基于 Spring 容器的 JBPM 流程文件部署模型①

范　菁　吕赛辉　熊丽荣 (浙江工业大学 软件学院 浙江 杭州 310023)

摘　要： 随着软件开发技术的不断更新，工作流系统越来越多地应用到企业业务流程当中。Spring 技术的成熟，使得如何高效地把 JBPM 框架整合进 Spring 框架成为了整个系统搭建的关键。在分析现有部署模型的基础上，结合流程构件开发中遇到的流程文件部署问题，提出了一种利用服务器启动参数控制部署信息并结合 SpringModules 反转控制部署的部署模型，这个模型使流程文件部署独立于 J2EE 系统，对 J2EE 框架的集成更加松耦合。

关键词： 工作流；JBPM；Spring

A JBPM Process File Deployment Model Based on Spring Container

FAN Jing, LV Sai-Hui, XIONG Li-Rong

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: With the development of software technology, work flow system is being used more and more frequently in the business process of enterprises. How to integrate JBPM into the Spring framework efficiently becomes one of the key points in the establishment of whole system's architecture. Based on analyzing the existing model of deployment and problems of process file deployment in the development of process component, a new deployment model is proposed, in which the deploy process is controlled by server startup parameters combining with SpringModules's inverse control. This new model makes the deployment of process file independent of J2EE system and provides a more loosely coupled integration.

Keywords: work flow; JBPM; spring

1　引言

工作流系统可以使企业业务流程处理自动化，目前工作流系统在国内的使用越来越普遍，这使越来越多的人开始研究如何搭建高效的开发模型，从而使工作流系统可维护性、可移植性、可扩展性等方面得到保障。一个高效的开发模型对工作流系统的开发至关重要。本文基于 Spring2.5 对 JBPM3.2.2 流程文件的部署模型进行研究。

JBPM3.2.X 为 Eclipse 提供了一个可视化开发插件，这在一定程度上简化了基于 JBPM 的工作流系统开发，特别是对流程文件的开发和部署[1]。但是这一切的简化开发都是基于可视化开发插件建立的工作流系统开发框架。但是在一般的企业系统应用当中，并不是整个企业系统都建立在工作流引擎上面，而只是其中某些企业业务流程需要工作流引擎的支持。所以在开发整个企业系统的时候，开发框架的建立以及各模块的集成是面向整个企业系统，因此需要考虑怎样把 JBPM 模块整合进现有的开发框架中去。在这种情况下可视化插件提供的自动部署流程文件功能就变的不可用。所以我们就需要一种新的流程文件部署模型来支持工作流系统的开发。

文献[1]提供了 API 和可视化部署，缺点是 API 部署在项目中通过硬编码实现，对不同的工作流系统代码移植性比较差，并且通过 API 把流程定义文件部署进入数据库，需要 JbpmContext 的事务控制支持，但是 JbpmContext 事务控制也需要硬编码来实现，这在一

① 基金项目:浙江省科技厅项目(2007c21011)
收稿时间:2009-05-19

定程度上降低了代码的可维护性、可移植性、可扩展性。可视化部署已经在本文前面说明不适合企业工作流模块作为一个开发子模块嵌入企业级系统开发项目中。

文献[2]提出了可视化部署，原则上跟文献[1]提出的思想一样，没有提供灵活的部署模型。文献[3]提出了结合 SpringModules 来整合 JBPM 和 Spring，但是没有提供灵活的配置模块,并且没有跟MVC框架集成。

本文分析考虑开发者在开发过程中需要频繁部署修改过的流程定义文件，并且每次通过重启服务器来测试项目时都需要考虑是否加载新的流程定义文件，结合这样的需要，提出一种通过继承MVC 核心控制类来引导服务器启动时的初始化工作，使其在初始化时通过 Spring 容器来读取资源文件的控制部署信息，同时结合 SpringModules 反转控制 JbpmContext 来部署流程定义文件的部署模型，该部署模型在服务器启动时结合 MVC 框架，灵活配置部署参数，实现了灵活的流程定义文件部署模型，充分满足了开发者的部署需要。

2 工作流系统

工作流系统是为解决企业业务流程分布性、重用性、普遍性[4]等特点而开发的软件包，它以形式化的流程定义文件作为输入，维护流程的运行顺序，并在使用者和系统之间分派任务。工作流系统管理业务工作流，需要计算机化的工作流模型支持。工作流模型通常包含三个子模型[5]，分别是过程模型，资源模型和组织模型。过程模型描述经营过程中的活动以及活动之间的关系，资源模型描述活动所需要的软硬件资源，而组织模型则描述活动的执行实体。其中过程模型是工作流模型的核心，主要的过程建模语言有 Carl Adam Petri 博士定义的 Petri 网[6]，美国 KBSI (knowledge –based system)公司推出的 IDEF[7], SAP 联合德国 Saarland 大学推出的 ARIS[8], BPMI 组织推出的 BPML(Business Process Modeling Language)[9],IBM、微软、BEA 公司联合推出的 BPEL4WS (Business Process Execution Language for Web Services)[10], W3C 组织推出 ebXML(与 BPEL4WS 类似)[11]，还有工作流管理组织推出的 XPDL[12]等，另外，很多人把 OMG 组织推出的 UML 语言的活动图也看成是一种业务过程建模语言。以上都是业界开发的标准，不同语言的侧重点不一样，各拥有数量不等的支持者，但没有一个标准能够解决所有的问题，例如 Petri 网有着严格的数学模型，可以用于企业业务流程的仿真。但 Petri 本身很抽象，不容易理解，很难直接用于对企业业务过程建模[13]。

JBPM 的过程建模采用的是改良的 UML 活动图[1]。做了两点改进，一是在用 UML 活动图表述业务流程时，只建模状态层(状态和控制流)，不包括动作(UML 活动图没有区分状态和动作，它们都用活动来表示)。二是如果多个迁移到达一个状态，缺省定义为不需要同步的合并(UML 活动图中默认是需要同步的联合)。JBPM 定义了自己的 XML 格式流程定义语言 JPDL，用它来精确描述 UML 活动图的每一个部分[13] 。工作流管理联盟(WFMC)[14]给出的工作流管理系统图如图 1：

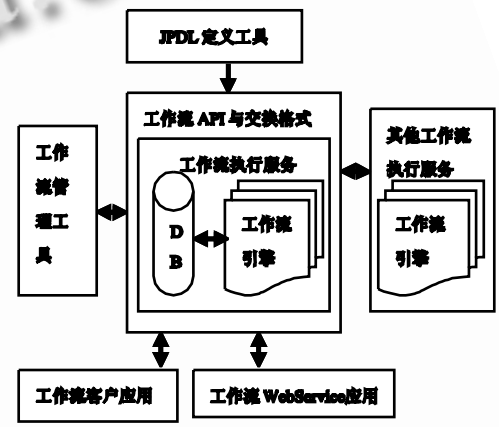

图 1 工作流管理系统图

JPDL 文件作为整个工作流引擎的唯一资源文件输入，整个工作流引擎围绕着 JPDL 文件执行业务流程、根据 JPDL 文件和其他工作流引擎进行交互以及根据 JDPL 定义的流程提供一些应用。在引言部分本文分析了基于 JBPM 部署 JPDL 流程定义文件的不足，且在分析开发者遇到的流程定义文件部署问题的基础上，提出了基于 Spring 和 SpringModules 框架的部署模型，并结合灵活的部署控制模块形成插件式部署模型。

3 Spring/SpringModules框架

Spring 是一个轻量级的 IOC 和 AOP 容器框架[15]。容器的主要功能有反向控制和面向切面编程两方面。

反向控制：Spring 提倡使用反向控制(IOC)来实现松耦合。实体对象通过 IOC 控制是被动接收依赖类而不是自己主动去找，也可以将 IOC 理解为 JNDI 的反转——对象不是从容器中查找它的依赖类，而是容器在实例化对象的时候主动将它的依赖类注入给它。

面向切面：Spring 对面向切面编程提供了强大的支持，通过将业务逻辑从系统服务(如监控和事务管理)中分离出来，以实现内聚开发。系统对象只做它们该做的业务逻辑，它们不负责其他的系统问题(如日志和事务支持)。

Spring 通过简单的组件配置就可以组合成一个复杂的系统。Spring 容器中对象是通过 XML 文件配置组合起来的。并且 Spring 提供了很多基础功能(事务管理，持久层集成等)，这使开发人员能够专注于开发应用逻辑。Spring 作为一个容器，包含并管理系统对象的生命周期和配置。开发者可以通过配置来设置系统的 Bean 类型，是单一实例还是每次请求产生一个实例，并且设置实例之间的关联关系。同时，Spring 容器没有像重量级 EJB 容器那样庞大、笨重。所有 Spring 的这些特性使开发者的代码更加清晰，更容易管理，更容易测试，这也为在 Spring 框架下开发各种子框架打下了很好的基础。

JBPM 引擎没有直接支持 Spring，需要第三方插件 SpringModules[16]支持。通过 SpringModules 的作用，可以使 JBPM 的 ActionHandler 类交给 Spring 容器管理。同时，在处理 JBPM 事务方面，它同样借鉴了 Spring 的 AOP 事务管理策略，使得 JBPM 内部的 JbpmConfiguration 配置也是通过 Spring 容器作为单例模式来管理。Spring、JBPM、Springmodules 三者之间的关系如图 2：

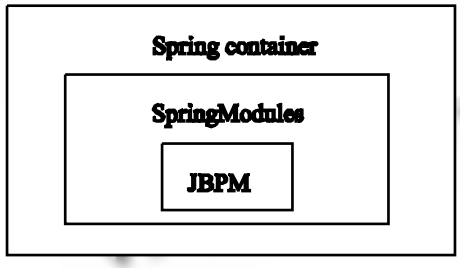

图 2　框架交互图

4　流程文件部署模型

流程定义文件一般包含流程定义语言定义的业务流程文件、流程图的图形表示文件以及用来描述流程图文件中节点位置信息的文件。它是企业业务流程的形式化描述，定义了整个工作流引擎运行业务流程的顺序。流程定义文件是联系企业业务流程和工作流引擎的桥梁，如何把工作流化的企业业务流程定义文件部署到数据库是相当重要的。是否正确部署关系到流程是否能够顺利执行、流程是否能够可视化监控以及流程图是否能够正确显示当前流程的执行进度。本文在结合 Spring 和 SpringModules 的基础上对 JBPM 提出了一个插件式的流程文件部署模块。图 3 是流程文件部署模型图：

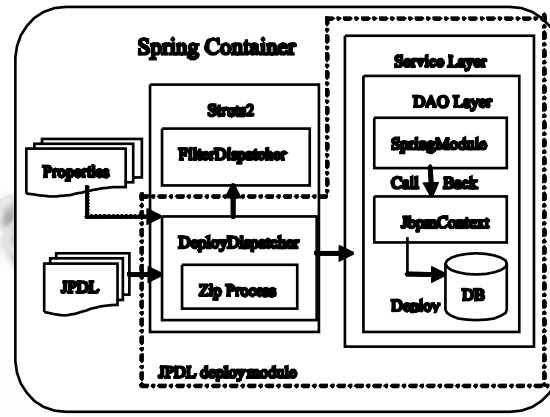

图 3　部署模型图

部署流程图中虚线框所示为流程定义文件部署模块，本部署模块可以直接打包成 jar 文件用于其他基于 Spring 和 JBPM 的工作流项目。首先在开发项目 Properties 文件中设置数据库连接信息和流程定义文件部署控制信息，同时在 Properties 文件中指定流程定义文件所在的位置信息，通过 MVC 控制器 DeployDispatcher 对流程定义文件进行 ZIP 压缩处理，并通过事务控制的 service 层来调用 DAO 层，以实现 DAO 层调用 SpringModules，使其反向控制 JbpmContext，最后通过 JbpmContext 实现将流程定义文件部署进入数据库。

Properties 文件用来定义流程文件的一些部署信息，例如流程文件所在位置信息和流程文件打包压缩成什么样的文件名进行部署以及是否在服务器启动时重新部署流程定义文件等。通过 Properties 文件部署模块可以非常方便地控制流程文件部署操作，相比较通过硬编码来实现控制部署操作，部署模块在很大程度上跟已有企业项目开发框架和代码级上进行了解耦。

JPDL 文件包含三个文件：processdefinition.xml、processimage.jpg 、gpd.xml。processdefinition.xml 是业务流程定义文件，主要用于定义企业业务执行顺序、参与业务流程管理的角色权限和一

些触发管理操作。processimage.jpg 文件是流程图的 JPG 格式图形展示，主要用于流程图进度显示和流程监控功能。gpd.xml 文件记录 processimage.jpg 文件中节点位置信息，用于流程图进度实时显示。

本文通过定义 DeployDispatcher 类继承 Struts2 中 MVC 控制器 FilterDispatcher 来实现流程定义文件的部署控制。DeployDispatcher 类通过读取 Properties 文件控制参数可以动态控制是否在服务器启动时部署流程定义文件。JPDL 文件通过 ZIP 压缩后由 DeployDispatcher 调度给 Service 层处理。Service 层主要通过 Spring AOP 来控制部署事务操作，由于压缩后的 JPDL 文件部署进入数据库需要事务原子性支持，而 Spring AOP 通过面向切面编程思想用 XML 配置文件在 service 层控制事务操作，使模块功能分层清晰，并避免了在数据库操作代码中嵌入事务控制代码，进一步提升了模块的耦合性。

Service 层用于控制事务属性，而 DAO 层用于数据库直接部署操作，这样的部署设计方案对以后部署模块的可扩展性和鲁棒性有很好的保障，因为如果以后考虑要使用不同的数据库来支撑工作流系统项目开发。由于本部署模型的事务控制通过 Spring AOP 在 service 层进行控制，并没有直接在 DAO 实现类里面进行硬编码，因此在切换数据库时不需考虑 service 层的变更操作，且 DAO 层只是面向接口编程的一个实现，所以也不需要相应的变更，唯独需要修改的部分是 DAO 层实现类，因此，只需要根据不同的持久层和不同的数据库修改相应的接口就可以了。DAO 层用 SpringModules 第三方插件来实现 Spring 和 JBPM 之间的交互，通过 SpringModules 反转控制 JbpmContext 来实现 JbpmContext 对流程定义文件的部署操作。这种通过反转控制设计灵活覆盖 JBPM 类原有实现方法，它比通过自定义类继承 JBPM 原有类，再覆盖其方法更加简便，并且减少了大量冗余代码，降低了维护成本。

5 信用卡申请流程应用

由于信用卡申请流程业务不断完善，并且其中伴随着申请流程的变更，这时候银行开发部门需要不断部署新的业务流程。本文提出的部署模型可以使银行开发部门实现自动化部署操作。

在本应用中，首先用 JBPM 流程设计器插件对信用卡申请流程进行定义，三个流程定义文件为 processdefinition.xml、processimage.jpg、gpd. xml。其中 processimage.jpg 信用卡申请流程图如图 4：

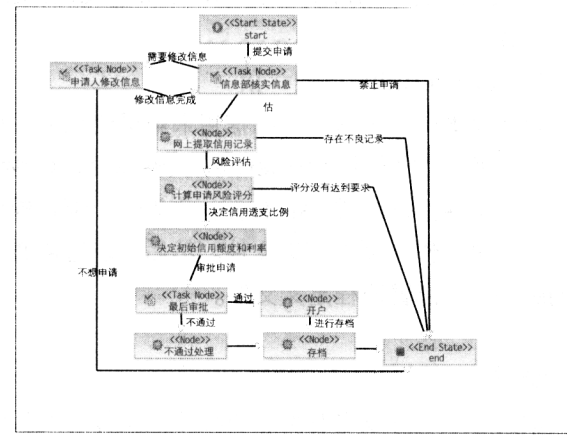

图 4 信用卡申请流程图

其中用于控制部署流程文件的 properties 的文件截图如图 5：

```
16 ##for deploy
17 deploy.processDefinition.at.start.server=true
18 zip.processDefinition.dir=D:\CreditApply\process\creditcard\
19 processDefinition.name=process.zip
```

图 5 Properties 文件截图

流程定义文件部署模块部署步骤如下：

第一步：用 ZIP 压缩程序对流程定义文件：jpd.xml,processdefinition.xml,processimage.jpg 进行压缩。

第二步：DeployDispather 通过 Spring 容器对 Properties 文件读取部署信息，然后将压缩后的流程定义文件交给 service 层方法 deployProcessDefinitionByZip(zipAddress)处理。

第三步：service 层将压缩后的流程文件进行适配器模块包装成 ProcessDefinition 对象 ，然后通过调用 DAO 层部署接口方法 deployProcessDefinition() 对刚刚包装的 ProcessDefinition 对象通过 SpringModules 反转控制 JbpmContext 来实现 JbpmContext 对流程定义文件进行的部署操作。部署部分代码如下：

public void deployProcessDefinition(final

```
ProcessDefinition processDefinition) {
getTemplate().execute(new JbpmCallback() {
public Object doInJbpm(JbpmContext context){
context.deployProcessDefinition(processDe
finition);
return null;  }  });  }
```

以上是本流程部署模块在信用卡申请流程上的应用，除流程定义文件和 Properties 文件以外，其他参与部署的文件都打包成 jar 文件，作为一个部署模块插件的部署工作流流程定义文件。然后可以直接通过启动服务器来部署流程定义文件。

6 结语

本文通过对开发者在部署流程定义文件时遇到的问题的分析，提出了一种通过继承 MVC 核心控制类来引导服务器启动时的初始化工作，并使其在初始化时通过 Spring 容器来读取资源文件的控制部署信息，同时结合 SpringModules 反转控制 JbpmContext 来部署流程定义文件的部署模型。通过信用卡申请流程构件开发和其他流程构件开发的验证，此部署模型体现出良好的可管理性、可扩展性和可维护性。由于 JBPM 不直接支持 JMS，以及 JBPM 任务实例实时监控不够完善，今后将主要面向这两个方面展开研究。

参考文献

1 JBOSS, JBPM. http://www.jboss. com/products/ jbpm/.

2 Cumberlidge M. Business Process Management with JBoss JBPM.Birmingham:Packt Publishing. 2007.114－116.

3 Harrop R, Devijver S, Leau C. Spring Modules Reference Documentation. 2007.79－82.

4 邝磊.基于 jBPM 的 BPM 系统的研究与设计实现[硕士学位论文].广州:中山大学, 2007.

5 范玉顺.工作流管理技术基础,北京:清华大学出版社, 2001.100－115.

6 Petri CA. http://www.scholarpedia. org/ article/ Petri_net.

7 IDEF. http://www.idef.com/.

8 ARIS. http://www.ids-scheer.com/en/ARIS/ARIS_ Software/3730 .html.

9 BPML. http://www.bpmi.org/.

10 BPEL4WS. http://www.ibm.com/developerworks/ library/specification/ws-bpel/.

11 ebXML. http://www.ebxml.org/.

12 XPDL. http://www.wfmc.org/xpdl.html .

13 王宇明.JBPM,工作流项目的研究与实践[硕士学位论文].天津:天津大学,2005.

14 WFMC. http://www.wfmc.org/reference-model. html.

Craig Walls, Ryan Breidenbach. Spring in Action. Connecticut: Manning Publications.2008.5－125.15 .

SpringModules. https://springmodules.dev.java.net/.16