

# 轻量级 J2EE 框架下 workflow 驱动的动态虚拟实验室<sup>①</sup>

马 华 荆 翔 邓芳洁 张文毅 (湖南涉外经济学院 计算机学部 湖南 长沙 410205)

**摘 要:** 综合考虑基于构件组装技术的虚拟实验室的系统需求,分析了 workflow 驱动的动态虚拟实验室的业务处理模型,介绍了轻量级 J2EE 框架(SSH)与 workflow 系统(Shark 和 JaWE)的集成模型,提出了一种轻量级 J2EE 框架下 workflow 驱动的动态虚拟实验室的设计和实现方法,给出了虚拟实验项目的实现机制、数据流和控制流的管理方法,以及实验流程的动态组装方法,最后,以应用实例说明了本文方法的有效性。

**关键词:** 虚拟实验室; workflow; 轻量级 J2EE 框架; SSH; Shark; JaWE

## Dynamic Virtual Laboratory Driven by Workflow Based on Lightweight J2EE Framework

MA Hua, JING Xiang, DENG Fang-Jie, ZHANG Wen-Yi

(Department of Computer, Hunan International Economics University, Changsha 410205, China)

**Abstract:** Considering the system requirements of virtual laboratory based on component composition technology, the business processing model of dynamic virtual laboratory, driven by workflow, is presented. The integration model of lightweight J2EE framework (SSH) and workflow system (Shark and JaWE) is introduced. The design and implementation method of dynamic virtual laboratory driven by workflow, based on lightweight J2EE framework, is proposed, and this paper analyzes the realization mechanism of virtual experiment project, the management approach of data and control flow, and the dynamic composition method of the experimental process. Lastly, the application case shows that this method is effective.

**Keywords:** virtual laboratory; workflow; lightweight J2EE framework; SSH; Shark; JaWE

## 1 引言

由于信息技术和网络技术的迅速发展,目前国内外出现了越来越多面向 Internet 的虚拟实验室系统<sup>[1]</sup>。在面向服务计算环境下,标准化的服务构件技术有利于实现虚拟实验教学资源的共享和重用,而 workflow 驱动的构件组装技术,使运行时虚拟实验项目的动态创建或重构成为可能<sup>[2,3]</sup>。当前,SSH(即 Struts, Spring, Hibernate)框架已经成为了企业级 Web 系统的主要开发框架<sup>[4]</sup>,它具有高内聚、低耦合、部署简单、资源占用低、廉价等优点。因此,基于 SSH 的轻量级 J2EE 框架可以成为开发 workflow

驱动的动态虚拟实验室系统的有效途径。

通过分析动态虚拟实验室的业务处理模型,以及 SSH 框架与 Shark、JaWE 的集成模型,本文介绍了一种轻量级 J2EE 框架下 workflow 驱动的动态虚拟实验室的设计和实现方法,分析了虚拟实验项目的实现机制、数据流和控制流的管理方法,以及实验流程的动态组装方法,最后,以应用实例说明了本文方法的有效性。

## 2 workflow 驱动的动态虚拟实验室业务处理模型

在基于构件组装的虚拟实验室系统<sup>[2]</sup>中,实验资源被封装为标准化服务后,应用构件组装方法创建可

<sup>①</sup> 基金项目:湖南省大学生研究性学习和创新性实验计划;湖南省教育科学规划课题研究成果(XJK08CXJ001);湖南省教育厅资助科研项目(07C425)

收稿时间:2010-02-09;收到修改稿时间:2010-03-27

重构的虚拟实验项目。这样，由 workflow 驱动的动态虚拟实验室的业务处理模型如图 1 所示。

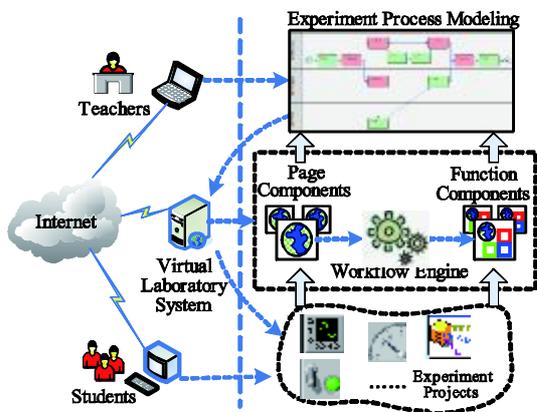


图 1 动态虚拟实验室的业务处理模型

教师通过 workflow 建模工具对虚拟实验项目 (Experiment Project) 进行建模，建模过程中页面构件 (Page Component) 和功能构件 (Function Component)，被分别定义为独立的活动 (Activity) 或路由 (Route) 节点，通过转移 (Transition) 联接这些构件，从而构建出新的实验项目。学生通过 Internet 访问虚拟实验项目时，页面的流转控制，以及虚拟实验仪器、仿真程序等资源的调用均由 workflow 引擎根据实验流程模型进行驱动。在 workflow 技术的支持下，教师可以实现运行时重构实验项目，即教师在运行时通过修改实验过程模型，从而动态改变学生参与虚拟实验过程时的操作流程和实验结果。

### 3 轻量级 J2EE 框架与 workflow 系统的集成

#### 3.1 SSH 与 Shark、JaWE

SSH<sup>[4]</sup> 即 Struts、Spring、Hibernate 开源框架。其中，Struts 是基于 MVC (Model-View-Control) 设计模式的 Web 开发框架。Spring 是重量级 EJB 的替代品，基于反转控制 (IoC) 和面向方面编程 (AOP) 技术，实现了依赖性规范和实际应用程序的代码分离。Hibernate 是对象关系映射 (Object Relational Mapping, ORM) 框架，完成对象和关系之间的平滑映射。

Shark<sup>[5]</sup>、JaWE<sup>[6]</sup> 是由 Enhydra 开发的开源 workflow 管理系统，Shark 是 workflow 引擎，JaWE 是图形化建模工具，它们遵循 WfMC (Workflow Management Coalition) 相关规范，使用 XPDL (XML Process

Definition Language)<sup>[7]</sup> 作为流程建模语言。

“ToolAgent” 机制支持调用服务化的外部构件资源。

#### 3.2 SSH 与 Shark、JaWE 的集成模型

SSH 与 Shark、JaWE 的集成模型如图 2 所示。表现层采用 Struts 框架，实现用户交互，调用 Action 处理用户请求。通过检查实验过程模型 XPDL，workflow 引擎驱动底层的功能构件。业务逻辑层采用 Spring 框架，负责业务层的管理、集成 Hibernate、事务管理、日志拦截、数据校验等。数据持久层用 Hibernate 完成数据库操作。JaWE 用于实验过程建模，通过 XPDL 中 Activity 的扩展属性 (Extended Attributes) 定义页面构件或功能构件的 URL 调用地址。workflow 引擎 Shark 负责运行动态解析实验过程模型，以指导页面流转，并通过 “ToolAgent” 接口调用外部的功能构件。

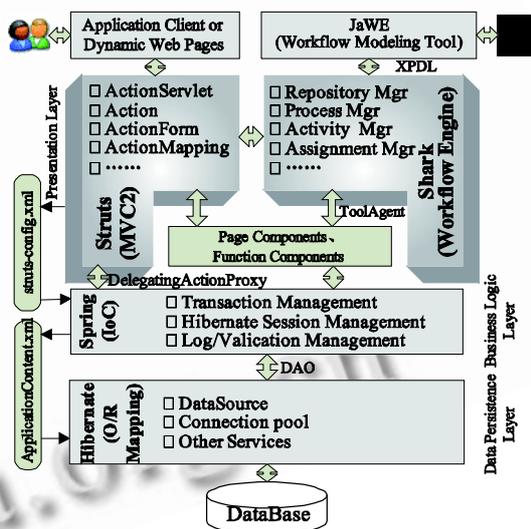


图 2 SSH 与 Shark、JaWE 的集成模型

### 4 workflow 驱动的虚拟实验室系统的实现

#### 4.1 虚拟实验项目的实现机制

一个虚拟实验项目的实现机制如图 3 所示。

表现层的 “页面构件” 以 JSP 开发，支持用户交互操作，传递用户提交的请求数据并接收返回的响应结果。Struts 框架接收用户请求后，由 ActionForm 封装请求的数据以及 XPDL 中的 workflow 相关数据，再传入由 Spring 管理的 Action 实例，由业务逻辑层 Spring 的 IoC 容器进行事务处理、日志管理等操作，在 Action 中驱动 Shark 引擎，解析 XPDL、调用 “功能构件”，从而完成实验运算。处于数据持久层的

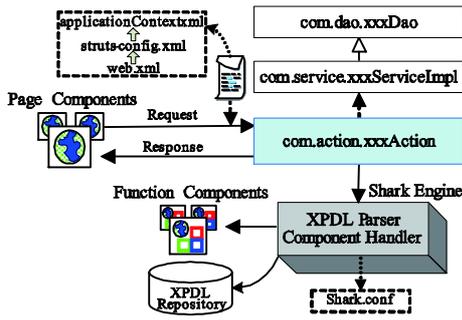


图3 虚拟实验项目的实现机制

Hibernate 处理来自业务逻辑层的 DAO 组件的请求，完成与数据库交互。同样，返回的结果将根据 XPD 的业务组装规则，向用户作出相应响应。对于来自系统中非“功能构件”处理的页面请求(如帐户管理、登录验证、实验项目管理等)，则由 Action 调用相应的服务、请求 DAO 组件进行处理。

为了实现 Spring 与 Struts 的集成，我们首先要把 Struts 的动作管理委托给 Spring，将 Spring 作为插件加到 struts-config.xml，其相关定义片断如下所示：

```
<struts-config>
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
<set-property property="contextConfigLocation"
value="/WEB-INF/applicationContext.xml" />
</plug-in>
</struts-config>
```

Spring 和 Struts 的集成采用 Delegating ActionProxy 方式，其相关定义片断示例如下：

```
<form-bean name="xxxForm" type="com.form.xxxForm"></form-bean>
.....
<action path="/xxx" parameter="method" name="xxxForm"
type="org.springframework.web.struts.DelegatingActionProxy">
<forward name="xxx" path="xxx.jsp"></forward>
</action>
.....
```

通过将截获的请求传入 Spring 管理的 Action 实例，从而实现页面层到服务层的衔接。在 applicationContext.xml 中配置 Struts 的 action，定义片断如下所示：

```
<bean name="/xxx" class="com.action.xxxAction" singleton="false">
<property name=".....">.....</property>
</bean>
```

这样 Spring 中的 DelegatingActionProxy 通过查找上下文配置中定义的 Action，执行其 execute 方法。以“操作系统”课程“进程调度”虚拟实验 Action 为例，其定义片断如下：

```
public ActionForward execute(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response) {
.....//获取当前实验项目的配置信息
SharkService ss=new SharkService();//初始化Shark引擎服务实例
String fC=ss.getFunctionComponent(strProj)//根据实验项目信息获取FC地址
Service service = new Service();//实例化axis的一个服务
Call call = (Call) service.createCall();
call.setTargetEndpointAddress(new java.net.URL(fC));//设置Web服务地址
.....//完成实验参数的设置、调用功能构件并获取实验结果
}
```

对于其它非“功能构件”处理的普通页面，如“实验列表管理”等，其相应的 Action 片断示例如下：

```
public ActionForward execute(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response) {
ExperimentServiceImpl esi = new ExperimentServiceImpl ();//实例化实验服务类
List list = esi.getAllInfo();//从ExperimentDao中获取全部实验项目信息
if(list==null) list = new ArrayList();
HttpSession session = request.getSession();
session.setAttribute("ExperimentQueryInfoO", list);//存入session
return new ActionForward("/Pages/admin/ExperimentQuery.jsp");//转发处理
}
```

由于 Spring 对 Hibernate 的 SessionFactory、事务管理进行了封装，可以通过 ApplicationContext 管理 SessionFactory，随应用启动时自动加载。SessionFactory 可被由 ApplicationContext 管理的任意一个 Bean 引用，如 DAO，Spring 可为 DAO 对象注入 SessionFactory 引用，从而可以使用 IoC 模式实现对持久层的统一管理。

#### 4.2 数据流和控制流的管理方法

伴随学生与页面构件的交互过程，实验输入数据和实验结果数据在实验过程的各个步骤中不断流动，即存在数据流。学生输入的不同实验输入数据将可能触发不同的实验操作流程，即实际运行过程中同一个实验可能出现多种实验步骤的组合，即存在控制流。本文提出的动态虚拟实验室系统中，将每一个实验步骤描述为工作流中的一个活动(Activity)，通过工作流 Activity 来管理实验过程中数据流和控制流。工作流驱动下的实验过程可以描述为图 4 所示。

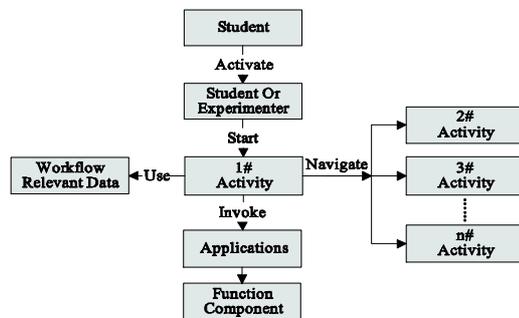


图4 工作流驱动下的实验过程

学生实验时，在工作流的驱动下每一个 Activity

调用相应的外部应用程序(Applications),学生交互的数据以及与实验相关的数据被映射成 workflow 相关数据 (Workflow Relevant Data), 根据转移条件 (Transition Condition), 实现数据流与控制流的导航。其中, workflow 相关数据和转移条件在流程定义文件 XPDL 中建模, Struts 中的 ActionForm 负责封装 workflow 相关数据, workflow 引擎 Shark 负责解析 XPDL。实验过程中由 workflow 相关数据决定实验的执行方向, 并直接决定了下一个活动的执行内容, 而 workflow 引擎发挥调度、协调的作用。

### 4.3 实验流程的动态组装方法

在上述实验过程中, 实验数据和实验流程的控制过程均与流程定义文件 XPDL 紧密相关。它是实现运行时实验流程动态组装的关键。由于实验资源是构件化的, 流程的执行过

程被转化为实验的建模过程, 一次实验的每一步操作被映射为一个活动(Activity), 待交互的数据建模为 workflow 相关数据(Workflow Relevant Data)。借助图形化 workflow 建模工具 JaWE, 可以很方便地完成实验流程动态组装过程以操作系统进程调度实验为例, 其实验过程建模如图 5 所示。



图 5 虚拟实验过程的可视化建模

一个由 JaWE 定义的 XPDL 片段示例如下:

```

<DataFields>
<DataField Id="ScheduleName" isArray="FALSE" Name="选择进程调度算法类型">
<DataType><BasicType Type="STRING"/></DataType><InitialValue>FIFO</InitialValue>
</DataField>.....
</DataFields>.....
<ExtendedAttribute Name="VariableToProcess_UPDATE" Value="arriveTimeA"/>
<ExtendedAttribute Name="VariableToProcess_UPDATE" Value="arriveTimeB"/>
<ExtendedAttribute Name="VariableToProcess_UPDATE" Value="serverTimeA"/>
<ExtendedAttribute Name="VariableToProcess_UPDATE" Value="serverTimeB"/>
<ExtendedAttribute Name="VariableToProcess_UPDATE" Value="timeSlice"/>
.....
<Transitions><Transition From="OS_Schedule_Wor1_Act1" Id="OS_Schedule_Wor1_Tra2"
To="OS_Schedule_Wor1_Act2"><Condition Type="CONDITION">ScheduleName="FIFO"
</Condition></Transition>
<Transition From="OS_Schedule_Wor1_Act1" Id="OS_Schedule_Wor1_Tra13" To="OS_Schedule_Wor1_Act3">.....
</Transition>

```

XPDL 文件将定义虚拟实验相关的页面构件、功能构件初始 URL, 以及实验输入参数。在 workflow 引擎 Shark 的驱动下, 通过相关操作调用即可完成数据和控制的流转。

### 5 应用实例

以文献[2]中构件组装方法为理论指导, 我们开发了轻量级 J2EE 框架下 workflow 驱动的动态虚拟实验室的系统原型。教师登录后的系统界面如图 6 所示。



图 6 教师登录后的系统界面

教师可以在线管理页面构件、功能构件, 并调用 Applet 封装后的 JaWE 进行基于 Internet 的实验流程建模, 创建或修改后 XPDL 将实时上载到服务器上, 并直接装载到 workflow 引擎实例库中。学生登录后的系统界面如图 7 所示。通过选择不同的课程, 学生可在不同的实验项目进行虚拟实验操作, 图 7 显示了操作系统课程中“进程调度”仿真实验的实验界面。实验过程完全由教师定义的 XPDL 文件所决定, 对于学生而言, 他无需关心给出当前仿真实验结果的特定功能构件, 这样, 教师可以在运行时方便地对实验项目进行重构。



图 7 学生登录后的系统界面

## 6 结束语

本文基于轻量级 J2EE 框架开发动态虚拟实验室系统的方法中, 虚拟实验资源以服务构件形式封装, 实验流程通过图形化建模工具 JaWE 被描述为构件的动态组装过程, 由工作流引擎 Shark 驱动, 实现了数据流和控制流的自动导航, 在轻量级 J2EE 框架(SSH)的支撑下, 可以大大降低虚拟实验室系统中各模块的耦合度, 提高系统的运行时性能, 实现系统的可控制性、可扩展性和可复用性, 从而支持运行时快速构建或重构实验项目。应用实例说明了本文方法的有效性。

### 参考文献

- 1 Grimaldi D, Rapuano S. Hardware and software to design virtual laboratory for education in instrumentation and measurement. Measurement, 2009, 42:485 - 493.
- 2 马华,陈振.基于构件组装的远程实验教学系统研究及应用.计算机系统应用, 2009,18(11):130 - 134.
- 3 马华,张红宇.一个支持服务协作的工作流管理系统.计算机系统应用, 2007,16(6):5 - 8.
- 4 戚琦,廖建新,王纯,武家春.基于敏捷方法的轻量级 J2EE 架构的应用.计算机系统应用, 2007,16(2):53 - 56.
- 5 Enhydra. Open Source Java XPD L Workflow.[2009-12-03]. <http://www.enhydra.org/workflow/shark/index.html>.
- 6 Enhydra. Open Source Java XPD L editor. [2009-12-03]. <http://www.enhydra.org/workflow/jawe/index.html>
- 7 WfMC. Workflow Process Definition Interface--XML Process Definition Language Version1.0. The Workflow Management Coalition Specification, October 25, 2002.