-8]. Díaz等[2]对将 Web应用 Portlet化进行了探讨,认为可以通过一个桥 Portlet来实现,但具体如何实现仍在探索之中. 因为到目前为止,尽管业界公布的 WSRP标准和 JSR168标准为 Portlet提供了统一标准,并允许将 Web应用系统引入到门户领域中来,但是,这两个规范并没有明确指出一个 Portlet应该怎样被实现,甚至一个已经存在的 Web应用系统,应该怎样被部署成为一个 Portlet.

针对目前各组织单位拥有许多运行良好的 Struts应用系统的现状,本文提出一种使用 Portlet技术将现有 Struts应用 Portlet化的模型,即将一个基于 Struts框架的应用实现为一个 Portlet,并进行实际应用,以验证所提模型的有效性.

1 Struts框架与 Portlet的关联性

Struts框架是 MVC设计模式的一种具体实现方式[9],而 Portlet技术是基于 MVC设计模式的[10],二者的运行机制分别如图 1、2所示.

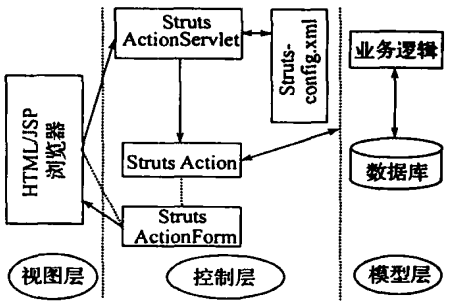

图 1 Struts框架工作原理

Fig. 1 The operational mechanism of Struts framework

在 Struts框架中,模型组件通常采用 JavaBean来实现,用来描述应用程序中的数据和操作该数据的方法,与用户界面无关;控制器组件

收稿日期: 2005-09-23; 修回日期: 2007-03-16.
基金项目: 国家自然科学基金资助项目(70272050; 70571007).
作者简介: 那日萨*(1970-), 男, 博士, 副教授.

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

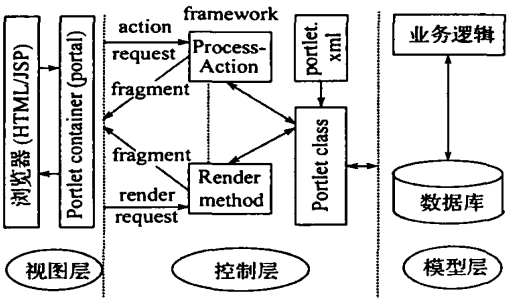

图 2　Portlet运行机制

Fig. 2　The operational mechanism of Portlet

是整个框架的核心部件,是通过专门的 Servlet来实现的,它用来处理属于 Struts处理范围的用户请求,将用户操作翻译为对模型组件的操作;视图部分可以采用 JSP实现, Struts自定义标记库也可以实现模型组件与视图组件的有效交互并能够增强显示功能. 所有这些处理流程,都在 Web服务器里执行,基于 Struts框架的 Web应用被认为是自我满足的应用, Web服务器被直接作为 Servlet(控制器)容器[9].

广义上来说, Portlet也是一种 Web应用程序,其运行环境也是建立在 Web服务器基础之上的, Portlet MVC模式的体现如图 2所示. 但与 Struts框架不同的是, Web服务器不能被直接作为 Portlet容器, Portlet必须另外配置容器才能运行. Portlet只有在被部署到 Portlet容器当中,与其他 Portlets一起调用才能显示其功能[11].

Portlet容器的功能是,用来管理 Portlet的生命周期并且提供其运行所需的必要环境,为 Portlet的参数选择提供持久性存取服务. 与 Servlet容器不同的是, Portlet容器不是作为一个独立运行的容器来实现的,但是从某种意义上来说,它是在 Servlet容器之上的一层,它重用了 Servlet容器提供的许多功能,例如, HTTP请求的处理、Web应用程序上下文处理、类加载器、session管理、安全等[12],这就为 Struts框架和 Portlet技术相结合提供了基础.

综上分析, Struts框架和 Portlet是 MVC设计模式的两种不同实现. 由于运行环境不同,二者之间在实现原理上存在一些差异,例如在视图层接收响应时, Portlet不同于 Struts框架,它要求控制层返回的是 fragment标记,视图层通过这些标记来显示相应的 JSP页面. 但是,相同的设计模式决定了二者之间存在着紧密的关联性,这也是 Struts应用能够实现为 Portlet的一个重要前提.

2　一种 Struts应用的 Portlet化模型

基于 Struts框架和 Portlet的关联性分析可以得知,虽然二者都是基于 MVC设计模式的 Web应用,但是二者控制层的实现方式是不同的. 基于 Struts框架的纯粹的 Servlet很难仅仅支持或者理解一个单独的请求事件. 而对于一个 Portlet来说,请求的处理必须在两个各自的 action或者 render事件中处理. 更进一步来说,在 action请求事件期间 Portlet不允许 Servlet访问. 这也正是 Struts应用 Portlet化的难点所在.

针对这一问题,本文的解决方案是,根据面向对象的原则,将原有 Struts应用的控制层封装成为能够被 Portlet容器所直接调用的接口,这样通过这一接口,就可以将 Struts应用实现为 Portlet. 而且由于该接口本身是按照 Portlet的标准来实现的,也就消除了 Struts框架和 Portlet标准的不兼容性. 基于这种思想,本文提出的门户集成 Struts应用的 Portlet化模型如图 3所示.

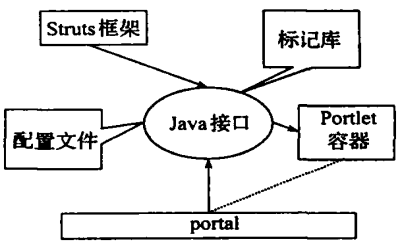

图 3　Struts应用的 Portlet化模型

Fig. 3　The model of turning a Struts application into a Portlet

本模型的核心部件是一个 Java接口,它允许一个 action请求事件映射到下层的 Servlet处理器的执行上 (例如,一个 Struts Action),从而为 Portlet访问 Struts应用中的 ServletContext提供了一个途径. 这样就能够为一个指定的 Portlet容器连接指定的 Struts应用.

除了一个封装好的 Java接口以外,基于 Struts框架的应用要能作为一个可以集成到 portal中的 Portlet来运行,还必须根据 portal的特性采用 XML语言来部署其相应的显示标记. 另外,由于 Portlet URL具有特定的格式,为使 Struts应用程序的 JSP适应门户服务器所期待的创建 Portlet URL的方式,需要将 Struts标记和

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

Portlet标记结合使用.

这样,将 Java接口、XML配置文件以及标记库三者结合起来,就可以将一个基于 Struts框架的应用 Portlet化,并最终作为 Portlet在 portal中运行.

3 Struts应用的 Portlet化模型实现

通过 Struts应用的 Portlet化模型,可以将任意一个遵循 Struts 1.1标准的应用实现为一个 Portlet,具体实现框图如图 4所示.在原有 Struts框架的基础上,本文用 ServletContextProvider API来封装 Struts框架中控制层的操作.并且添加了 B C两个配置文件,它们分别是 Portlet部署描述符和 Struts-Portlet部署描述符: portlet.xml用于部署 Portlet自身的相应信息,是每个 Portlet必须配置的; struts-portlet-config.xml为 Portlet url进行配置,对 Struts标记产生的默认值进行修改.

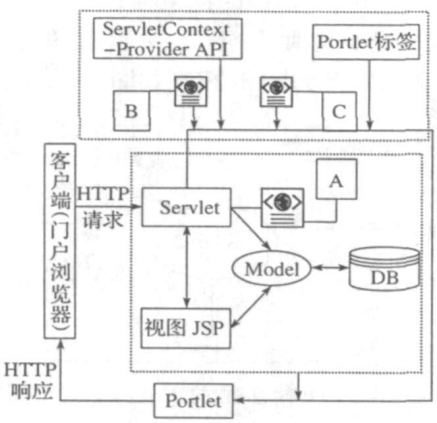

图 4 Struts应用的 Portlet化模型实现框图

Fig. 4 The implementation of the Portlet model turning a Struts application

根据图 4中的实现方式,本文将一个基于 Struts 1.1的应用所构建的客户关系管理系统进行了 Portlet化,具体实现步骤如下:

① 编写 ServletContextProvider API并通过修改 Struts应用的 struts-config.xml文件来定制该接口中的 Request Processor,将〈controller〉元素中的 processorClass 指向 ServletContext Provider. PortletRequestProcessor. 这样当 web.xml定义了一个请求确定了应该向何处转发后,就由 ServletContext Provider API替代了原有的 Struts控制器来记录所有请求与动作的映射关系了.

② 创建 struts-portlet-config.xml文件. 此文件定义了 portlet-url-type元素中 3种不同的子元素,即 action render和 resource,此文件能够指定它们的路径属性和哪种特定 urls的前缀相匹配.

③ portlet.xml文件是 Portlet的部署描述文件,此文件包括该 portlet application中一个或多个 Portlet的定义,主要记录了 Portlet自身的一些信息,例如程序作为 Portlet的入口以及 Portlet的显示名称、描述、模式和所使用到的 portlet class等.

这样,一个基于 Struts框架的客户关系管理系统就可以作为一个 Portlet在任意一个遵循 JSR168的门户中运行了.其实现效果如图 5所示,其中图(a)为 Portlet化之前的 Struts应用,图(b)是 Portlet化之后运行在 portal中的效果.

(a) (b)

图 5 基于 Struts框架的客户关系管理系统

Fig. 5 CRM system based on the Struts framework

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3 结　语

随着企业信息化的日益发展,企业应用集成已经越来越重要,而在这个过程中企业所面临的最直接的问题就是如何能够提高原有系统的重用性. 本文提出的 Struts应用的 Portlet化模型正是为了满足这样一种需求,本模型通过 Java语言和 XML语言结合使用,为将已经存在的基于 Struts框架的应用集成到门户中提供了一个解决方案,从而提高了现有系统的可重用性. 此模型已经在实际应用中进行了验证,结果表明任意一个遵循 Struts 1.1的应用均能够通过此模型实现为一个可以集成到门户中的 Portlet.

参考文献:

[1] DÍAZ O, PAZ I. Invoking web applications from portals: customization implications [C]// **Adaptive Hypermedia and Adaptive Web-based Systems Third International Conference, AH 2004**. Berlin: Springer, 2004

[2] DÍAZ O, PAZ I. Turning web applications into portlets: raising the issues[C]// **2005 IEEE/IPSJ International Symposium on Applications and the Internet**. Trento: IEEE Computer Society, 2005

[3] BELLAS F. Standards for second-generation portals [J]. **IEEE Internet Computing**, 2004, **8**(2): 54-60

[4] DÍAZ O, RODRÍGUEZ J J. Portlets as web components: an introduction [J]. **J Universal Comput Sci**, 2004, **10**(4): 454-472

[5] CHRSTIAN K M, BURGHARDT M, HAGENHOFF S. Web services for Portlet integration into corporate portals [C]// **Proceedings of the International Symposium on Web Services & Applications**. Las Vegas: CSREA Press, 2004

[6] MORAGA M A, CALERO C, PAZ I, *et al*. A reusability model for Portlets [C]// **Web Information Systems Engineering - WISE 2005 Workshops**. New York: Springer, 2005

[7] DÍAZ O, ITURRIOZ J, IRASTORZA A. Improving portlet interoperability through deep annotation [C]// **Proceedings of the 14th International Conference on World Wide Web**. Chiba: ACM, 2005

[8] TRUJILLO S, PAZ I, DÍAZ O. Enhancing decoupling in Portlet implementation [C]// **Web Engineering - 4th International Conference**. Munich: Springer, 2004

[9] Apache. Jakarta Struts [EB/OL]. [2005-10-16] http://jakarta.apache.org/struts/

[10] CHRISTIAN W. Portal server technology [J]. **IEEE Internet Computing**, 2002, **6**(3): 73-77

[11] STEFAN H. The Java Portlet specification [J]. **it-Information Technol**, 2004, **46**(5): 233-244

[12] Java Community Process. JSR168 Portlet Specification1.0 [S]. CA: [s.n], 2003-10-7

Research on model of turning a Struts application into a Portlet during portal integration

ZHAO Narisa*, GUO Yu-peng, JIA Yun-lei

(School of Manage., Dalian Univ. of Technol., Dalian 116024, China)

Abstract The primary issue of portal integration is migrating an existing application to the portal on the condition of modifying the source code as less as possible. Focusing on the issue that how to integrate a Web application based on Struts framework into the portal, the operational mechanism of the Struts framework and Portlet and the relationship between them are analyzed. Then a model, which can turn an existing Struts application into a Portlet, is proposed and implemented. It is proved that an application based on Struts standards can be turned into a Portlet with the model, and the Portlet can be integrated into the portal as a component, thus reusing ability of the existing systems is improved effectively.

Key words portal integration; Portlet; Struts; MVC

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net