厦门理工学院
XIAMEN UNIVERSITY OF TECHNOLOGY

毕业设计（论文）外文文献翻译

院　　系：计算机与信息工程学院

年级专业：

姓　　名：

学　　号：

附　　件：Spring Web Mvc Framework for Rapid Open Source J2EE Application Development

指导老师评语：

指导教师签名：

年　　月　　日

运用 spring mvc 框架进行快速的开源 J2EE 应用程序开发：案例研究

摘要—当今，web 应用程序的开发竞争非常激烈，时代要求开发出的应用程序非常准确、经济和高效。人们致力于开发出能提高生产效率和降低复杂性的应用程序，转变程序员开发 java2 平台企业版的 web 应用程序的方法已经成为一个基本的运动。本文中我们讨论的重点是怎样开发出 j2ee 兼容的软件而无需使用企业 java bean（EJB）。最好的选择之一就是运用 spring 框架， spring 框架提供了许多服务，但是相比于 EJB，它的侵入性大大降低了。这种转变背后的驱动力是 web 应用程序开发和实施领域提高生产率和降低复杂性的需要。本文中，我们将简要介绍 spring 的基本体系结构并且给出一个运用了 spring mvc 框架的案例研究实例。

关键词：mvc，spring，xml

Ⅰ、简介

现今，网络问题是非常复杂的。由于公司和组织的需求都在不断的增加，应用程序开发的复杂性和系统性能是需要解决的主要问题。不同类型的通讯设备的复杂性在不断增加，而业务要求应用程序使用网络和许多通讯设备，并且互联网上数据负载不断的增加，这些迫使我们不得不考虑起应用程序的体系架构问题。现在，让我们讨论在保持应用程序模型视图结构不变的情况下，spring web mvc 快速应用程序开发框架是如何快速工作的。

Spring 框架具有丰富的功能集，我们将简要讨论这些功能。

1.控制反转：控制反转，即 IOC。它是有线服务或者将组件添加到应用程序所使用的技术之一。IOC 是“一种软件设计模式和相关的编程技术集”，与传统的交互模式相比，运用 IOC 后，系统的控制流是反向的。在 IOC 容器内不是应用程序调用框架而是框架调用应用程序指定的组件。IOC 可以被解释为“把运行时所需要的资源或者依赖注入到相关的资源中去”，这也被称为依赖注入。org.springframework.beans.factory.BeanFactory 是 spring IOC 容器的实际表现，它是负责控制和管理 bean 的。BeanFactory 接口是 spring IOC 容器的主要接口。Bean 就是由 spring IOC 容器实例化和管理的对象。这些 bean 和他们之间的依赖关系反应在容器所使用的配置元数据中。

2.构造函数依赖注入：我们可以使用 java 类的构造函数来加载 bean 的值。首先

定义一个只有单一构造函数的类，然后使用 details.xml 文件提供构造函数所需要的值，最后用一个实现了 beanfactory 接口方法的类来加载 xml 文件。这是使用 xml 文件把值加载到 java 文件的构造函数中，这种方法适用于向构造函数传递值。

3.Setter 依赖注入：给每一个 bean 定义其 get 和 set 方法。我们可以利用 set 方法在设定 bean 中的值。Set 方法会覆盖掉从 bean 中加载的值。

4.接口：我们可以在 spring 中定义接口类。为了实现这一点，必须为 java 程序导入接口，然后我们可以利用接口中定义的方法来使用 spring 和 xml 文件。

5.继承：一个 java 类可以获得另一个 java 类的属性，就像只有一个 java 程序一样。继承有三种子类型：1)抽象型：spring 中定义成抽象类的 bean 只能被继承。2)父子型：定义的的继承层次结构想父子关系一样。3)父- 子 –孙子型：这种类型的继承关系可以定义 3 个或者更多的类层次结构。

6.自动装配：自动装配用于将 xml 文件中属性的键和值映射到 java 文件中去。有 4 种装配的类型：byName（通过名字）、byType（通过类型）、constructor（通过构造函数）、autodetect（自动检测）。如果没有定义装配的类型，那么默认是以通过名字的方式来装配的。

7.Bean 的作用域：spring 中定义的所有 bean 有四种类型的作用域，即 session，request，singleton，global-session。这些是用来控制 bean 的访问范围的。

8.引用 bean：在 xml 文件中的一个 bean 可以从其他的 bean 分配值。这尝尝用于从一个 bean 中读取值然后再分配给另一个 bean。

II、spring 的主要组件

Spring 框架依然遵循 mvc 的思想原则。它是为桌面应用和基于互联网的应用而设计的。Spring 由三个相互协作的核心组件组成。1、控制器：处理业务逻辑中的跳转逻辑和同服务层的交互。2、模型：控制器和视图之间的桥梁，包含着控制器给予视图所需要的数据。3、视图：呈现请求后的响应，从模型中提取数据。Spring 的核心组件如下所示：

1.DispatcherServlet：它是 spring 的前端控制器的实现。Web.xml 接收请求并且把它转移到 DispatcherServlet，DispatcherServlet 是与请求进行交互的第一个控制器，它也被称为 Servlet 的执行。它控制着应用程序的完整流和跳转流。

2.控制器：这个是使用者为了处理请求而创建的组件，它封装了跳转逻辑。控制器将服务委托给服务对象。

3.视图：视图是负责渲染呈现输出的。在输出的结果集、显示设备和通讯设备的基础上，不同类型的输出要选择不同的视图。

4.ModelAndView：ModelAndView 是 spring 框架的核心部分。它实现了应用程序的业务逻辑，由控制器创建。它使视图和请求联系起来并且存储了业务逻辑和模型数据。当一个控制器调用它的时候，它才会执行。在执行时，它将返回视图的数据和名字。

5.ViewResolver：输出是如何显示的取决于从 ModelAndView 中接收的结果。ViewResolver 是将逻辑视图名映射到实际视图的实现。这部分将确定和实现输出的是什么媒体以及如何去显示它。

6.HandlerMapping：它是 DispatcherServlet 将传入的请求映射到单个控制器时所使用的战略性接口。它识别请求并且调用相应的处理程序来提供服务，处理程序会调用控制器。

Ⅲ、spring 的体系结构

Spring 框架为开发 web 应用程序提供了功能全面的 mvc 模型。Spring 具有可插拔的 mvc 架构，他可以配置多种视图技术，例如：jsp、velocity、tiles、iText 等等。Spring mvc 分离了控制器，模型对象，分派器和处理对象的角色。对象和控制器的清楚分离，使他们更容易进行定制。图 01 显示了执行流程。

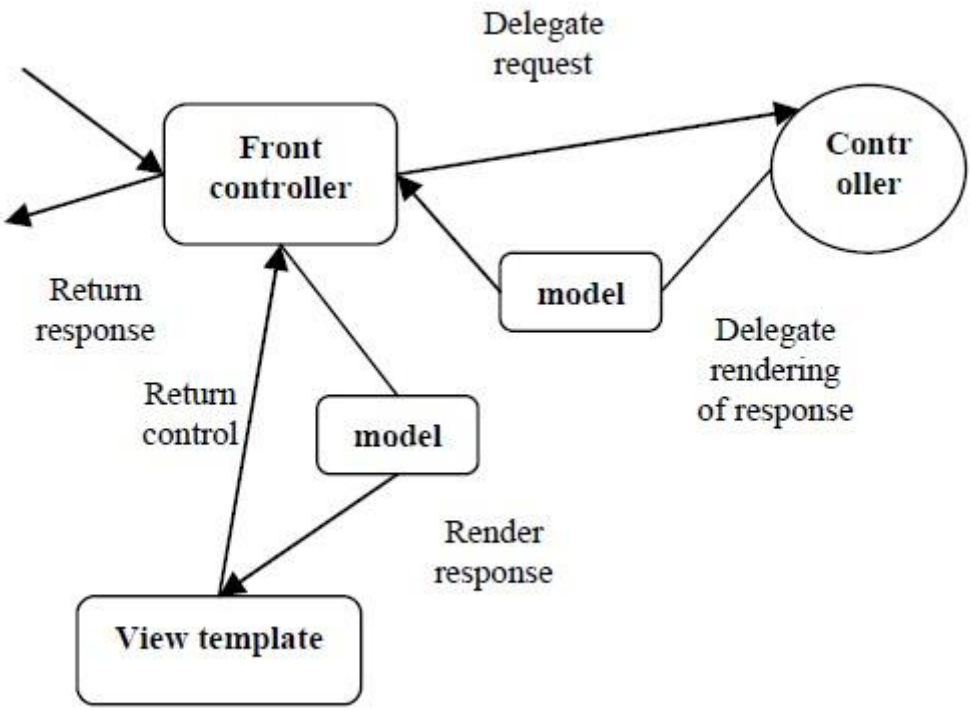

图 01

下面的图 02 显示了 spring 模型的序列图。图中 dispatcher Servlet 是应用程序的

入口点，一旦 dispatcher Servlet 获取请求服务，它就会决定处理程序。所有的处理程序和 servlet 都是相互映射的。处理程序开始运行和调用相应的控制器，并且将请求的参数传递给控制器，这个时候控制器开始工作，它包含了业务逻辑，同时 ModelAndView 与控制器关联起来，在执行时，控制器会把 ModelAndView 返回给 dispatcher Servlet，这个时候的 ModelAndView 包含了数据 和视图名。Dispatcher Servlet 从控制器从获得 ModelAndView，然后 Servlet 会调用相应的视图解析器。视图解析器会识别出视图的名称并且通过视图名来提取相应的数据，最后，它会以适当的格式向用户呈现各自的数据。

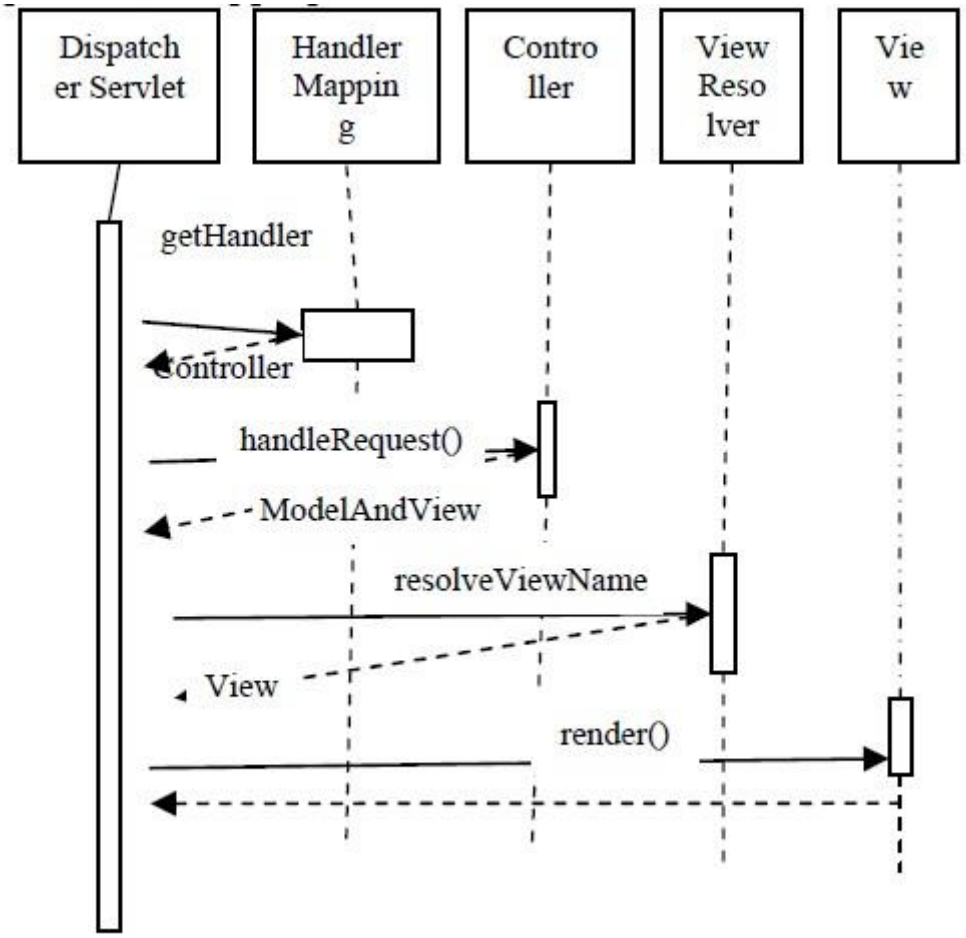

图 02spring 框架中应用程序的流程

Ⅳ、spring 和 xml

Xml 广泛使用在 spring 框架中，它简化了开发流程，节约了时间。Xml 用于存储在应用程序执行时所需要的一些数据。

Web.xml 文件是应用程序的入口点，它会告诉你进一步的跳转路径。它负责加载应用程序上下文并且指定了负责调度 servlet 的 xml 文件名。

web.xml:-

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="1.0" xmlns="http//java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemalLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app-2_5.xsd">
<listener><listener-Class>
org.springframework.web.context.ContextLoaderListener
</listener-Class></listener>
<servlet>
<servlet-name>dispatcher</servlet-name>
<servletclass>
org.springframework.web.servlet.DispatcherServlet
</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/send/*</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

以上就是 xml 文件，xml 将与传入到服务器的请求经行交互。xml 文件中明确了要调度的 servlet 的名称，应用程序上下文和 index.jsp 作为欢迎页面。url 的类型定义为*.*，这意味着它将会接受任意类型的传入请求。

ApplicationContext.xml – ApplicationContext 建立在 BeanFactory 之上，它使 AOP 特征、消息资源处理和事件传播更容易集成在一起。 BeanFactory 提供了框架的配置和基本的功能，ApplicationContext 则大大提高了应用的性能。当在 j2EE 的环境下开发应用程序，ApplicationContext 是必不可少的。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
```

```
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-beans-2.5.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">
<bean id="superClass" class="packagename.SuperClass" />
<bean id ="subClass" class=" packagename.SubClass">
</bean>
<property name="superClass" ref="superClass"/>
</beans>
```

上述文件将加载 superclass.java（超类）的 bean 和 superclass.java 本身，同时它还定义了参考类。

Dispatcher-servlet.xml- spring 的 web mvc 框架是一个以请求为驱动的 web mvc 框架，它是围绕一个 servlet 设计的，该 servlet 将请求转发给控制器并且为处理应用程序提供了大量的功能。DispatcherServlet 与 spring 的应用程序上下文完全集成在一起并且允许你使用 spring 的功能，它是应用程序运作的中央控制单元。DispatcherServlet 定义了视图解析器、bean 类以及他们的在应用程序中的映射。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns=http://www.springframework.org/schema/beans
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-beans-2.5.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">
<beanid="viewResolver"  class="org.springframework.web.servlet.view.Internal-
ResourceViewResolver">
<property name="prefix">
```

```
<value>/WEB-INF/views/</value></property>
<property name="suffix"><value>.jsp</value></property>
</bean>
<bean id="urlMapping" class="org.springframework.web.servlet.handler.
SimpleUrlHandlerMapping">
<property name="mappings">
<props><prop key ="/*">dispatchController</prop>
</props>
</property>
</bean>
<bean id="dispatchController" class=" packagename.DispatchController"></bean>
</beans>
```

Ⅴ、spring 是如何快速运作的

快速应用程序开发一直以来是行业的要求。目前有很多开发工具来帮助快速应用程序开发，但是随着技术和需求的持续增加，能够解决开发规模不断扩大的应用程序的工具和架构是急需的。工具不同于架构，spring 是 mvc 架构中的一种架构，它可以支持规模较大的应用程序。这种技术和架构一旦实现，它很容继承其它的应用程序而不触及现有的代码。Spring 使用 xml 文件来帮助我们添加新的映射、请求、java bean 等等到应用程序中去。

Ⅵ、架构的优点

让我们看看 spring mvc 架构可以为一个项目带来哪些架构上的好处。

1.spring 可以有效的组织程序的中间层对象，而 EJB 则不会影响它。Spring 的配置管理服务可以运用在任何架构层和任何运行时环境中。

2.spring web mvc 框架是一个强大的，灵活的，精心设计的应用程序框架，用于使用 mvc 设计模式快速的开发 web 应用程序。

3.spring 消除了单例模式的扩散，这是一个主要的问题，它减少了程序的可测试性和面向对象性。

4.清晰的角色划分：spring mvc 很好的将组成 web 框架的各个组件所扮演的角色划分开。所有的组件，例如控制器、命令对象、设置器等，每个组件都扮演着独特的角色。

5.适配控制器：如果你的应用程序不需要一个 html 表单，你可以写一个简单的版本的 spring 控制器，该控制器不需要表单控制所需要的的所有额外的组件。Spring 提供了几种类型的控制器，每一种控制器用于不同的用途。

6.Spring 不再需要使用各种自定义属性文件格式，在整个应用程序和项目中配置操作始终是一致的。

7.Spring 提供了良好的编程习惯来降低编程接口的成本，而不是类的成本。

8.用 spring 构建的应用程序尽可能少的依赖于它的 API，在 spring 的应用程序中大多数的业务对象都没有依赖于 spring。

9.使用了 spring 构建的应用程序是非常容易进行单元测试的。

10.Spring 可以使用 EJB 来实现选择，而不是决定应用程序的架构。

11.你可以选择 POJO 或者本地 EJB 实现业务接口，而不影响代码的调用。

12.Spring 提供了 EJB 的一个替代品，这种适合很多应用程序。它可以在不使用 EJB 容器的情况下使用 AOP 提供声明式事务管理。

13.无论是使用 JDBC 或者 hibernate 的 O/R 映射时，spring 提供了一致的数据访问框架。同时，它也提供了一致的和简单的编程模型，例如 JDBC,JMS,JavaMail,JNDI 和许多的 api,这些使得 spring 是一个理想的架构。

14.它是使用 POJO 创建应用程序的框架，同时它也掩盖了开发程序的复杂性。

15.当在使用 JDBC 时，spring 解决了连接泄露的问题，我们只需要编写必要的 SQL，它也解决了数据库返回错误的问题。

Ⅶ、总结

Spring web mvc 框架是一个在快速应用程序开发的环境下为应用程序提供环境的框架。在此框架下，我们可以信赖应用程序的一致性，性能和稳定性。由于这是一个开放源码的环境，所以建议开发者继续使用这种技术用于大尺寸的 web 应用程序环境。

Spring Web MVC Framework for rapid open source J2EE application development: a case study

Praveen Gupta
Research Scholar, Singhania University
Pacheri Bari, Rajasthan, India
Prof. M.C. Govil
Govt. Mahila Engineering College
Ajmer, Rajasthan, India

Abstract— Today it is the highly competitive for the development of Web application, it is the need of the time to develop the application accurately, economically, and efficiently. We are interested to increase productivity and decrease complexity. This has been an underlying theme in a movement to change the way programmers approach developing Java 2 Platform, Enterprise Edition (J2EE) Web applications. Our focus is how to create J2EE-compliant software without using Enterprise Java Beans (EJB). The one of the best alternative is the Spring framework, which provides less services but it is much less intrusive than EJB. The driving force behind this shift is the need for greater productivity and reduced complexity in the area of Web application software development and implementation. In this paper, we briefly describe spring underlying architecture and present a case study using Spring web MVC Framework.
Index Terma: MVC, Spring, XML

I. INTRODUCTION

Web is the very complex issues these days. Since the desire of the companies and organizations are increasing so the complexity and the performance of the web programming matters. Complexity with the different types of communication devices is increasing. The business is demanding applications using the web and many communication devices. So with the increase load of the data on the internet we have to take care of the architecture issue. Let us discuss how it works fast using spring web mvc framework the rapid application development while maintaining the Model View

Architecture of the application.

Spring frameworks comes with rich set of features, let us discuss these features in brief.

1. Inversion Of Control: Inversion of Control or IoC is one of the techniques used to wire services or components to an application program. The IoC is “A software design pattern and set of associated programming techniques in which the flow of control of a system is inverted in comparison to the traditional interaction mode.” In IoC instead of an application calling the framework, it is the framework that calls the components specified by the application. The IoC can be explained as "Injection of required resources or dependency at run-time into the dependent resource" which is also known as Dependency Injection. The org.springframework.beans.factory.BeanFactory is the actual representation of the Spring IoC container which is responsible for containing and managing the beans. The BeanFactory interface is the central IoC container interface in Spring. A bean is simply an object that is instantiated and managed by a Spring IoC container. These beans and the dependencies between them are reflected in the configuration metadata used by a container.

2. Constructor Dependency Injection: we can use the java class constructor to load the bean values. A java Class is defined with a constructor of single field. Details.xml file provides the value to be passes to the constructor. Now another java loads the xml file using the BeanFactory Method. This uses the xml file to load values in the constructor of the java file. This is used to pass values to the constructor.

3. Setter Dependency Injection: With every bean we defined the getters and setters. We can also use setters method to set the values in the beans. setters method overrides the values loaded from the beans.

4. Interface: we can define the interface class in spring. To implement this we will import interface to the java program. Now we can use methods defined in interface using spring and xml.

5. Inheritance: One java class can acquired the properties of another class just like a java program. There are three sub types of it. 1. Abstract: Beans declared abstract cannot be inherited in the springs. 2. Parent Child: we can define hierarchy like parent child. 3. Parent - Child - Sub Child Relationship: in this we can define hierarchy for 3 or more classes.

6. Autowiring: Autowiring is used to map the property name, values in xml file with java file. There are four types to integrate it. byName, byType, constructor, autodetect. If nothing is defined about it then byName is the default.

7. Scope of Beans: All beans defined in spring are having scope of four values prototype, session, request, singleton,global-session. This is used to control the access of the beans.
8. Reference Beans: One bean in the xml file can be assigned values from the other bean. This is used to read values from one bean and assign to another bean.

II. MAJOR SPRING COMPONENTS

In the spring we also follow the principals of the MVC. It has been designed more for the desktop and internet based applications. Spring consist of three core collaborating components. 1. Controller: Handles navigation logic and interacts with the Service tier for business logic2. Model: The contract between the Controller and the View Contains the data needed to render the View Populated by the Controller 3. View: Renders the response to the request Pulls data from the model. Core components in the spring MVC are as follows.

1. DispatcherServlet: this is the spring's front controller implementation. Web.xml receives the request and transfer it to the DispatchServlet. This is the first controller which interacts to the requests. It is also known as implementation of the Servlet. It controls the complete flow of the application and navigates the flow of application.
2. Controller: this is the user created component for handling requests. It encapsulates the navigation logic with it. Controller delegates the services for the service object.
3. View: view is responsible for rendering output. Different views can be selected for the different types of output bases on the results and the viewing device, communication devices.
4. ModelAndView: ModelAndView is the core part of the spring framework. It implements the business logic of the application. It is created by the controller. It associates the view to the request. It stores the business logic and Model data. A controller calls it and it will execute. On execution it will return the data and name of view.
5. ViewResolver: How the output is to be displayed depends on the result received from ModelAndView. It is used to map logical view names to actual view implementations. This part identifies and implement what is the output media and how to display it.
6. HandlerMapping: Strategy interface used by DispatcherServlet for mapping incoming requests to individual Controllers. It identifies the request and calls the respective handler to provide the services. Handler will call to controller.

III. SPRING ARCHITECTURE

The Spring framework provides a full-featured MVC module for building Web applications. with spring's pluggable MVC architecture. It is configurable with multiple view technologies Ex Java Server Pages, Velocity, Tiles, iText etc. Spring MVC separates the roles of the controller, model object, dispatcher Servlet and the handler object. Clear separation of objects and controllers makes them easier to customize. The figure 01 shows the view of the execution flow.

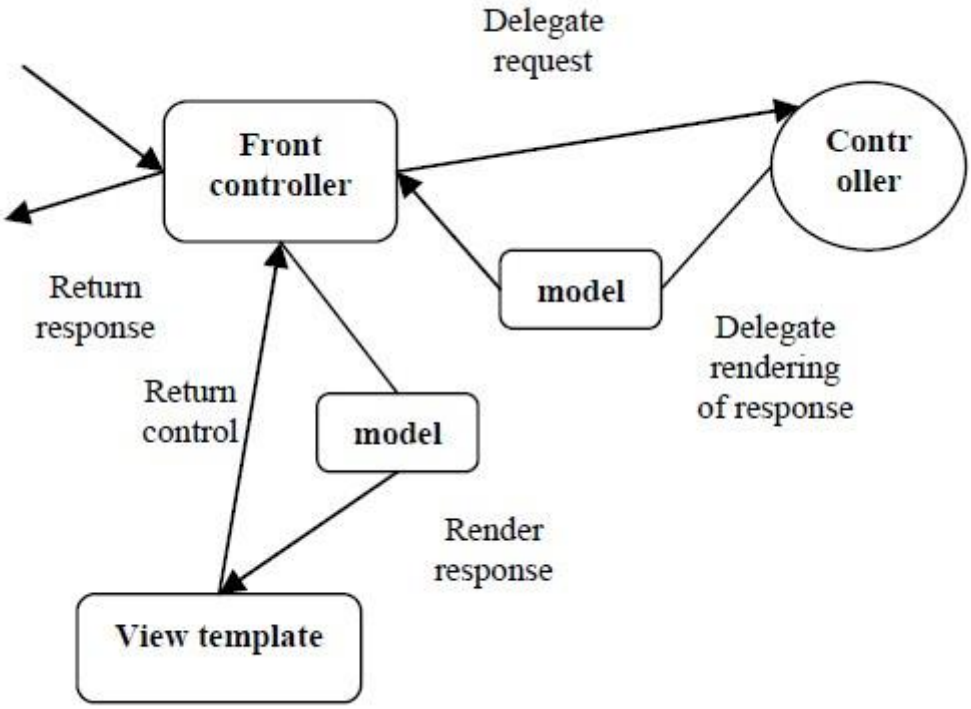

Figure 01

The following figure 02 shows the sequence diagram of the spring model. In this the dispatcher Servlet is the entry point for the application. As soon the Dispatch Servlet get the request for the services and it will decides the handler.All handlers are mapped with the Servlet. Handler will come in action and will call the respective controller and the pass the request parameters to it.Now controller comes in action, it contains business logic and a ModelAndView is associated with the controller. On execution it will return the ModelAndView to the Dispatch Servlet. This ModelAndView contains the data and view name.Dispatcher Servlet gets the ModelAndView from the controller. It contains the data and view name. Servlet will call the view resolver. View resolver will identify the name of the view through which data is to be presented. Finally it will present the

data to the respective and appropriate format to the user.

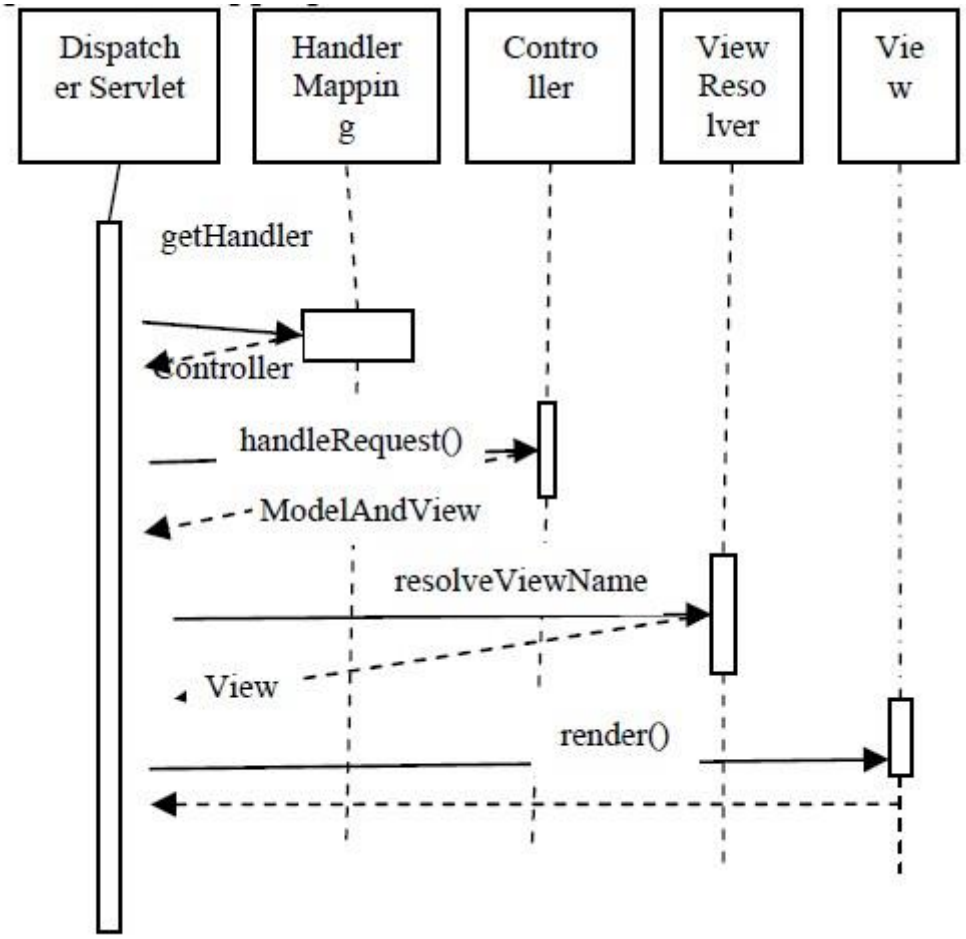

Figure 2：the sequence diagram of the spring model

IV. SPRING AND XML

Xml is widely used in the spring framework. It simplify the development process and saves time. xml is used to store the data, which is used during the execution of application.

web.xml is the entry point in the application. It will tell you the further path of navigation. It loads the application context class and the tells the name of the dispatcher Servlet xml file.

web.xml:- <?xml version="1.0" encoding="UTF-8"?>

<web-app version="1.0" xmlns="http//java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemalLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app-2_5.xsd">

```
<listener><listener-Class>org.springframework.web.context.ContextLoaderListener</listener-Class></listener>
<servlet>
<servlet-name>dispatcher</servlet-name><servletclass>
org.springframework.web.servlet.DispatcherServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/send/*</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

The above web.xml .xml will interact with the incoming request on the server. This file defines the name of the Servlet which is dispatcher, ApplicationContext and the index.jsp as welcome page. url pattern defined as *.* means it will all types of incoming request.

ApplicationContext.xml:- The ApplicationContext is build on top of the BeanFactory. It provides an easy integration with Springs AOP features, message resource handling, event propagation. The BeanFactory provides the configuration framework and basic functionality. ApplicationContext adds enhanced capabilities to application.

While building applications in a J2EE-environment ApplicationContext must be used.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-beans-2.5.xsd
http://www.springframework.org/schema/tx
```

```
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">
<bean id="superClass" class="packagename.SuperClass" />
<bean id ="subClass" class=" packagename.SubClass">
</bean>
<property name="superClass" ref="superClass"/>
</beans>
```

The above file loads the bean of the SuperClass.java and SubClass.java. It also defines the reference class.

Dispatcher-servlet.xml:-

Spring's web MVC framework is a request driven web MVC framework, it is designed around a servlet that dispatches requests to controllers and provides much functionality for handling the applications. DispatcherServlet is completely integrated with the Spring ApplicationContext and allows you to use feature of springs. Dispatcher Servlet is the central controlling unit for the working of the application. It is used to define the view resolver, beans, handlers and their mapping of the application.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns=http://www.springframework.org/schema/beans
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-beans-2.5.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">
<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<property name="prefix">
<value>/WEB-INF/views/</value></property>
<property name="suffix"><value>.jsp</value></property>
</bean>
<bean id="urlMapping"
class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
```

```
<property name="mappings">
<props><prop key ="/*">dispatchController</prop>
</props>
</property>
</bean>
<bean id="dispatchController" class=" packagename.DispatchController"></bean>
</beans>
```

V. HOW IT WORKS RAPID

Rapid Application Development is the requirement of the industry since a long time. There are many development tools which have helped it. But gradually technology and the requirements goes on increasing so e need the tools and the in fact architecture which can handle the growing size of the application. Tools are different from the architecture. Spring is the architecture in MVC which can support the large applications. In this technology and architecture once implemented it is easy to inheritance the application without touching the existing code. It's use of the xml files helps us to add the new mappings, requests, java beans etc to the application.

VI. ARCHITECTURAL BENEFIT

let's look at some of the Architectural benefits spring web MVC Framework can bring to a project.

1. Spring effectively organize your middle tier objects, EJB doesn't affect it. The configuration management services can be used in any architectural layer and in any runtime environment.
2. The Spring Web MVC Framework is a robust, flexible, and well-designed framework for rapidly developing web applications using the MVC design pattern.
3. Spring eliminate the proliferation of Singletons. This is a major problem, reducing testability and object orientation
4. Clear separation of roles: Spring MVC nicely separates the roles played by the various components that make up this web framework. All components like controllers, command objects, and valuators' each component plays a distinct role.

5. Adaptable controllers: If your application does not require an HTML form, you can write a simpler version of a Spring controller that does need all the extra components required for form controllers. Spring provides several types of controllers, each serving a different purpose.
6. Spring eliminate the need to use a variety of custom properties file formats, by handling configuration in a consistent way throughout applications and projects.
7. Spring provides good programming practice by reducing the cost of programming to interfaces, rather than classes.
8. Applications built with it depend on as few of its APIs. Most business objects in spring applications have no dependency on spring.
9. Applications built using spring are very easy to unit test.
10. Spring can make the use of EJB an implementation choice, rather than the determinant of application architecture.
11. You can choose to implement business interfaces as POJOs or local EJBs without affecting calling code.
12. Spring provide an alternative to EJB that's appropriate for many applications. It can use AOP to deliver declarative transaction management without using an EJB container.
13. Spring provides a consistent framework for data access, whether using JDBC or an O/R mapping, Hibernate. It provides a consistent and simple programming model in areas like JDBC, JMS, JavaMail, JNDI and many APIs which makes it an ideal architectural.
14. This is the framework which builds applications using POJOs. It also conceals complexity from the developer.
15. While using JDBC it solve the problem of connection leak, we need to write only necessary SQL, it also solves the problems of error returned from database.

VII. CONCLUSION

Spring WEB mvc framework is an framework which provides the environment for the application in the RAD environment. In this framework we can rely for the consistency, performance and reliability of the application. Since this is an open source environment so it’s recommended for the developers to go ahead with this technology for the large size of web application environment.

REFERENCES

[1] Shu-qiang Huang, Huan-ming Zhang, ” Research on Improved MVC Design Pattern Based on Struts and XSL” , in Information Science and Engineering ISISE 08 International Symposium on, 2008, vol. 1 PP. 451 – 455

[2] Juanjuan Yan; Bo Chen; Xiu-e Gao, "Le Wang; Research of Structure Integration Based on Struts and Hibernate" , in 2009 WRI World Congress on Computer Science and Information Engineering,2009, vol. 7,PP. 530-534

[3] Wojciechowski, J.; Sakowicz, B.; Dura, K.; Napieralski, A.,"MVC model, struts framework and file upload issues in web applications based on J2EE platform", in Proceedings of the International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science 2004, 2004, ,PP 342-345

[4] Erxiang Chen; Minghui Liu, "Research and Design on Library Management System Based on Struts and Hibernate Framework", in WASE International Conference on Information Engineering ICIE 09, 2009, Vol. 2,PP. 310-313

[5] Yonglei Tao; "Component- vs. application-level MVC architecture", in Frontiers in Education 2002 FIE 2002. 32nd Annual,2002, Vol 1,PP. T2G-7 - T2G-10

[6] Meiyu Fang, "Design and Implement of a Web Examination System Using Struts and EJB" ,Seventh International Conference on in Webbased Learning 2008, ,, 2008,pp. 25-28

[7] Wang Ning; Li Liming; Wang Yanzhang; Wang Yi-bing; Wang Jing,"Research on the Web Information System Development Platform Based on MVC Design Pattern",in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008 ,Vol 3,pp. 203-206

AUTHORS PROFILE

Praveen Gupta has received his Bachelor of Engineering in Electronics Engineering from Nagpur University,Nagpur, India, and Master in Technology in Information Technology from Punjabi University, India. He is working as Technical Leader in an software company based at Navi Mumbai, India.He is a member of various Technical Societies viz. Association of Computer Electronics and Electrical Engineers(ACEEE),He is a research scholar and pursuing his PhD at Singhania University , Pacheri Bari,

Rajasthan, India. His main research interests include: MVC, Java Design Patterns and Frameworks.

Prof. M.C. Govil has received Doctorate degree from IIT. He is working as professor and principal at govt women engineering college, Ajmer, India. He is having more than 20 years of experience in education and research. He is member of various Societies, he has published many papers. His research interest includes communication, web technologies, software engineering etc.