



Hauptseminar Informatik

Java Web Frameworks

14.06.2002

Alexander Löser



Gliederung

- Frameworks
- Model View Controller
- Struts
- Turbine
- Barracuda
- Espresso
- Fazit

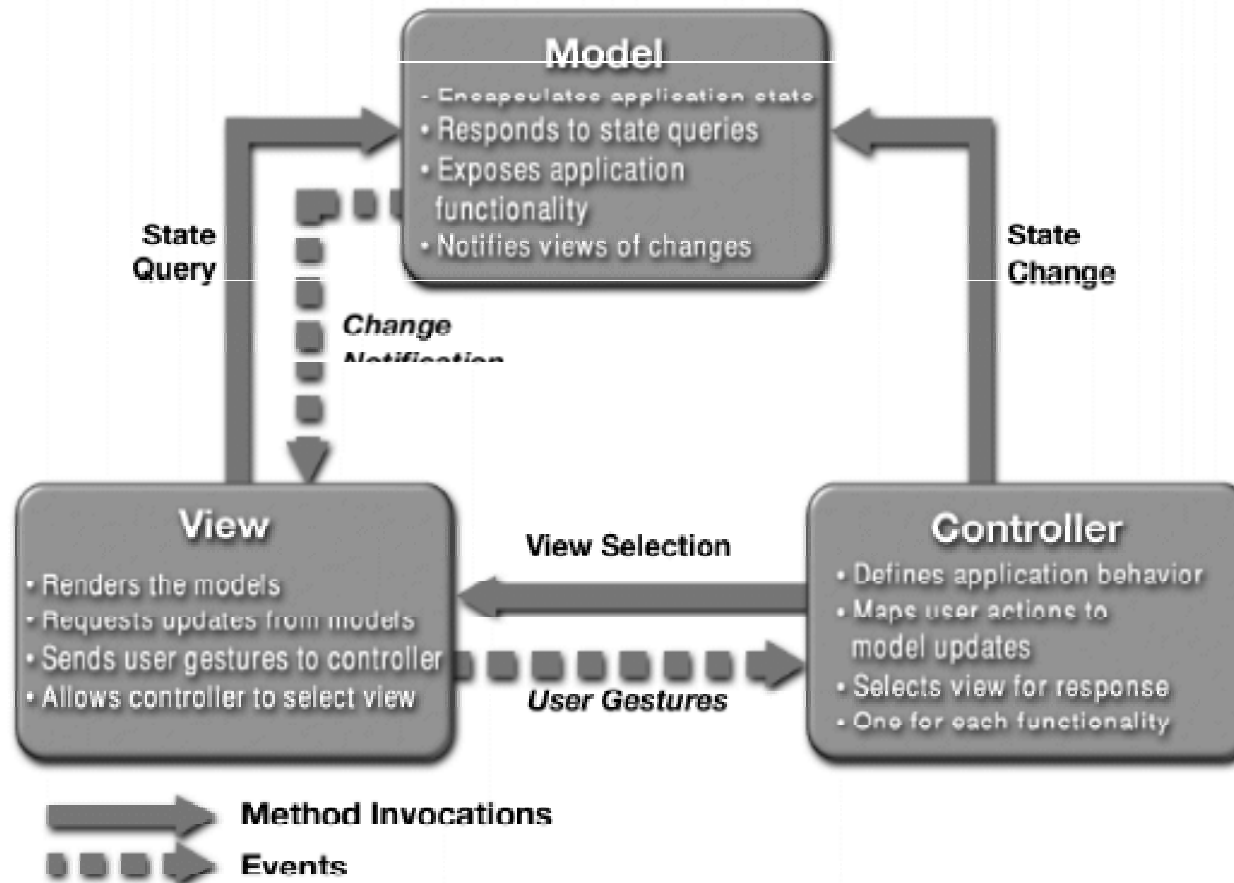


Frameworks

- Wiederverwendung von:
 - Analyse
 - Design
 - Code
- Implementierung von Design Patterns (Umsetzungsempfehlungen)



Model View Controller (MVC)



[Quelle: http://java.sun.com/blueprints/patterns/j2ee_patterns/model_view_controller/index.html]

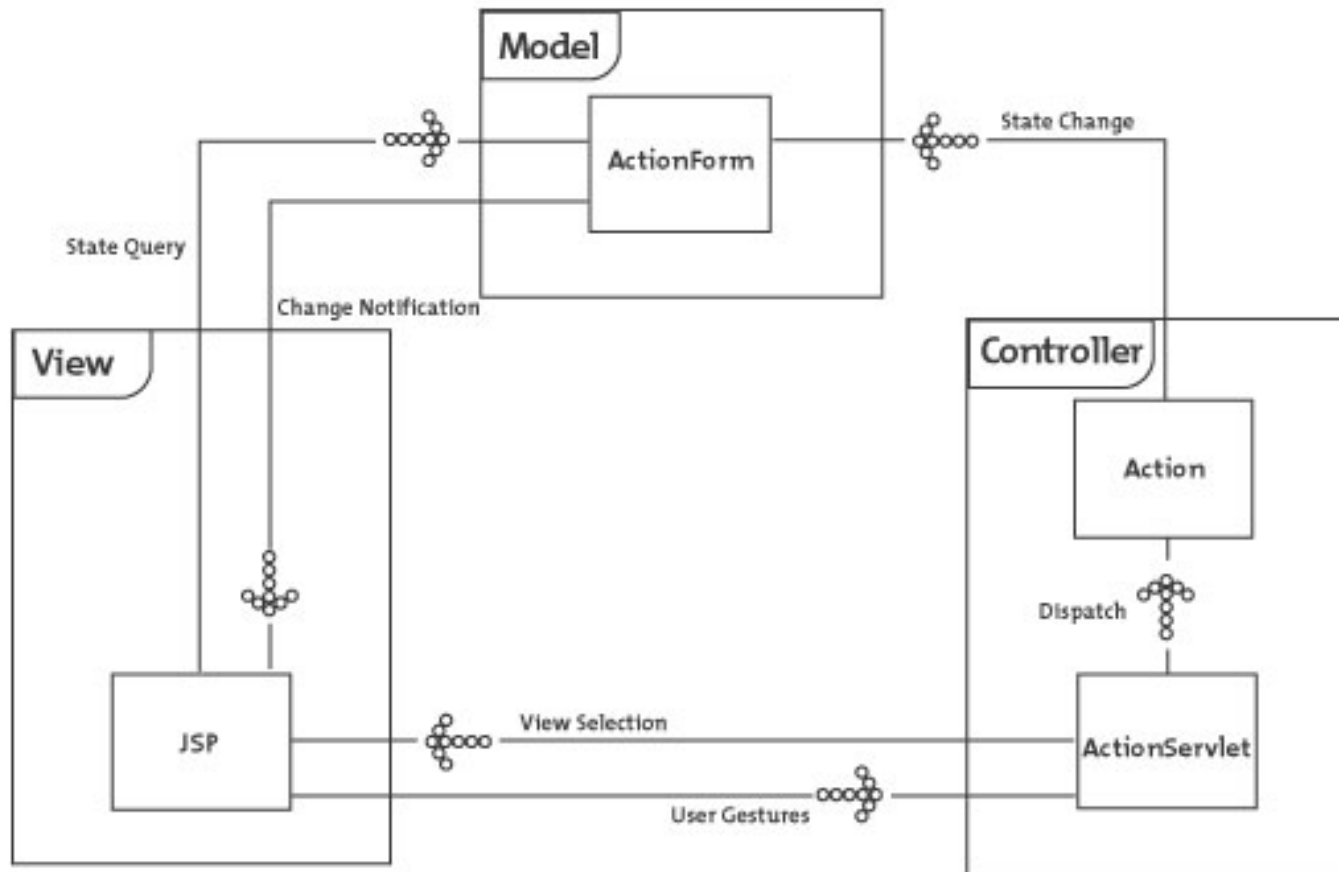


Struts – Überblick

- Aus dem Open Source Projekt Jakarta
- Model 2 MVC
 - Trennung von View und Controller
 - Im Gegensatz zu Model 1 MVC
- View-Gestaltung mit Java Server Pages (JSPs)



Struts - Architektur (1/4)



[Quelle: <http://www.sys-con.com/java/archives3/0610/sweeting/fig2.jpg>]

Hauptseminar Informatik: Java-Web-Frameworks (Alexander Löser)



Struts - Architektur (2/4)

- Zentrales ActionServlet nimmt Requests entgegen
- Request-Mapping auf Actions (struts-config.xml)

```
<struts-config>
  <action-mappings>
    <action
      path="/logon" type="org.example.LogonAction"
      name="logonForm" scope="request">
      <forward name="failure" path="/error.jsp"/>
      <forward name="success" path="/subscription.jsp"/>
    </action>
  </action-mappings>
</struts-config>
```



Struts - Architektur (3/4)

- **Action**
 - Geschäftslogik ausführen
 - Singleton -> müssen threadsafe sein
 - Model füllen
 - Rückgabe: ActionForward -> neue View bestimmen



Struts - Architektur (4/4)

- ActionForm
 - getXXX()
 - setXXX()
 - validate()
 - reset()
- View
 - JSP + Tag Libraries

```
<html:form action="/logon" focus="username">  
  <html:text property="username" size="16"/>  
  ...  
  <html:password property="password" size="16"/>  
  ...  
  <html:submit/>  
</html:form>
```



Struts - Fazit

- Relativ einfach
- Trennung Inhalt/Layout/Logik
- Aber: kein Zwang
- Keine Unterstützung für Dialoge

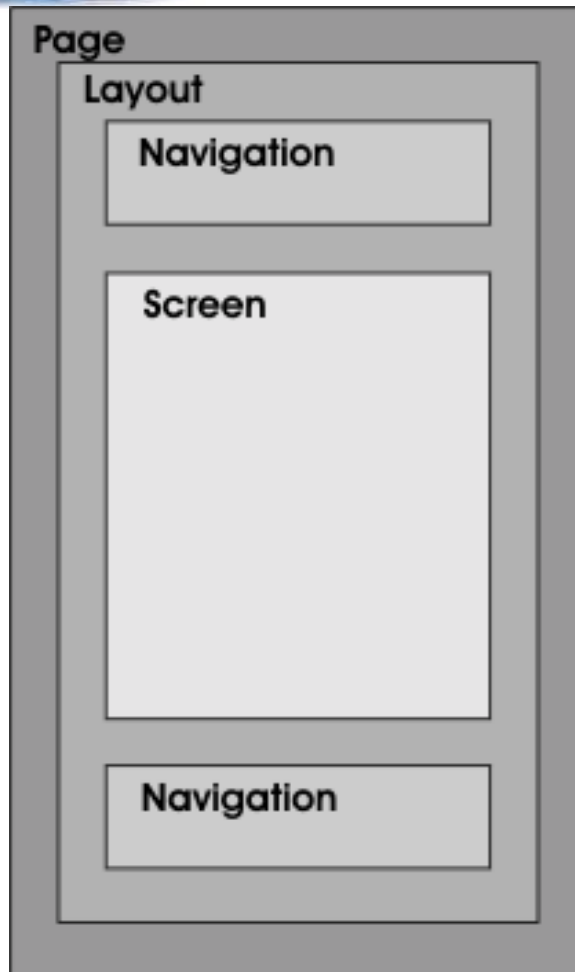


Turbine - Überblick

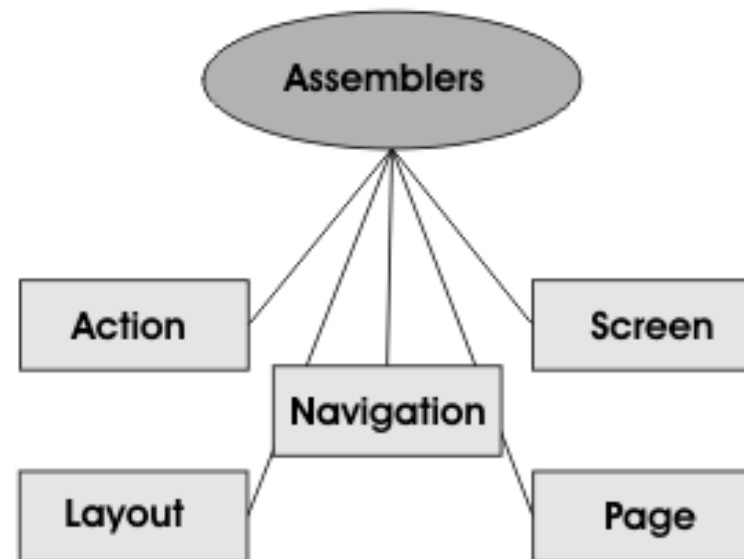
- Ebenfalls aus Jakarta Projekt
- Komplexere Architektur
- Turbine Development Kit (TDK)
- Zusätzliche Services verfügbar
- Template-basierte Views
- Model 2+1MVC



Turbine - Module (1/3)



[Quelle: <http://jakarta.apache.org/turbine/turbine-2/fsd.html>]





Turbine – Module (2/3)

- **5 Module:**
 - **Actions**
 - **Business-Logik**
 - **Evtl. EJB-Aufrufe**
 - **Aufruf durch Page-Modul**



Turbine - Module (3/3)

- Page (Container)

- Layout
- Screen
- Navigation



Template + Java-Klasse

- Template-Engine: Velocity

```
#formCell ("User" "username" "")  
<input type="text" size="30" name="$name"  
value="$value">
```

- Module werden nacheinander ausgeführt um HTML-Seite zurückzuliefern
- Ext. Aufrufe möglich (z.B. EJB)



Turbine - Services

- **Security-Services (ausgeklügeltes Benutzerkonzept)**
- **OR-Mapping (Torque, Castor)**
- **Template-Unterstützung (Velocity, Freemarker, WebMacro)**
- **Logging-Services**
- **Pool-Service, Naming-Service u.v.m.**



Turbine - Fazit

- **Komplettes Basisframework**
- **Nicht primär auf J2EE ausgerichtet**
- **Aber: vielfältige Integrationsmöglichkeiten im Frontendbereich**
- **Einsatz im Portal-Framework Jetspeed**

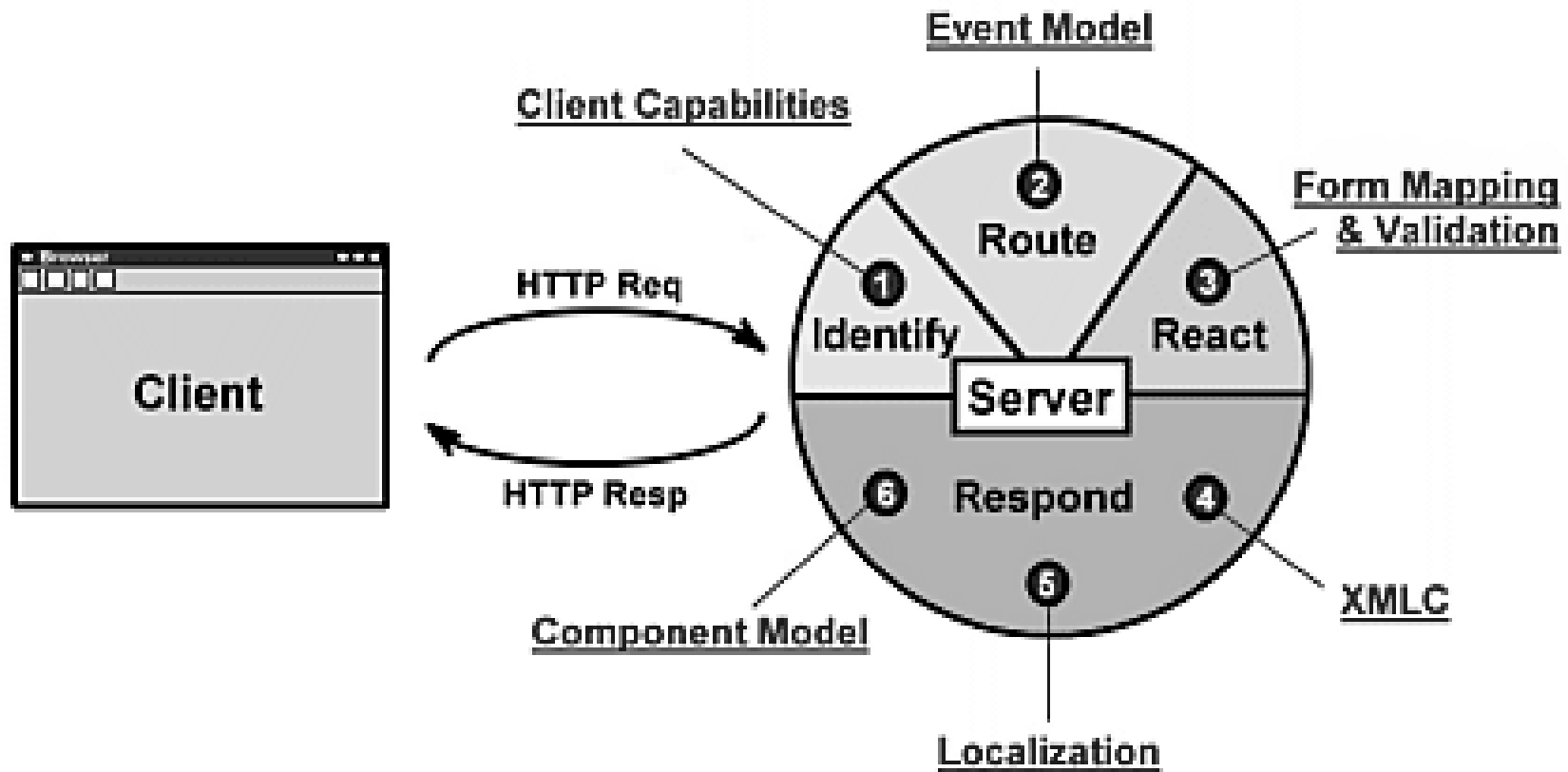


Barracuda – Überblick

- **Enhydra.org Projekt (Lutris Tech.)**
- **Model 2 MVC**
- **Push-Ansatz (XMLC)**



Barracuda – Komponenten





Barracuda - XMLC

- Generierung von Java-Klassen aus HTML / XML
- Exakte Kopie als DOM
- Zugriff auf Tags über Attribute
`<div Id=„uniqueid“ Class=„groupname“> ...`
- Localization möglich
- Nach Manipulation zurückliefern der HTML-Seite



Barracuda – Events (1/2)

- Serverseitiges Event-Modell
 - Request-Events (-> Controller)

```
class AttemptLoginHandler
    extends DefaultBaseEventListener
{
    public void handleControlEvent
        (ControlEventContext con) throws EventException...
    { ...
        if (valid==LoginServices.VALID) {
            throw new ClientSideRedirectException
                (new GetMainScreen());
        } ...
    }
}
```



Barracuda – Events (2/2)

– Response-Events

```
class RenderLoginScreenHandler
    extends BTemplateViewHandler
{
    public TemplateModel getTemplateModel()
        {return new LocalTemplateModel();}
    public Class getTemplateClass()
        {return LoginHTML.class;}
}
```



Barracuda – Fazit

- Saubere Rollen-Trennung zwischen Web-Designer und Programmierer
- Vergleichsweise hoher Lern- und Implementierungsaufwand



Expresso – Überblick

- Open Source Framework von Jcorporate Ltd.
- Ähnlich komplett ausgestattet wie Turbine
- Integration Struts



Expresso – Komponenten (1/2)

- Kernkomponenten

- Schema

- beschreibende Java-Klasse

```
addDBObject("org.example.MyBussinessDBObject" );  
addController("org.example.MyController" );
```

- DatabaseObjects

- Einfaches OR-Mapping

```
setTargetTable("MyTable");  
addField("ST_ID", "int", 0, false,  
        "Text Channel Autoincrement ID");  
...  
addKey("ST_ID");
```




Expresso – Komponenten (2/2)

– Controller

- Von Struts Actions abgeleitet
- Endlicher Automat

```
addState(new State("displaySomething",  
                  "Display Something"));
```

```
protected ControllerResponse  
    runDisplaySomethingState(...) {...}
```

- Führen bei Zustandswechsel Actions aus
- Muss Aufgaben von MVC-Controller und Model übernehmen



Expresso – Funktionen

- Security-Konzept
- Weitere Funktionen
 - Logging
 - Caching
 - Connection Pooling
 - U.v.m



Expresso – Fazit

- **Keine strikte Trennung von Model und Controller**



Fazit – Vergleich

	Struts	Turbine	Barracuda	Expresso
MVC	Pull Model 2	Model 2+1	Push Model 2	Model 2
Model	ActionForm	Beans	Components	DBObjects
View	JSP / Servlet	Templates / (JSP möglich)	XMLC	JSP / Templates / XML/XSLT
Controller	Zentr. Servlet + Action Mapping	Zentr. Servlet + Actions, Screens	Eventsteuerung	Zentr. Servlet
Zus. Services	nein	ja	(nein)	ja



**Vielen Dank für die
Aufmerksamkeit!**