

# workflow 驱动的 Web 应用快速开发框架研究<sup>①</sup>

## Research on Rapid Web Application Development Framework Driven by Workflow

马 华 (湖南涉外经济学院计算机学部 湖南长沙 410205)  
张红宇 (中南大学商学院 湖南长沙 410083)  
陈 振 (湖南涉外经济学院计算机学部 湖南长沙 410205)

**摘 要:** Web 应用的开发周期越来越短,需求的快速变化要求 Web 应用能够实现持续改进和功能增强,从而产生了对 Web 应用快速开发框架的迫切需求。针对此,基于 workflow 驱动的 Web 应用模型,提出了一种 Web 应用快速开发框架 WADF-Workflow。该框架支持遵循 WfMC 工作流规范的业务流建模,并支持基于 workflow 引擎的页面流控制、数据流控制和业务逻辑处理的分离。最后,给出了基于开源 workflow Shark 和 JaWE 的框架实现和应用实例。

**关键词:** workflow Web 应用 快速开发框架 Shark JaWE

### 1 引言

当前,随着市场竞争越来越激烈,Web 应用的开发周期正变得越来越短,需求的快速变化要求应用系统必须具备“持续改进和功能增强”的基本特点<sup>[1]</sup>,以满足建立“实时企业”、“实时政府”的迫切需求。在当前以 Web 为中心的软件开发模式下,Web 应用快速开发框架的研究成为热点。Struts、Spring 和 Hibernate 等应用框架的出现,为解决 Web 应用开发中的一些问题提供了支持,为研究一种有效的整体性解决方案提供了基础。

针对此,提出了一种 workflow 驱动的 Web 应用快速开发框架 WADF-Workflow,该框架支持遵循 workflow 管理联盟(WfMC)相关规范的可视化业务流建模,支持基于 workflow 引擎的页面流控制、数据流控制和业务逻辑处理的分离。从而,WADF-Workflow 可以有效地满足 Web 应用系统的“持续改进和功能增强”需求,大大地提高 Web 应用的开发效率。

### 2 workflow 驱动的 Web 应用模型

一个支持“持续改进和功能增强”的 Web 应用快

速开发框架,需要实现页面流控制、数据流控制和业务逻辑处理的分离:

(1) 页面流控制,指 Web 系统中页面之间由提交(Submit)动作引发的流转关系的映射机制。

(2) 数据流控制,指决定一个业务处理过程状态的数据(如当前登录用户的用户名和用户类型、审批通过或未通过的标识等)在不同的页面之间流动与传输的机制。

(3) 业务逻辑处理,指页面的提交处理与具体的业务对象(Business Object)间的绑定机制。

针对以上三个问题,提出了 workflow 驱动的 Web 应用模型,如图 1 所示。

该模型中,一个 Web 应用(Web Application)由若干个业务流定义(Process Definition)所构成(例如,在一个“审批系统”中,至少需要“提交申请材料”和“处理申报材料”两个流程),每个业务流定义由多个活动(Activity)组成,并拥有其 workflow 相关数据(Relevant Data)。workflow 相关数据为实现 Web 应用中的数据流控制提供了直接支持,并为活动定义转移条件(Transition),以形成分支、同步、循环等复杂的流程控制结构。

① 基金项目:①湖南省教育厅科学研究项目(07C425); 湖南省十一五规划课题(XJK06CXJ027)。

每个“活动”映射应用中的一个页面,页面的提交操作,由与该活动绑定的“业务对象”(Business Object)类来实现,同时,“后续页面”(Forward)指示了当前页面提交操作后的其它用户表现。

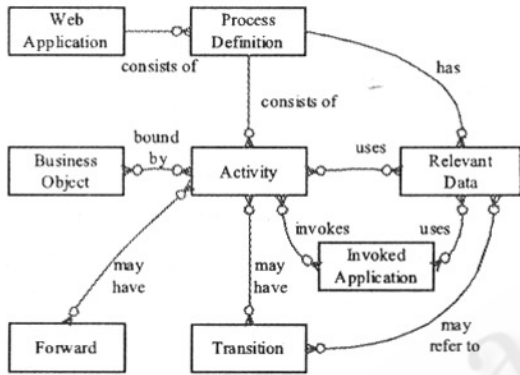


图 1 工作流驱动的 Web 应用模型

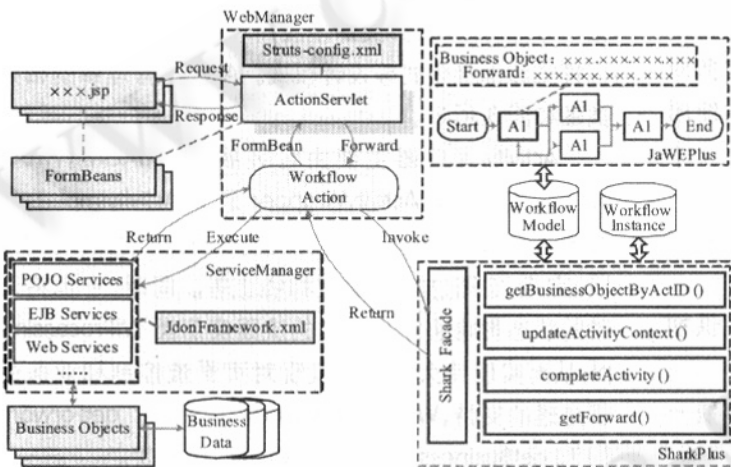


图 2 工作流驱动的 Web 应用快速开发框架

根据 WfMC 的定义<sup>[2]</sup>,一个业务流 (Process Definition) 可以以“子工作流”的形式,作为另一个流程的一部分。这样,一些常用的业务流,如异常处理等,可以封装为共享工作流,从而提高系统的重用性。

### 3 工作流驱动的 Web 应用快速开发框架

#### 3.1 总体架构

以基于面向服务架构的动态 EAI 平台<sup>[3]</sup>为基础,工作流驱动的 Web 应用快速开发框架 (WADF - Workflow) 如图 2 所示。

WADF - Workflow 主要由四大组件构成,分别是 JaWEPlus、SharkPlus、ServiceManager 和 WebManager。

(1) JaWEPlus,是一个基于开源项目 JaWE<sup>[4]</sup>扩展的、遵循 WfMC 规范<sup>[2]</sup>的工作流建模工具,它负责定义 Web 应用的业务流程,其中主要约束了每个页面的业务逻辑处理对象、Forward 页面,以及页面流中所涉及的工作流相关数据。

(2) SharkPlus,是一个基于开源项目 Shark<sup>[5]</sup>扩展的、遵循 WfMC 规范的工作流引擎,通过扩展的 Facade 接口,可以提供对页面流、数据流控制和业务逻辑处理的支持。

(3) ServiceManager,是一个基于开源项目 JdonFramework<sup>[6]</sup>实现的服务管理器,提供了对 Web 应用中各个页面相应的业务对象的管理,支持基于 Web 服务、EJB、POJO 等类型的业务对象的实现。在 JdonFramework.xml 的支持下,可以实现业务对象逻辑名和物理名的映射。

(4) WebManager,是一个基于开源项目 Struts 实现的 Web 管理器,提供了工作流驱动的 Web 应用的中枢管理。在 WorkflowAction 的调度下,实现与 SharkPlus 的 Fa? acade 接口的交互,以及对 JdonFramework 框架中的业务对象的调用。并在 Struts - config.xml 支持下,实现页面的正确流转。

#### 3.2 关键技术

(1) JaWE 和 Shark,是国际组织 Enhydra 的开源工作流项目。JaWE 是一个图形化的 XPD 建模工具。Shark 是一个完善的工作流引擎,它完全基于 WfMC 的相关标准,提供了包括工作流引擎、监控与管理工具和客户端工具等。

(2) JdonFramework,是一个适合开发中小型 J2EE 应用的、基于构件(组件)的轻量快速开发框架。它结合了 Spring 和 RoR 的优点,实现了 loc/DI 设计思想,支持 Autowiring,遵循约定优于配置 (Convention Over Configuration),是一种严格坚持分层架构的快速开发框架。

(3) Struts,提供了 MVC 模式的“分离显示逻辑和业务逻辑”的支持。基于 Struts 构架的 Web 应用基本符合 JSP Model2 的设计标准,是传统 MVC 设计模式的

一个变化类型。

### 3.3 基于 WADF - Workflow 的 Web 应用开发流程

基于 WADF - Workflow 的支持,一个新的 Web 应用的快速开发流程如下:

(1) 根据业务需求,使用可视化业务流定义工具 (JaWEPlus) 定义 Web 应用的业务功能,定义页面流、数据流和业务逻辑处理关系。

(2) 实现流程定义中各个活动 (即 Web 页面) 所依赖的 JSP、FormBeans 和业务对象 (Business Object)。

(3) 在 WebManager 的 Struts - conf. xml 中配置 Forward 的逻辑名和实现页面,在 ServiceManager 的 JdonFramework.xml 中配置 BusinessObject 的逻辑名和实现类。

(4) 启动 WADF - Workflow 框架,新的 Web 应用即可在工作流引擎 (SharkPlus) 的支持下,实现 Web 应用的页面流、数据流和业务逻辑处理的自动完成。

WADF - Workflow 框架下,通过对已有 Web 应用业务功能相关的流程定义、JSP、FormBeans 和业务对象的修改,即可实现 Web 应用的“持续改进和功能增强”。

### 3.4 基于 WADF - Workflow 的 Web 应用执行流程

WADF - Workflow 框架下,一个工作流驱动的 Web 应用系统工作过程如下:

(1) 用户登录系统,WebManager 向用户提供初始界面;

(2) 用户的任何提交请求,将被 ActionServlet 统一转发给 WorkflowAction 进行处理;

(3) WorkflowAction 通过调用 SharkPlus 的 Faade 接口 getBusinessObjectByActID(), 获取与当前页面绑定的业务逻辑处理对象名,并根据在 ServiceManager 中 JdonFramework.xml 定义的逻辑映射名,定位正确的业务对象,实现业务逻辑的处理;

(4) WorkflowAction 通过调用 SharkPlus 的相应接口,实现业务流所属工作流活动的上下文更新 (updateActivityContext()), 结束当前页面 (completeActivity()), 并获取下一页的逻辑名 (getForward());

(5) ActionServlet 根据 WorkflowAction 返回的 Forward,在 struts - config.xml 中定位下一个正确的 jsp 页面;

(6) 继续执行步骤 (2) ~ (5), 直至用户退出

系统。

## 4 框架的实现

本框架的实现,主要涉及对 Shark 和 JaWE 的扩展,以及基于 Struts 框架和 JdonFramework 框架的集成。这里简要介绍对 Shark 和 JaWE 的扩展。

### 4.1 Shark 的扩展

SharkPlus 基于 Shark1.1.2 版本进行扩展,由 XPDL 解析器、Repository 管理器、Package 管理器、Process 管理器、Activity 管理器、Assignment 管理器、流程监控管理器和 Engine 管理等组件构成。其中 Engine 管理器是核心组件,在 SharkPlus 中,该角色由 org.enhydra.shark 包下的 Shark 类来具体承担,但 Shark 类的职能又实际上几乎全部由 SharkEngineManager 类来实现。SharkPlus 继承了 Shark 原有的架构,在扩展类的设计上,遵循了开放封闭原则、依赖性倒置原则和接口分离原则等,并灵活地应用了包装器模式、抽象工厂模式和单例模式等设计模式。Activity 管理器的相关类图如图 3 所示。

Activity 管理器主要由包装器类 (WfActivityWrapper) 来实现。WfActivityWrapper 是 WfActivityImpl 的包装类,它封装了所有与活动实例状态相关的操作,如激活、完成、忽略、挂起、恢复和终止等。同时,它也将活动的状态影响通知其所属的流程实例类 WfProcessImpl,从而实现持久。为了实现对页面流控制和业务逻辑处理的支持,WfActivityWrapper 中添加了 getForward() 和 getBusinessObjectByActID() 两个方法。

### 4.2 JaWE 的扩展

JaWEPlus 基于 JaWE1.4.2 版本进行扩展,实现了对“业务对象” (BusinessObject) 和“后续页面” (Forward) 的支持。我们的扩展主要是在活动的属性定义中提供对页面流控制的相关支持。通过新增的 Tab 页实现活动的“BusinessObject”和“Forward”属性的定义。为了保持与 WfMC 的工作流建模语言 XPDL 的兼容性,新增的属性被作为 Activity 的“扩展属性”保存。其中,属性“BusinessObject”的值为业务对象的逻辑名,属性“Forward”的值为后续页面的逻辑名。

为了实现以上功能,主要扩展了以下几个类:

(1) 初始化活动的属性面板时,调用 org.enhydra.jawe.graph 包中的 showPropertyDialog() 方法,该方

法从 `ProcessEditor` 类中 `getElementEditingDialog` 方法得到 `XMLElementDialog` 实例, 然后调用 `org.enhydra.jawe.xml.elements.Activity` 中 `getPanel()` 方法向面板添加相关的 `Tab` 页。

实现原理。

该系统中, “登录页面” 活动节点, 由 `org.hieu.application.login` 进行了用户登录校验后, 会将校验结果保存在两个流程变量中, `isValid` 保存登录成功与否

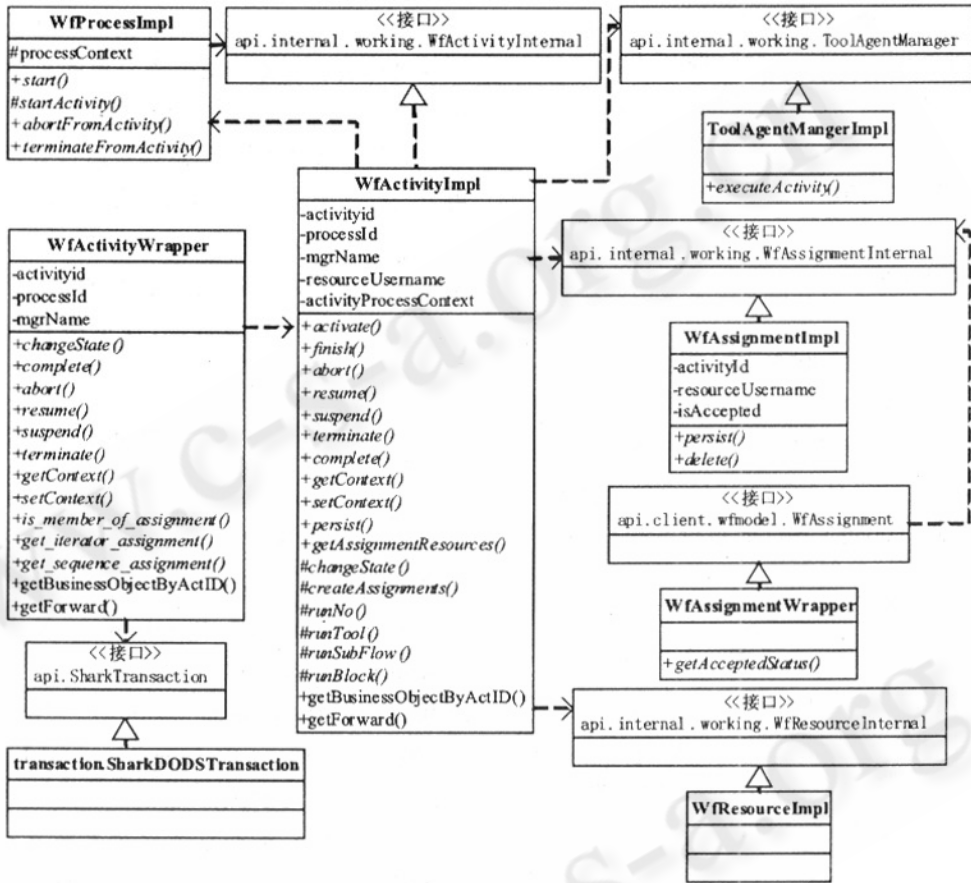


图 3 Activity 管理器类图

(2) 获取 `Activity` 中持有的 `refExtendedAttributes` 实例, 即与活动相关的扩展属性实例, 实现将新增数据添加到“扩展属性”中。

(3) `org.enhydra.jawe.xml.elements` 包的 `ExtendedAttributes` 类中, 在 `getPanel()` 方法中创建一个 `XMLWebFlowPanel` 实例, 该类是新添加的一个主要实现表单中相关操作的类。

### 5 应用实例

我们的研究在“政务审批系统”的开发中得到应用<sup>[6]</sup>。如图 4 所示, 以“中外合资企业合同、章程审批”处理流程为例, 介绍基于 `WADF - Workflow` 的系统

的标识 (yes 或 no), `userType` 保存用户类型标识 (guest 或 admin), 在下一个 `Route` 型活动“登录成功?”中, 将根据 `isValid` 的值进行判断。如果 `isValid` 为 yes, 则进入“企业用户?”判断, 根据 `userType` 的不同值, 页面分别流转到“申报材料填写” (`apply.jsp`) 或“未处理申报列表” (`newapp.jsp`)。分别实现企业用户提交申报材料和政府工作人员处理申报材料的两个不同业务流程。

“申报材料填写”活动节点的 `XPDL` 定义片断如下:

```

<Activity Id = "Application_Wor1_Act12" Name = "申报材料填写" >

```

```

<ExtendedAttributes >
<ExtendedAttribute Name = " BusinessObject"
    Value = " org. hieu. application.
submit" / >
<ExtendedAttribute Name = " Foward"
    Value = " confirm_application" / >
<ExtendedAttribute Name = " XOffset" Value = "
130" / >
<ExtendedAttribute Name = " YOffset" Value = "
30" / >
</ExtendedAttributes >
</Activity >
    
```

“申报材料填写”页面 ( apply. jsp ) 提交后, 由业务对象 ( org. hieu. application. submit ) 进行申报材料信息的校验, 将校验结果保存在工作流上下文变量中, 并将页面流转到“申报材料确认”页面 ( “confirm\_application” ), 在新的页面 ( confirm. jsp ) 显示校验结果, 并等待确认。

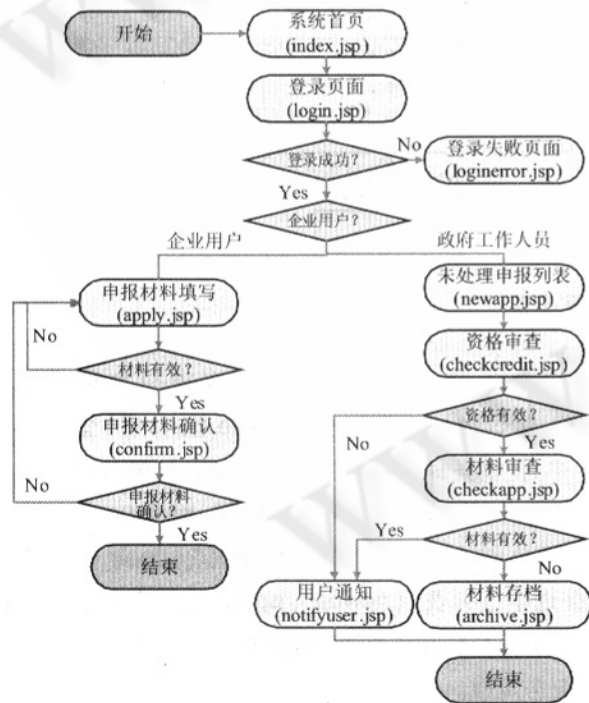


图 4 流程建模

## 6 总结

基于开源框架 Struts 和 JdonFramework, 通过扩展开源工作流项目 Shark 和 JaWE, 设计并实现了工作流驱动的 Web 应用快速开发框架 WADF - Workflow。该框架满足了 Web 应用快速开发框架对于页面流控制、数据流控制和业务逻辑处理分离的三个主要需求, 支持 Web 应用的“持续改进和功能增强”, 并遵循 WfMC 定义的工作流基本规范, 从而可以很好地解决 Web 应用中的数据绑定问题。同时, 由于其保持了与 XPDL 的兼容性, 为 Web 应用间的互操作提供了支持。该框架采用可视化页面流建模, 极大地提高了系统的开发效率。多个实际项目<sup>[7]</sup>证明, 该框架具有良好的应用前景。

### 参考文献

- 1 Rakesh Agrawal, Roberto J. Bayardo Jr., Daniel Gruhl. Vinci: A Service - Oriented Architecture for Rapid Development of Web Applications. In: Proc. of the Tenth International World Wide Web Conference, HongKong, May 2001: 355 ~ 365.
- 2 Workflow Management Coalition, " WfMC - TC00 - 1003 1996, Reference model and API specification", 1996.
- 3 马华、李建华, 面向服务架构在动态企业应用集成中的应用, 计算机工程与设计, 2006, 27 ( 13 ): 2507 ~ 2510.
- 4 ObjectWeb. Enhydra Shark. <http://shark.objectweb.org/>.
- 5 ObjectWeb. Enhydra JaWE. <http://www.enhydra.org/workflow/jawe/>.
- 6 Jdon. JdonFramework. <http://www.jdon.com/index.html>.
- 7 刘星沙、马华、夏明伟等, 支持跨区域协同的政务审查系统研究, 计算机工程与设计, 2007, 28 ( 16 ): 3981 ~ 3984.