简化Java EE应用程序客户机开发的演示文稿框架

**摘要**：本文的主要目的是报告软件框架的构建，该软件框架极大地简化了具有瘦客户端的JavaEE应用程序的表示层的开发。JavaEE应用程序中表示层的设计原则之一是将业务层流控制和来自表示层的调用分开。为此，JavaEE应用程序通常使用Mult-2架构，其中包括著名的模型-视图-控制器（MVC）设计模式。MVC模式的优点是可以清楚地分离关注点，从而使应用程序更加灵活，最终更易于管理和更新。然而，MVC模式的使用承担了许多重复的任务，所有受支持的应用程序都必须执行这些任务，这使得开发工作既单调又复杂。为了克服这些缺陷和局限性，在本文中，我们提出了一个Web框架，用于开发JavaEE应用程序，简化了许多开发方面和任务，从而使Web应用程序更加灵活、可重用和可维护。这一贡献的出发点是调查和分析最常见的现有Java EE框架，在功能性、适用性和它们之间的优点比较方面具有相似的目的。这项研究使我们深入了解JavaEE应用程序的表示层，并为我们的框架收集适当的需求，重点是易于使用。然后用Java EE技术设计并实现该框架，并最终通过自组织测试方法进行评估。

# 1 引论

Java企业版（JavaEnterprise Enterprise，JavaEE）是开发企业应用程序的事实上的Java计算平台[1 ]。该平台为开发和运行企业软件（包括网络和Web服务）以及其他大型、多层、可扩展、可靠和安全的网络应用程序提供了API和运行时环境。JavaEE目前处于版本7，软件主要是在Java编程中开发的，它扩展了Java平台、标准版（JavaSE），提供了对象关系映射、分布式和多层体系结构以及Web服务的API。该平台包含了一个主要基于运行在应用服务器上的模块化组件的设计。

另一方面，软件框架和组件通常是为构建复杂的软件系统而开发的，在满足功能性和非功能性需求时减轻了开发人员的辛勤工作，并且在生产力、质量和成本方面越来越认识到其战略重要性。为此，许多Web框架已经出现，用Java EE来支持复杂企业软件的开发。

然而，虽然JavaEE框架要求简化Web应用程序的开发，但许多应用程序必须采取尖锐的学习曲线来实现相反的要求。事实上，开发人员抱怨的主要问题是未能满足“易于使用”的需求，增加了某些重复和乏味的任务。为了克服这些限制，我们提出了一个JavaEE Web框架，它极大地利用了设计模式，为开发人员提供了许多优点：（i）简化和规范输入参数的验证；（ii）将演示文稿从分离的组件中的业务层解耦；（iii）集中控制WOR；KFLOW管理；（iv）高级软件重用；（v）简化许多重复和乏味的开发任务。最终，生成的Web应用程序更加灵活、可重用和易于维护。

为此，在本文中，我们首先从第二节开始，对现有的JavaEE表示框架以及我们的框架中使用的核心JavaEE设计模式进行了广泛的调查。框架和设计模式的分析结果都设定了开发框架的要求，这是从第三节生命周期的所有阶段报告的。在第四节中，框架是通过开发用于评估目的的NA-ve应用程序进行测试的。最后，第五部分总结了论文的主要思想和评价结果，并概述了今后的工作。

# 2 背景

在本节中，我们将回顾支持JavaEE的主要软件框架和设计模式，我们将在后面的章节中介绍我们的框架。

## A Struts 2

Apache Struts是支持Web应用程序开发的JavaEE框架，使其实现更简单更简单。为此，通过自动化某些任务，该框架减轻了应用程序领域中常见的和繁琐的开发工作。Struts 2还通过在域工作流中提供架构解决方案，使JavaEE应用更加健壮和灵活。

在JavaEE三层体系结构（即，表示、逻辑和域层）中，Struts 2通过实现模型-视图-控制器（MVC - II型）设计模式来支持表示层，该框架可以以不同的方式设置，为视图提供特定的组件，并允许与OT的平滑集成。她的流行框架，如Hibernate和Spring[4]支持模型组件。

动作是MVC中Struts2框架的核心。每个URL都映射到一个特定的操作，该操作提供服务于用户请求所必需的处理逻辑。在将数据从用户请求传输到视图（JSP或任何类型的结果）以及确定哪个结果应将视图呈现为对请求的响应时，操作也扮演着重要的角色。

最后，Struts2包含一个Web标记库，以便更容易地呈现动态数据。标签与评估和国际特性交互，以验证输入和区域设置输出。该库还可以与Java服务器页面（JSP）、FreeMarker、O速度（5）一起使用。最后，外部标记库也可以在框架中使用，例如Java服务器页面标准标签库（JSTL），并支持使用Java Server FACTS（JSF）组件。

综上所述，Struts2主要通过减少繁琐的配置任务，使开发人员更容易开发Web应用程序。为此，框架提供了一组智能的默认值，并通过遵循某些约定使用注释来自动配置。

Struts2与其他表示框架（参见下一小节）相比的优点是：

简单的架构

简单易学，学习曲线短

完成数据转换和验证

最简洁、最成熟、功能最丰富

最常用的框架

Struts 2的缺点是：

不良事件管理

更低的用户界面组件重用级别

## B Spring MVC 3

Spring是一个简化Java企业应用程序开发的框架。Spring使用简单的JavaBeans，其结果类似于以前版本的EJB3的复杂企业JavaBeans（EJB），EJB3需要使用复杂的方法，如EJBActudiAd（）、EJBSufVATE（）等。在春季，通常任何组件是一个简单的Java类（普通的Java对象或POJO）。

为了降低复杂性，Spring提出了四种策略：

用简单的Java类或POJOS开发光。

通过注入依赖关系和接口方向实现弱耦合。

按方面和公共约定的声明性编程。

通过使用模板和方面（如数据库连接管理）减少重复代码。

与其他表示框架相比，Spring的优势在于：

通过自动连接和检测减少重要的配置工作。

通过注释使用简单POJO

使用面向方面编程（AOP）来管理事务、安全性等。

MVC Spring3包括与Web服务REST的兼容性。

此外，MVC框架Spring还包括消息传递、测试实用程序等。

与其他框架直接集成

Spring的主要缺点是学习曲线长，这主要是因为它提供了许多功能。

## C Java Server Faces 2

Java Server FACTS（JSF）是Java社区过程（JCP）在2001发布的Java规范请求（JSR-127）规范。在2006中，JSF 1.2被纳入JavaEE 5作为JSR 252。在这个版本中，使用了JSP，但是由于不同的生命周期，JSF出现了集成问题，因此Facelets被作为JSP的替代品引入，具有更多的功能，例如对XHTML页面的支持。最后，JSF 2被开发为JSR 314成为JavaEE 6中用于Web开发的首选选项。JSF 2受到许多开源Web框架的启发，因此具有许多特性。

JSF的目的是用Java Swing、AWT、SWT和相关API来开发类似于桌面应用程序的Web应用程序。通过这种方式，JSF通过提供一个Web框架来简化Web应用程序的构造，该框架管理用户执行的Web操作，并将其转换为发送到服务器以更新Web页面的事件。通过这种方式，JSF开发了Web应用程序，其中客户机窗口是HTML页面，而不是JFrame或类似的页面。

JSF应用程序是通过faces servlet管理HTTP请求并生成HTML代码的标准Web应用程序。JSF体系结构允许使用不同的Web语言，可以在不同的设备中查看页面，也可以使用组件、事件和侦听器创建页面，与Swing工作类似。例如，JSF有一组UI工具，例如按钮、复选框、文本字段等，允许轻松集成第三方组件。

与其他表示框架相比，JSF 2的优势在于：

作为Java社区发布并包含在Java EE中的正式规范（当前JSF 2是Java EE 6的一部分），所有JavaEE服务器都包括JSF。

从开放源码框架中合并许多特性。

使用简单的pojos，如“托管bean”，只需满足bean要求，还可以使用注释。

广泛的API和标签集，为相关客户创建HTML表单，类似于通过JAVA Swing开发桌面应用程序。

使用不同于HTTP的通信协议生成不同HTML格式的图形。

在同一组件中管理复杂的GUI，并允许使用来自第三方的比其他框架更多的组件库。

通过简单的标签使用jquery和ajax。Ajax支持不那么复杂。

更好地支持和集成到开发工具中，如NetBeans和Eclipse。

JSF 2的主要缺点是：

学习曲线比其他框架（如struts）慢。

缺乏对客户机层的验证（即Struts通过使用javascripts获得支持）。

# 3 Java EE 演示框架

在本节中，我们将介绍我们的JavaEE演示框架。首先，框架的要求是从上一节介绍的相关工作中获得的。然后给出了框架的设计，最后给出了实现方案。框架评估将在下一节中提供。

## A 需求

在前一节中，我们已经介绍了三种用于表示层的最常用的Java EE框架。根据分析的优点和缺点，我们可以确定以下任何表示框架都应该具备的理想特性列表，这些特性构成了我们框架的需求（参见第二节的回顾）：

为了分离视图和模型，应该考虑MVC2模式，其中servlet充当客户机请求的控制器。此控制器将根据输入和输出参数以及处理用户请求的结果集中选择下一个视图的逻辑。为此，还需要使用应用程序控制器模式。

自动数据传输和将客户请求转换为操作

自动数据验证

声明性工作流，以便在给定用户请求的情况下，命令和视图的序列独立于应用程序代码声明。

需要标记库来避免硬编码JSP

支持国际化，使应用程序以用户首选语言显示，而不重新编码应用程序

易于扩展，以便添加处理用户请求的新功能/命令，并将适当的结果作为对请求的响应。

综上所述，上述需求决定了我们的框架将遵循Struts2面向操作的框架的大部分特性。接下来，框架设计将根据设计模式做出某些决策，以满足这些需求，并克服所提到的Struts的一些缺点。

## B 框架设计

我们的框架的设计将广泛使用JavaEE呈现设计模式。MVC 2将与一个名为frontController（从httpservlet继承）的前端控制器一起使用。在此上下文中，将使用应用程序控制器模式从控制器中提取应用程序逻辑。此外，Context对象模式用于用名为RequestContext的类封装来自请求的数据，以便FrontController可以从其已知的请求数据创建RequestContext对象，并将其委托给ApplicationController。最后，将使用服务到工人模式，以更好地设计操作，并改进控制集中和用户请求处理（有关设计模式的审查，请参见第二节）。

应用程序的表示层将由操作和表单组成。动作和表单的自动调用将通过API反射实现，从而满足我们框架的这一要求。表单属性的调用和验证将由基于过滤器的类实现，这些类与表单协作，但没有执行前和执行后的行为。每个操作都将实现命令模式，该模式在我们的框架设计中被广泛使用（例如，它被应用于过滤器和结果中）。通过命令，我们可以向内容先验未知的对象请求操作。

最后，我们的框架结构将充分利用Java架构的XML绑定[10 ]。这样，客户机将通过XML文件定义其表示层，在XML文件中定义表单、操作、结果和过滤器。此文件将根据XML方案进行验证。此外，结果是一个具有属性的元素，该属性允许通过XML文件定义导航路径（即正常路径、错误路径或临时路径）。通过这种方式，我们可以将导航参数化为框架的目标之一。

综上所述，表示层将根据形式和动作来定义，因此在配置中会有形式和动作，每个都由一个类实现。此外，每个操作都将定义结果。

## C 框架实现

框架的封装结构如图1所示。

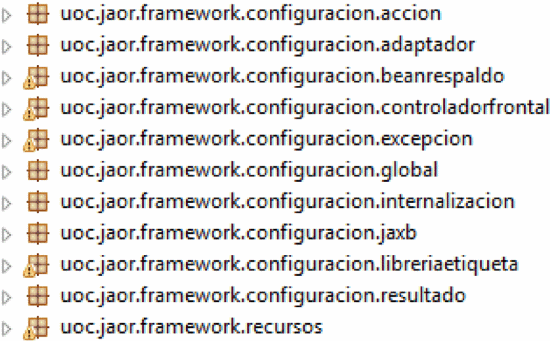


图1 包结构

下面将介绍框架中最具代表性的包（参见图1）：

Accion包括接口操作、实现上下文对象模式的RequestContext类和实现相应设计模式的ApplicactionController类。此实现设置了使用框架的操作的要求，以及当框架的前控制器委托执行时如何处理这些操作。

BeanRespaldo除其他元素外，还包括处理表单的类以及表单上的所有表单和筛选器必须遵守的接口。

ControlAdorFrontal包括实现前控制器模式的servlet。

global包括工厂类，负责通过jaxb生成客户机XML中定义的对象来定义客户机的表示层。

resultado包括那些使用servlets实现结果的类。定义了一个特定的接口，使用此框架的所有结果都必须符合。

最后，用于框架开发的库是GlassFish 3.1.2和JAXB。Eclipse Java EE IDE用于Web开发（靛蓝版），带有GLASISFISH插件。

# 4 框架评估

本节介绍了一个名为pruebaderbyuoc的简单测试应用程序（见图2），用于评估我们的演示框架，该框架的开发已在上一节中报告。测试在服务器glassfish[3]中执行。

测试的目的是使用我们的框架为书店开发一个Web应用程序。应用程序的要求包括认证（见图3和4）、作者列表（见图5）、特定作者的搜索书籍（见图6）、按姓名搜索作者和添加新书。

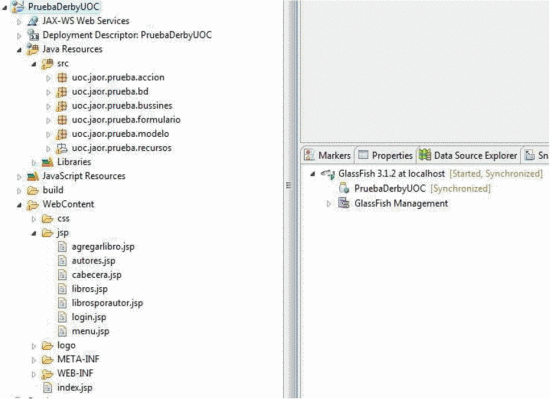


图2 测试包结构

测试应用程序的主要包如图2所示：

Accion使用实现Accion接口的基本AccionBase类和从基类继承的其余类定义应用程序的操作。

Business主要实现Facade模式，以拥有对该包表示的业务层的唯一访问点。

Formulario定义应用程序的所有表单。表单的基类是实现Formulario接口的Formulariobase。所有表单都继承自基类，例如登录时使用的FormularioEntrada（参见图3），作者搜索书籍时使用的FormularioBrosdeautor（参见图6）。

Modelo表示集成层，其中定义了用户、作者和书籍的主要实体。



图3

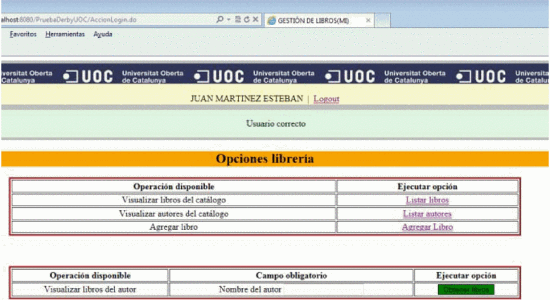


图4

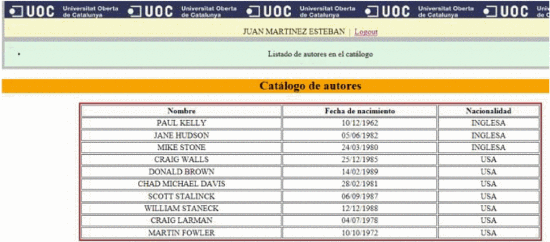


图5

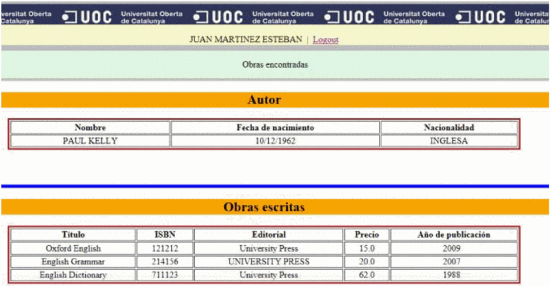


图6

# 5 结论以及未来的工作

在本文中，我们提出了一个软件框架，以开发支持表示层的Java EE应用程序。首先，对现有的JavaEE表示框架进行了广泛的调查，并在我们的框架中使用了核心的JavaEE设计模式。框架和设计模式的分析结果都为开发我们的框架设定了需求，这是从生命周期的所有阶段报告的。最后，通过开发一个用于评估目的的Na\_ve应用程序来测试框架。

从评估中，我们得出结论，我们的框架简化和更容易发展的测试应用程序遵循Struts 2的行动，这提供了一个清晰的模型和实现许多Java EE设计模式，例如MVC-2和应用控制器。然而，我们的框架通过实现与Struts模式一起工作的某些设计模式来获得更好的性能。例如，服务到工作者和ContextObject模式有助于提供一个集中的控制点，允许通过结果进行参数化导航，属性的调用和验证变得更加容易。此外，我们的框架还支持过滤器来拦截输入请求和输出响应，允许以灵活的方式进行前处理和后处理。最后，框架提供了国际化和支持注释的标记库。

我们计划进一步评估我们的框架，利用它在各个领域开发完整的Web应用程序，以便在开发新的框架迭代之前获得详尽的反馈。我们还研究了在框架中实现更多的JavaEE模式，以使开发工作更容易，并且具有更多的可重用和可维护的应用。