Java Struts Framework

Juan Fco. Rodríguez Hervella

Departamento de INGENIERIA TELEMATICA
www.it.uc3m.es

Indice

- Introducción
- Struts Framework
- Ejemplo
- Conclusiones.

Introducción

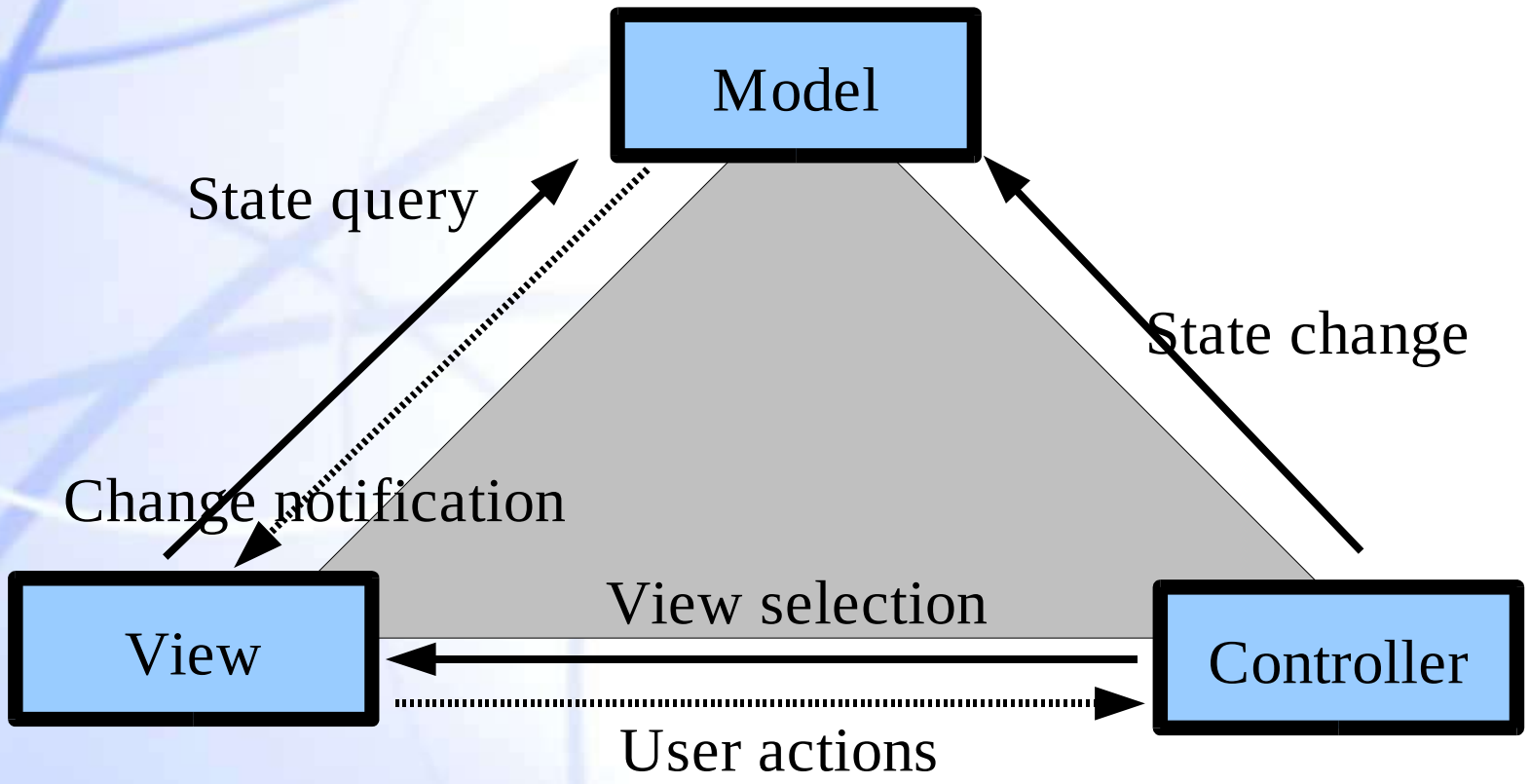
- Entorno para la construcción de aplicaciones web.
- Basado en el patrón Modelo-Vista-Controlador (MVC)

  Model 1: JSP-centric

  Model 2: Struts

- http://jakarta.apache.org/struts

Introducción

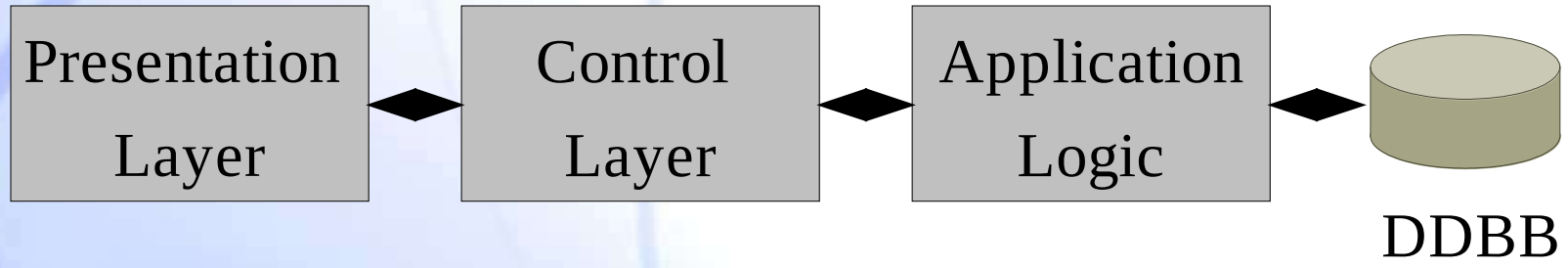


Introducción

- Desacopla la "vista" del "modelo"
- El "controlador" selecciona:

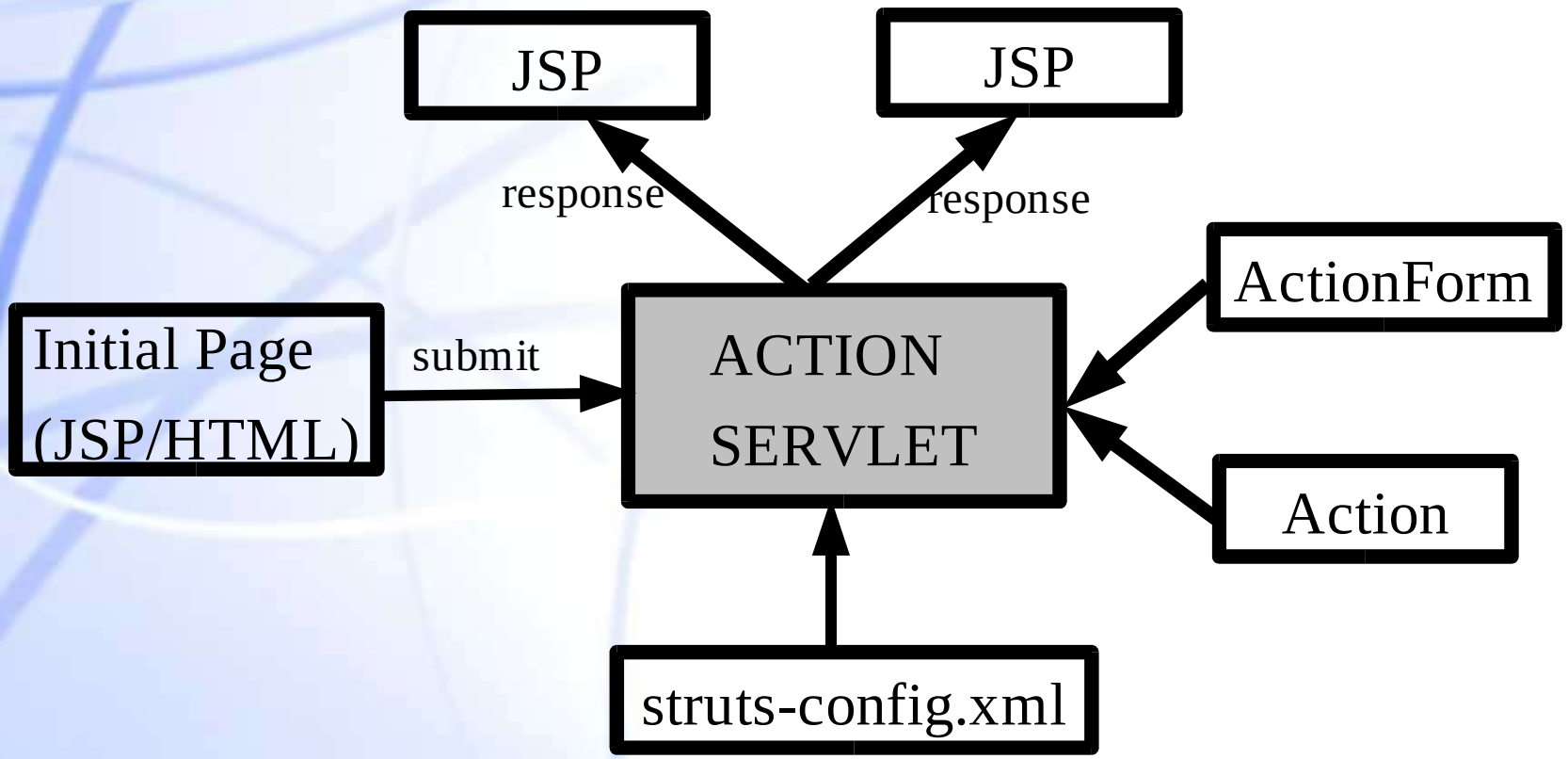
  Los datos.

  La "vista" que visualiza los datos.

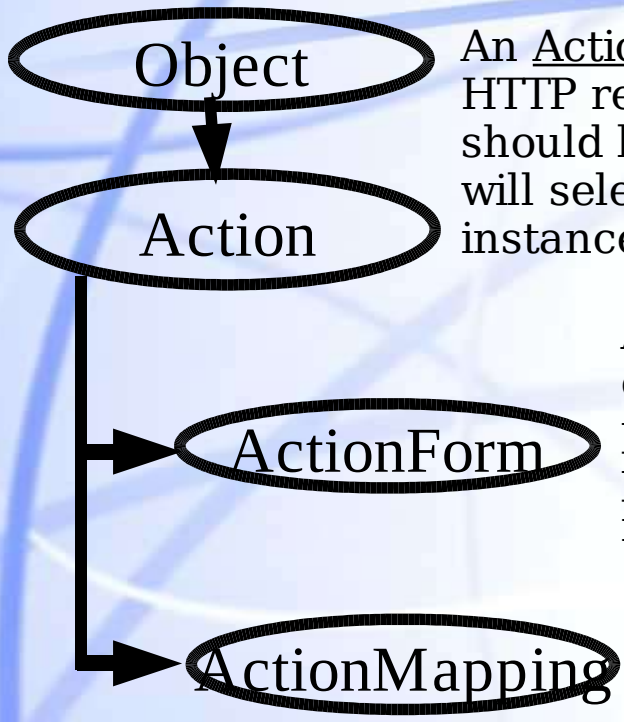


Struts Framework

Struts Framework



Clases

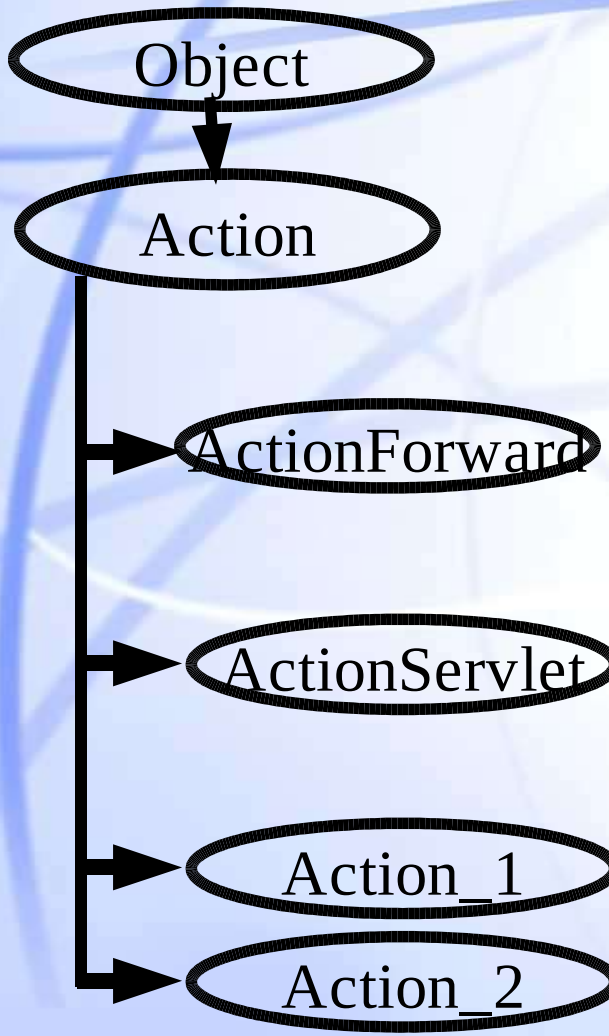


An Action is an adapter between the contents of an incoming HTTP request and the corresponding business logic that should be executed to process this request. The controller will select an appropriate Action for each request, create an instance (if necessary), and call the execute method.

An ActionForm is a JavaBean optionally associated with one or more ActionMappings. Such a bean will have had its properties initialized from the corresponding request parameters before the corresponding Action.execute method is called.

An ActionMapping represents the information that the controller knows about the mapping of a particular request to an instance of a particular Action class. The ActionMapping instance used to select a particular Action is passed on to that Action, thereby providing access to any custom configuration information included with the ActionMapping object.

Clases



An ActionForward represents a destination to which the controller might be directed to perform a “forward” or “redirect” as a result of processing activities of an Action class. Instances of this class may be created dynamically as necessary, or configured in association with an ActionMapping instance for named lookup of potentially multiple destinations for a particular mapping instance (if necessary), and call the execute method.

ActionServlet provides the "controller" in the Model-View-Controller (MVC) design pattern for web applications that is commonly known as "Model 2".

User defined actions

Resumen

Clase	Descripción
ActionForward	Datos para realizar un cambio de estado
ActionForm	Los datos asociados a un cambio de estado
ActionMapping	Evento de cambio de estado
ActionServlet	El controlador que recibe peticiones de usuario, cambios de estado y reeenvía las vistas seleccionadas
Action	La parte del controlador que interacciona con el modelo para ejecutar los cambios de estado y comunica al ActionServlet la vista seleccionada.

Ejemplo de uso

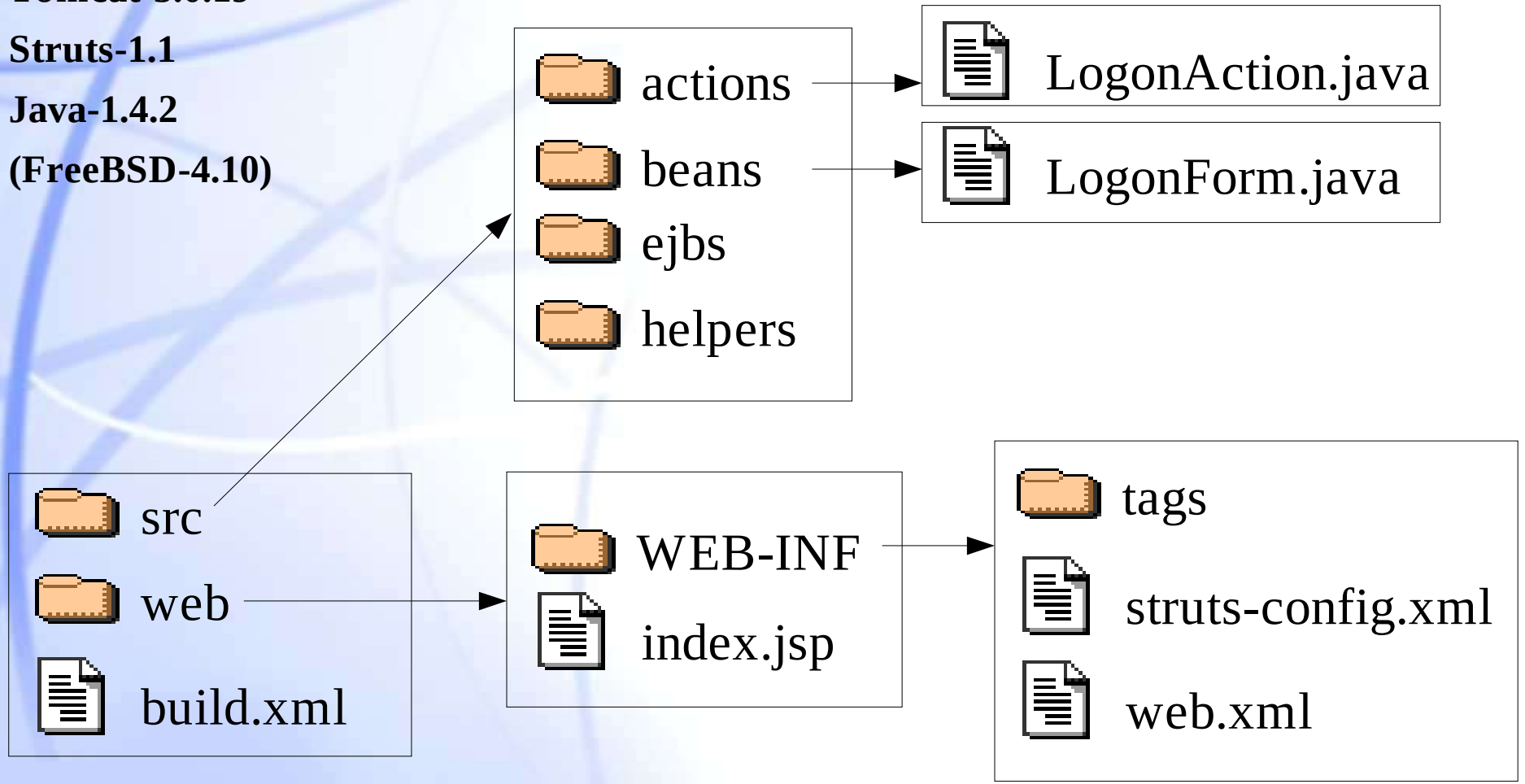
Ejemplo de uso

Tomcat-5.0.19

Struts-1.1

Java-1.4.2

(FreeBSD-4.10)



LogonAction.java

```
public class LogonAction extends Action {

public ActionForward execute (ActionMapping mapping,   ActionForm form,
                              HttpServletRequest req, HttpServletResponse res) {

    LogonForm lf= (LogonForm) form;

    String username = lf.getUsername();

    String password = lf.getPassword();
```

LogonAction.java

```
...
if( gestor.doLogin( username, password ) == false ) {
        return mapping.findForward("failure");
} else {
        ....
        return mapping.findForward("administrador");
        ....
}
}  /* execute() */
} /* class */
```

LogonForm.java

```
public class LogonForm extends ActionForm {

    protected String username;

    protected String password;


    public String getUsername(){return this.username;};

    public String getPassword(){return this.password;};


    public void setUsername(String username){this.username=username;};

    public void setPassword(String password){this.password=password;};

}
```

index.jsp

```
<%@ taglib uri="/WEB-INF/tags/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/tags/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/tags/struts-logic.tld" prefix="logic" %>
<html><body><center>
<html:form action="/LogonSubmit.do" focus="username">
<table border="0" width="100%">
<tr> <td align="center">Username:<html:text property="username"/></td></tr>
<tr> <td align="center">Password:<html:text property="password"/></td></tr>
<tr> <td align="center"><html:submit/><html:reset/></td></tr>
</table>
</html:form></center></body></html>
```

struts-config.xml

```
<struts-config>
  ...
  <form-beans>
    <form-bean      name="logonForm"
                type="beans.LogonForm">
    <form-property name="username" type="java.lang.String"/>
    <form-property name="password" type="java.lang.String"/>
    </form-bean>
  </form-beans>
  ...
```

struts-config.xml

```
...
<action-mappings>
 <action    path="/LogonSubmit"
            type="actions.LogonAction"
            name="logonForm"
           scope="session">
    <forward name="failure" path="/index.jsp"/>
  </action>
</action-mappings>
 ...
</struts-config>
```

Conclusiones

Conclusiones

- **Puntos fuertes:**

  Basado en patrones de diseño (MVC, layers...)

  Potente conjunto de librerias JSP (tag-libraries).

  “http-centric”, “model-neutral”.

- **Puntos débiles:**

  No hay modelo de eventos.

  Sólo un “ActionServlet” por aplicación web.

  “model-neutral”, “learning-curve”, “nomenclature”

Bibliografía

- [1] T. Husted, “Struts in Action”, Manning Publications, 2003
- [2] M. Floyd, “EJB design patterns: advanced patterns, processes and idioms”, NY, John Wiley & Sons, 2002
-
- [3] http://jakarta.apache.org/struts