

Vergleich der Model 2 Umsetzung von Struts und Cocoon

Tobias Kieninger

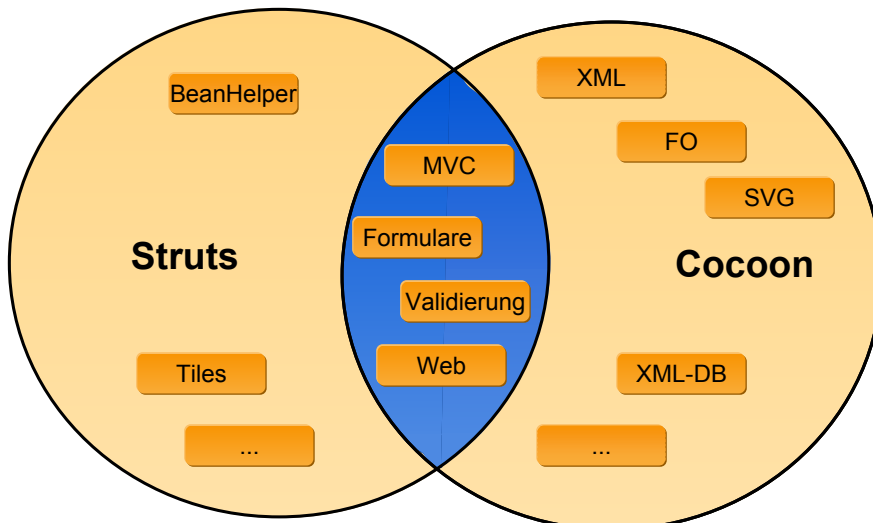
Kieninger@oio.de

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de

Struts



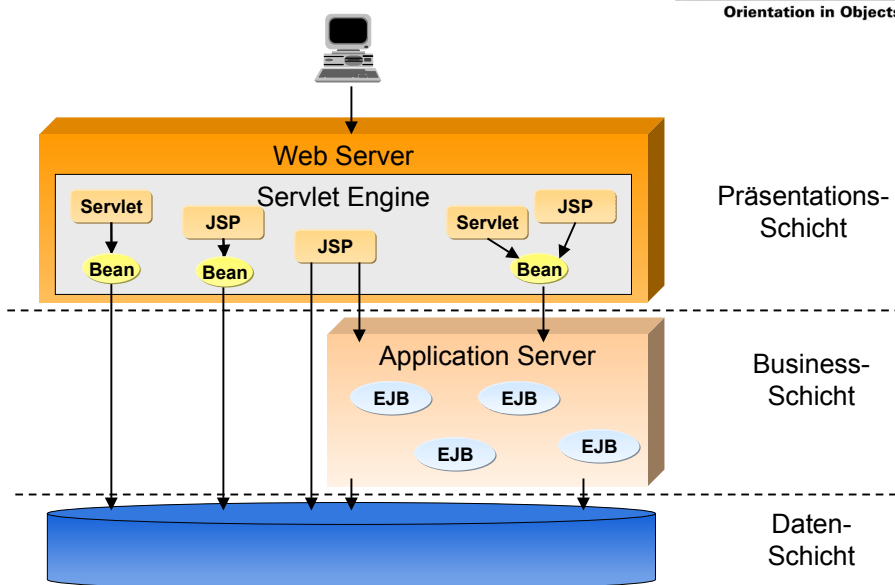
Ziele der Session



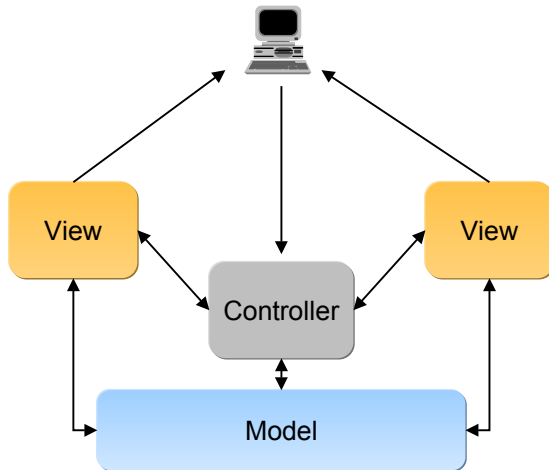
Inhalt

- Model 1, Model 2, MVC, Front Controller...
- Struts
 - Formularhandling
- Cocoon
 - XMLForm
 - Flows & Continuations
- Gegenüberstellung

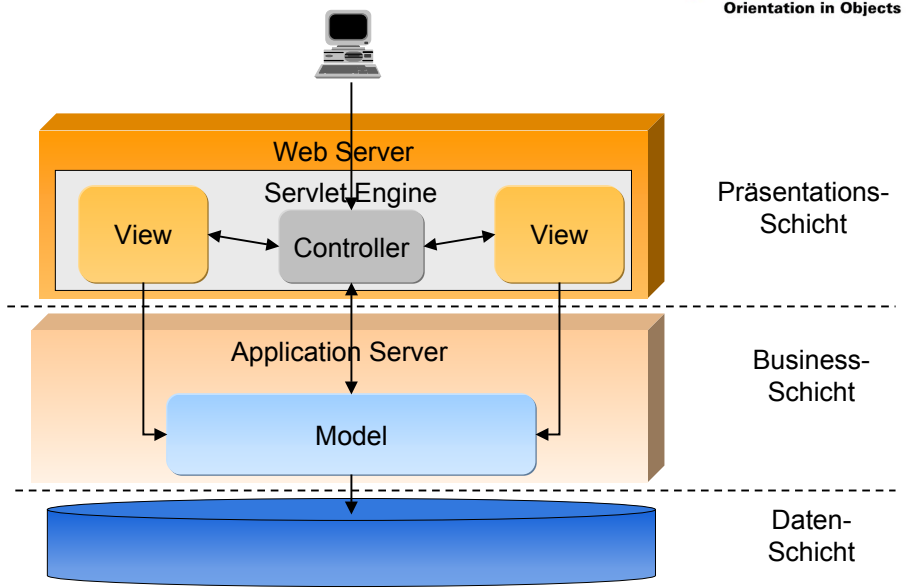
J2EE - Model 1



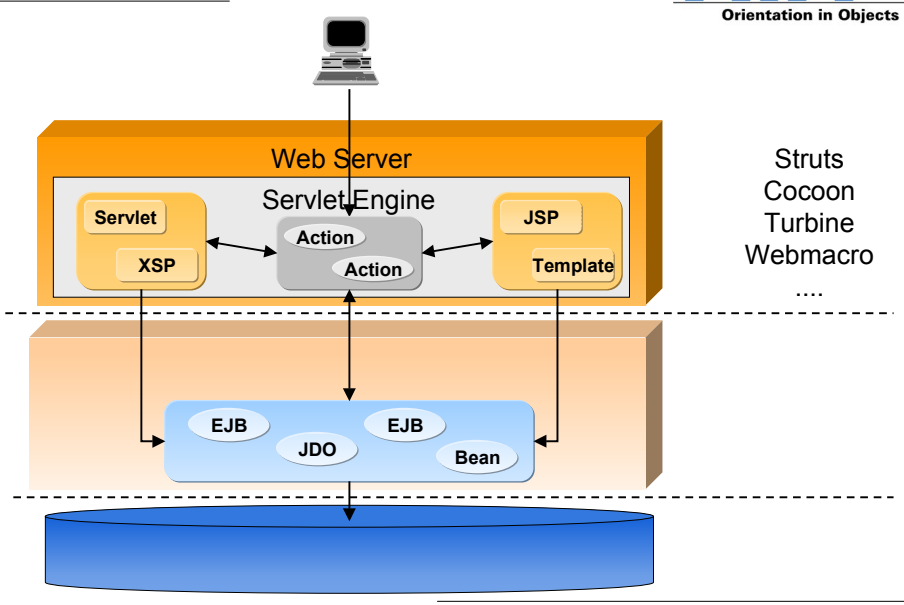
Model View Controller



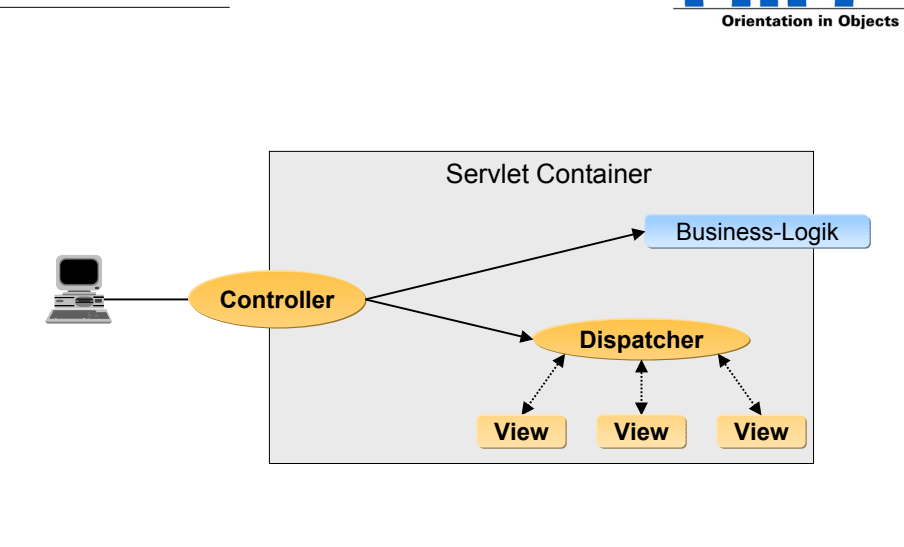
J2EE - MVC im Webbereich



Der Trend? Frameworks!



Model 2, Front Controller, Dispatcher View



Quelle: J2EE-Patterns (Bien)

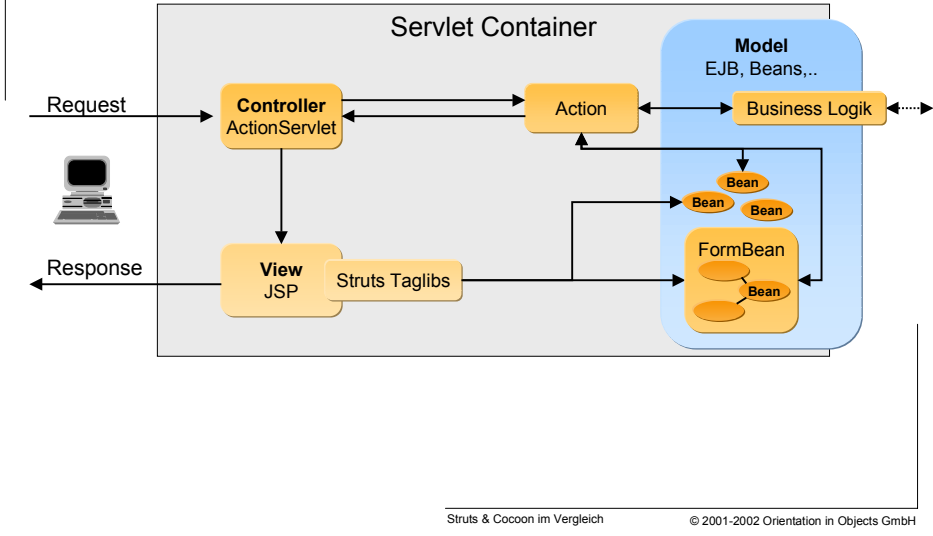
Inhalt

- Model 1, Model 2, MVC, ...
- **Struts**
 - Formularhandling
- Cocoon
 - XMLForm
 - Flows & Continuations
- Gegenüberstellung

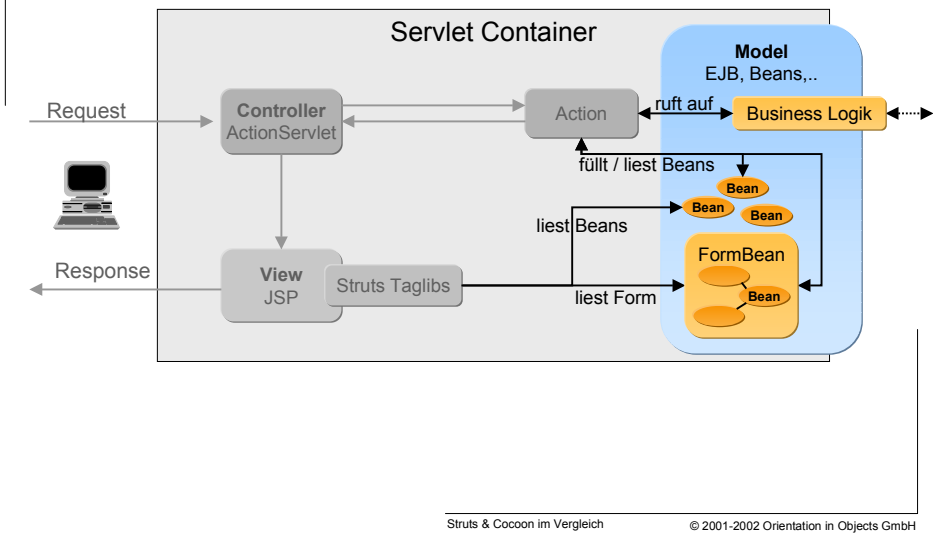
Was ist Struts?

- Open Source Framework für Web Anwendungen
- **Umsetzung diverser J2EE Muster**
- Unter der Lizenz der Apache Software Foundation
- Basiert auf Standardtechnologien
 - JSP, JavaBeans, ResourceBundle, ...
- Ermöglicht den Model-View-Controller Ansatz
- Teil des Apache Jakarta Projektes
 - <http://jakarta.apache.org/struts>

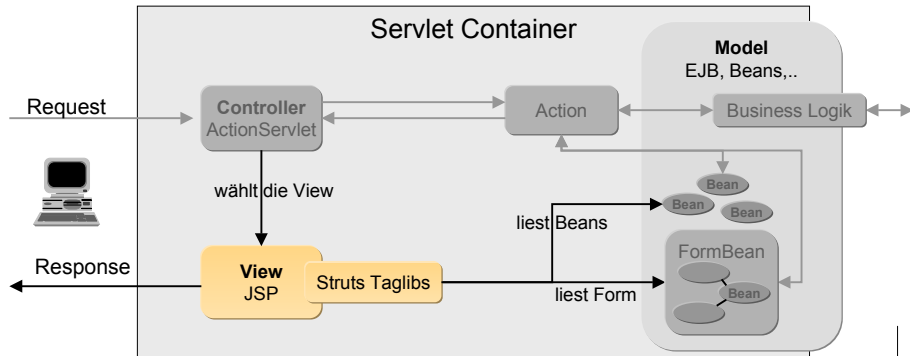
Struts allgemein



Struts Model



Struts View



JSP mit Struts Tags

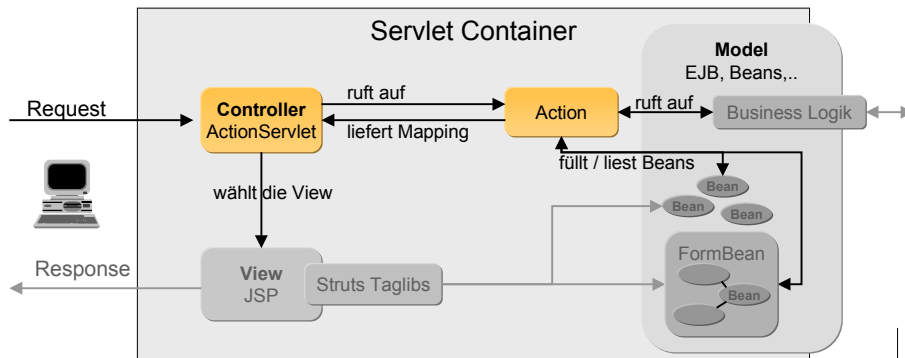
```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<h1><%=request.getAttribute („hello“) %>!</h1>

<html:errors/>

<table>
  <logic:iterate id="aPage"
                name="pages" type="de.oio.util.web.Page">
    <tr>
      <td><bean:write name="aPage" property="host"/></td>
      <td><bean:write name="aPage" property="head.title"/></td>
    </tr>
  </logic:iterate>
</table>
```

Struts Controller



Action

```
public class HelloWorldAction extends Action {

    public ActionForward execute(ActionMapping mapping,
        ActionForm form, HttpServletRequest request,
        HttpServletResponse response) throws Exception {

        // Request verarbeiten / Aufruf der Business Logik
        ...

        // Ergebnis publizieren
        request.setAttribute("hello", "world");
        session.setAttribute(...);
        application.setAttribute(...);

        // Nächste Seite wählen
        return (mapping.findForward("success"));
    }
}
```

execute() für Version 1.1
perform() für Version 1.0

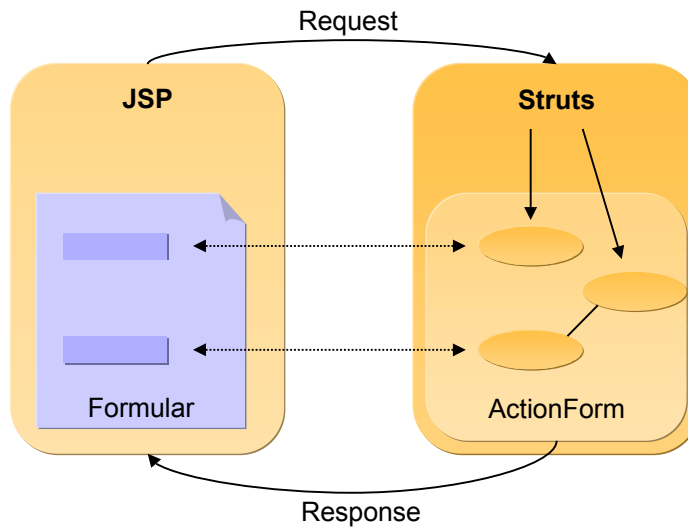
Konfiguration der Action

```
<struts-config>
...
<action-mappings>
  <action path="/HelloWorld" validate="false"
    type="HelloWorldAction">
    <forward name="success" path="/HelloWorld.jsp"/>
  </action>
</action-mappings>
</struts-config>
```

Inhalt

- Model 1, Model 2, MVC, ...
- Struts
 - Formularhandling
- Cocoon
 - XMLForm
 - Flows & Continuations
- Gegenüberstellung

Struts Formularhandling



Beispiel ActionForm Klasse

```
public class LoginForm extends ActionForm {  
  
    private String passwd, username;  
  
    public String getUsername() { return username; }  
    public void setUsername(String aUser) { user = aUser; }  
  
    ...  
  
    public void reset(ActionMapping mapping,  
                     HttpServletRequest request)  
    { username = ""; passwd = ""; }  
  
    public ActionErrors validate(ActionMapping mapping,  
                                HttpServletRequest request)  
    {  
        ActionErrors errors = new ActionErrors();  
        return errors;  
    }  
}
```

Entwurf der View

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html>
...
  <html:form action="login.do" method="get">
    <html:text property="user"/><br/>
    <html:password property="passwd"/><br/>
    <html:submit property="submit" value="Login"/><br/>
  </html:form>
...
</html>
```

Die Action

```
public class LoginAction extends Action
{
  public ActionForward perform(ActionMapping mapping,
    ActionForm form, HttpServletRequest request,
    HttpServletResponse response) {

    if(form instanceof LoginForm)
    {
      LoginForm lf = (LoginForm) form;
      if("qwertz".equals(lf.getPasswd()))
      { return mapping.findForward("home"); }
      else
      { return mapping.findForward("login"); }
    }
  }
}
```

Fazit Struts?

- Produktivere Entwicklung und Wartung von Web Anwendungen
- Einfache Validierung
- Struts kombiniert:
 - JSPs
 - Servlets
 - Custom Tags
 - Message Ressourcen
- Unterstützt Entwicklung im Team



- Klar
- Strukturiert
- Geradlinig

Inhalt

- Model 1, Model 2, MVC, ...
- Struts
 - Formularhandling
- **Cocoon**
 - XMLForm
 - Flows & Continuations
- Gegenüberstellung

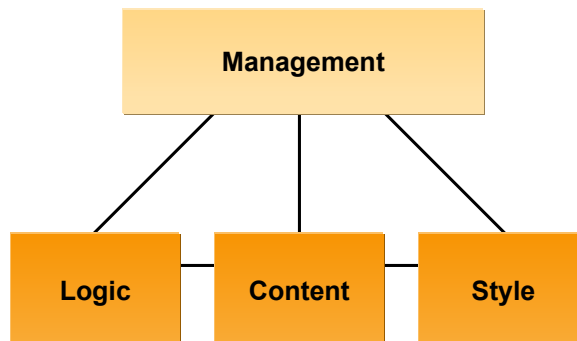
Cocoon

Apache Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as XML, XSL, SVG, etc..) to provide web content.

Quelle: Cocoon 2 README

- Unter der Apache Software License
- <http://xml.apache.com/cocoon/>

Separation of Concerns (SoC)

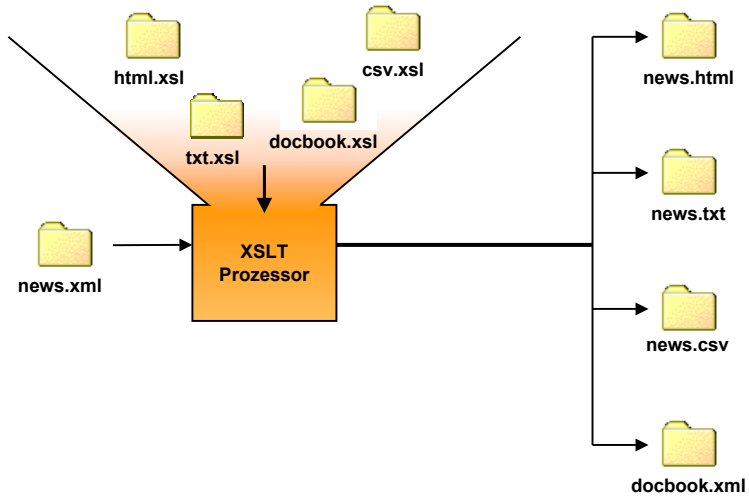


Quelle: Cocoon Dokumentation

XSLT Prozessor



Orientation in Objects



Transformation mit XSLT



Orientation in Objects

```
...  
<cd>  
  <titel>Music</titel>  
  <interpret>Madonna</interpret>  
</cd>  
...
```

XML-Quelle

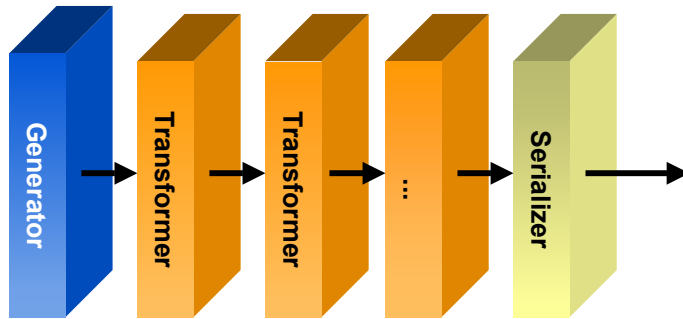
```
...  
<xsl:template match="cd">  
  <b><xsl:value-of select="titel"/></b>,  
  Interpret: <xsl:value-of select="interpret"/><br/>  
</xsl:template>  
...
```

XSL-Stylesheet

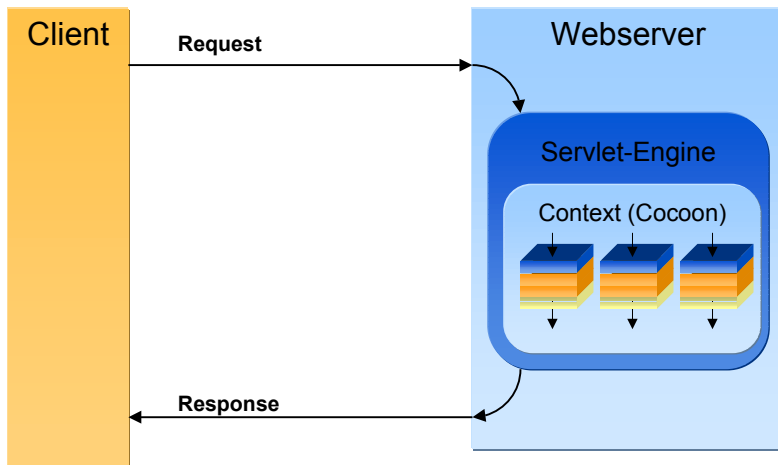
```
<b>Music</b>, Interpret: Madonna<br/>
```

Output

Pipeline Model



HTTP mit Pipeline



HelloWorld in 3 Schritten - 1

Die XML-Quelle (hello.xml)

```
<?xml version="1.0"?>
<page>
  <title>Erstes Beispiel - OIO Seminar</title>
  <greeting>Hello World</greeting>
</page>
```

HelloWorld in 3 Schritten - 2

Das XSLT-Stylesheet

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:template match="page">
    <html>
      <body>
        <h1><xsl:value-of select="title"/></h1>
        <p><xsl:value-of select="greeting"/></p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```


HelloWorld in 3 Schritten - 3

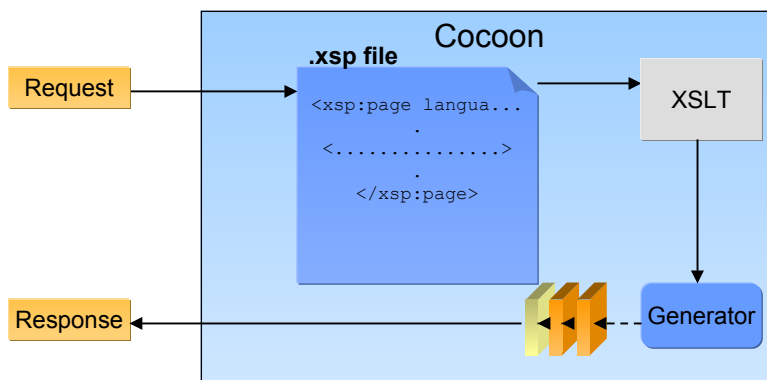
Definiert die Pipeline, die zu einem Seitenaufruf gehört

`http://localhost:8080/cocoon/hello/hello.html`

Auflösung durch Sitemap

```
<map:match pattern="hello/hello.html">
  <map:generate type="file" src="hello/hello.xml"/>
  <map:transform src="hello/hello.xslt"/>
  <map:serialize type="html"/>
</map:match>
```

eXtensible Server Pages



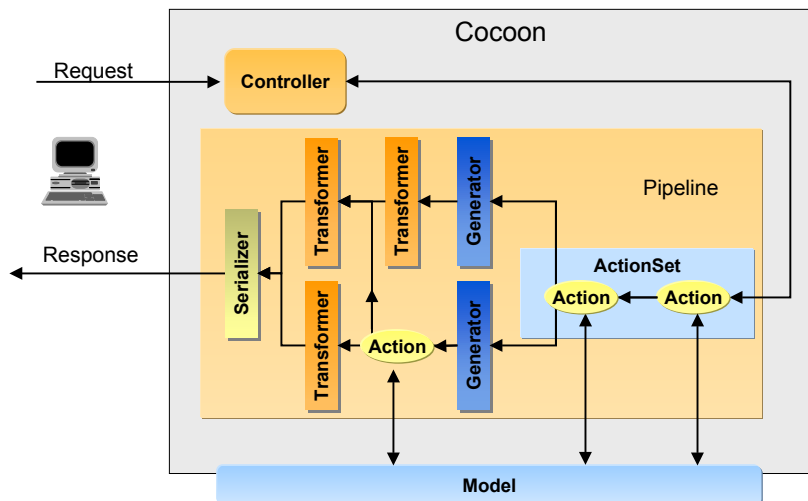
XSP - Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  ...

  public BigInteger getBig() {
    return new BigInteger("10000000000000000000");
  }

</xsp:logic>
<page>
  <xsp:expr>getBig()</xsp:expr>
  <xsp-request:get-attribute name="hello"/>
</page>
</xsp:page>
```

Cocoon Actions



Action

```
public class HelloWorldAction extends AbstractAction {  
  
    public Map act (...) {  
        // Request verarbeiten / Aufruf der Business Logik  
        ...  
        // Ergebnis publizieren  
        request.setAttribute("hello", "world");  
  
        // Nächste Seite wählen  
        Map sitemapParams = new HashMap();  
        sitemapParams.put("nextpage", "success");  
        return sitemapParams;  
    }  
}
```

Konfiguration der Action

```
<map:components>  
    <map:actions>  
        <map:action name="HelloWorldAction"  
            src="de.oio.HelloWorldAction"/>  
        ...  
    </map:actions>  
</map:components>  
...  
  
<map:match pattern="HelloWorld.html">  
    <map:act type="HelloWorldAction">  
        <map:generate type="serverpages" src="{nextpage}.xsp"/>  
    </map:act>  
    <map:serialize/>  
</map:match>
```

success.xsp & error.xsp

success.xsp:

```
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  <html>
    <p>Die Aktion wurde erfolgreich durchgeführt.
      Folgende Ergebnisse wurden erzielt:
      <xsp-request:get-attribute name="hello"/>
      ...</p>
  </html>
</xsp:page>
```

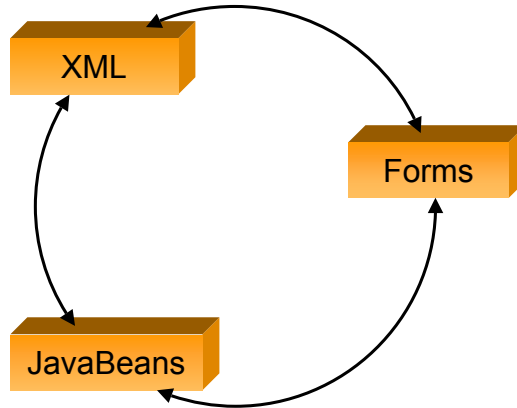
error.xsp:

```
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  <html>
    <h1>Es ist ein Fehler aufgetreten!</h1>
  </html>
</xsp:page>
```

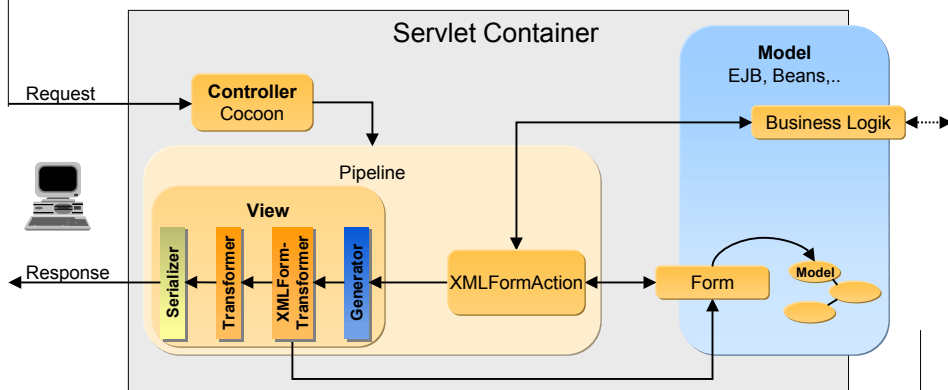
Inhalt

- Model 1, Model 2, MVC, ...
- Struts
 - Formularhandling
- Cocoon
 - XMLForm
 - Flows & Continuations
- Gegenüberstellung

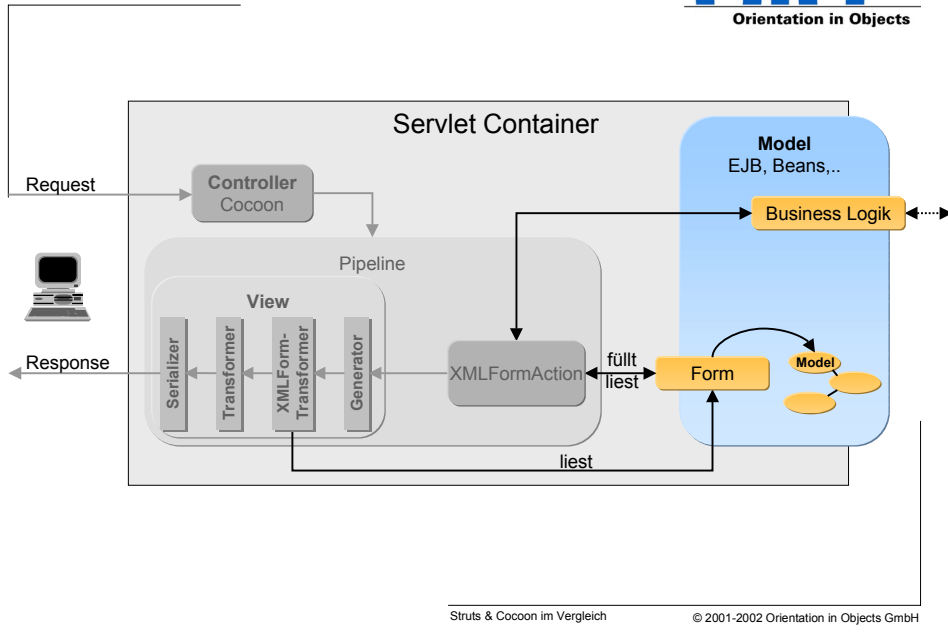
XMLForm (Alpha)



Cocoon XMLForm allgemein



Cocoon XMLForm - Model

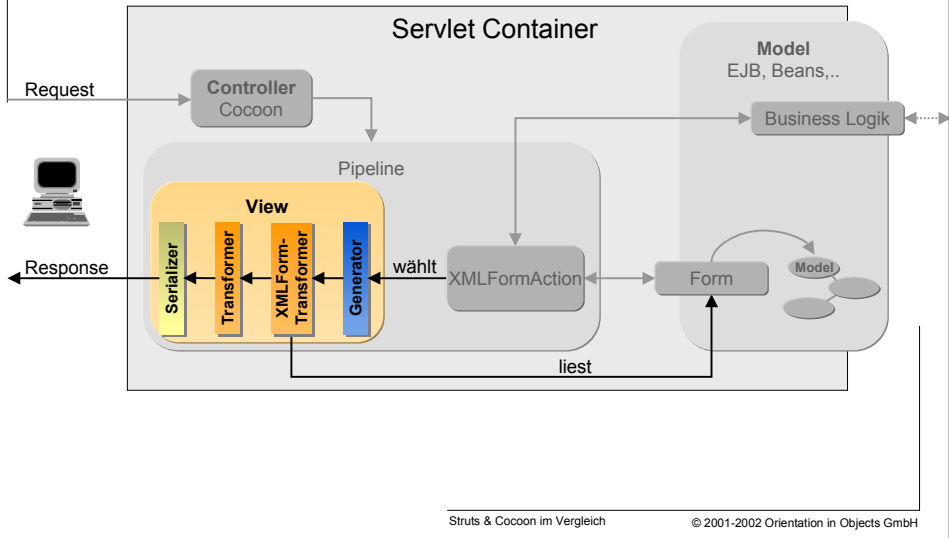


Die LoginBean (Model)

```
public class LoginBean
{
    private String user = "";
    ...

    public String getUser()
    {
        return (this.user);
    }
    public void setUser(String user)
    {
        this.user = user;
    }
    ...
}
```

Cocoon XMLForm - View



XForms

Presentation Options

XForms User Interface

XHTML

WML

Proprietary User Interfaces

XForms Model

XMLFormAction

```
public class HelloAction extends AbstractXMLFormAction
{
    public Map perform()
    {
        Form f = getForm();
        LoginBean lb = (LoginBean) f.getModel();

        if(form instanceof LoginForm)
        {
            LoginForm lf = (LoginForm) form;
            if("qwertz".equals(lf.getPasswd()))
            { return page("success"); }
            else
            { return page("login"); }
        }
    }
}
```

Fazit XMLForm

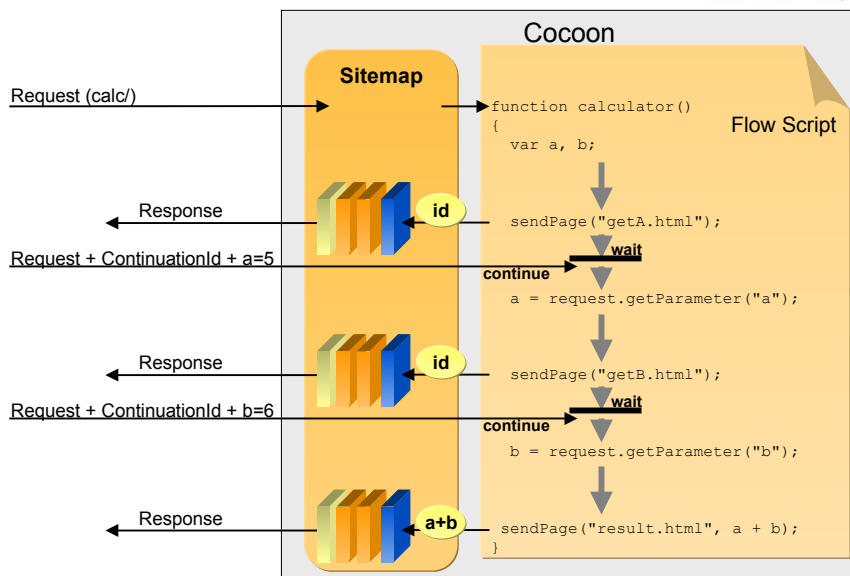
- Gute Formularbehandlung
- Erweiterung der vorhandenen Actions

- Grenzen?
 - Alphastadium
 - Dokumentation
 - Debugging

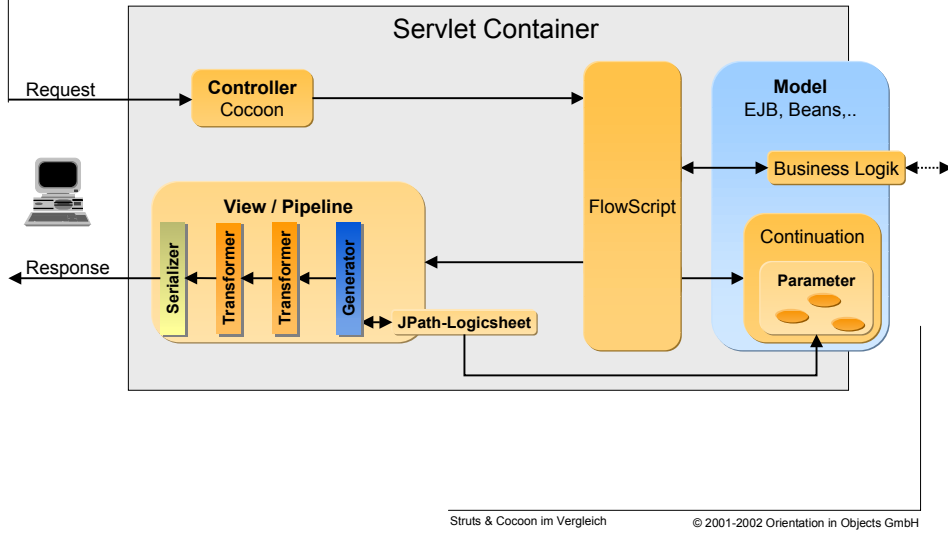
Inhalt

- Model 1, Model 2, MVC, ...
- Struts
 - Formularhandling
- Cocoon
 - XMLForm
 - **Flows & Continuations**
- Gegenüberstellung

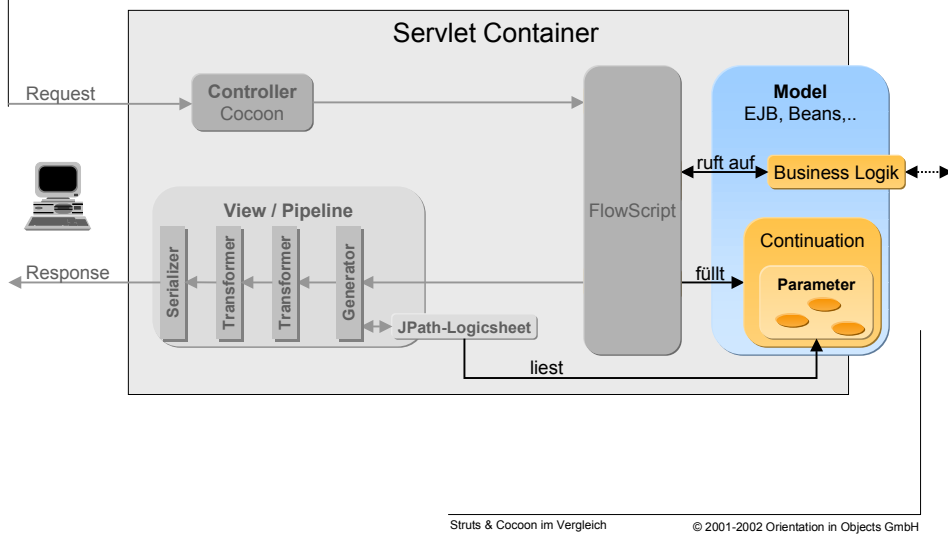
Flow Control (Alpha)



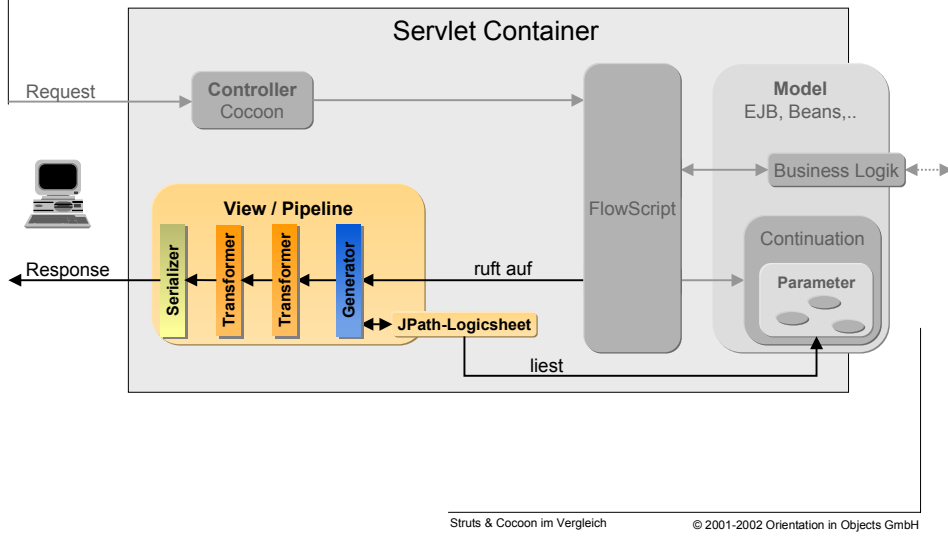
Cocoon Flow - allgemein



Cocoon Flow - Model



Cocoon Flow - View



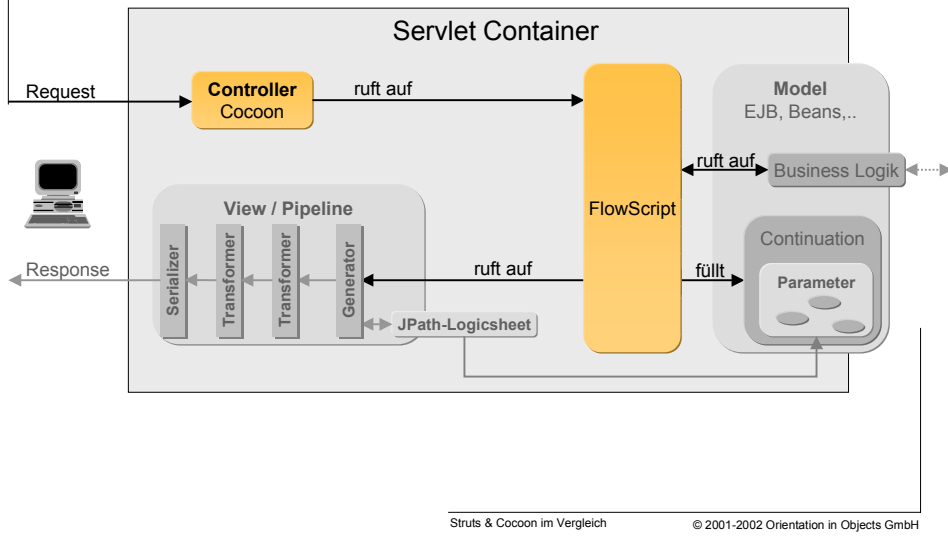
Login.xsp

```
<form>
  <xsp:attribute name="action">
    <xsp:expr>"/kont/"<jpath:continuation/></xsp:expr>
  </xsp:attribute>

  Login:
  <input type="text" name="login">
    <xsp:attribute name="value">
      <jpath:value-of select="login"/>
    </xsp:attribute>
  </input>

  ...
</form>
```

Cocoon Flow - Controller



Rhino Script

```
function login()
{
    login="";

    while (!"qwertz".equals(passwd))
    {
        sendPage("login.html", {"login" : login});

        login = cocoon.request.getParameter("login");
    }
}
```

Fazit Flows & Continuations

- Keine Actions mehr notwendig
- Leichte Programmierung des Screenflows

- Validierung von Formularen im Skript?
- Internationalisierung?
- Keine Verbindung zu XForms oder Webformularen
- Modularisierung?

Inhalt

- Model 1, Model 2, MVC, Front Controller...
- Struts
 - Formularhandling
- Cocoon
 - XMLForm
 - Flows & Continuations
- **Gegenüberstellung**

Struts

- JSP
- Taglibraries
- Struts Taglibs
- Struts Framework (Actions / Forms)

Cocoon

- XML
- XSLT / XPath
- Cocoon
- XSP
- Logicsheets
- XMLForms
 - XForms
 - Schematron
 - XMLForm Komponenten
- Continuations
 - Cocoon Flow Komponenten
 - JPath-Logicsheet
 - Rhino JavaScript

Struts Patterns

- „Synchronizer Token“ für doppelten Submit
- „Base ActionForm“ zur Wiederverwendung von Funktionen
- „Link only to Actions“ zur Abschirmung von JSPs
- „Use ActionMapping Parameter property to distinguish operations“
- Verwendung von BeanUtils zum anlegen von VOs

Quelle: <http://husted.com/struts>

Anwendung von Cocoon



“MVC is one of the possible examples of 'separation of concerns' (SoC). Since Cocoon is designed around SoC, it is entirely possible to design a web application in Cocoon using this pattern, even if, admittedly, webapp frameworks that have their focus on this are easier to use and much faster to learn than an XML-based framework.”

Stefano Mazzochi

Quelle: Bayer, Thomas im Interview mit Stefano Mazzochi
(<http://www.oio.de/public>)

Offenheit & Integrationsfähigkeit



- Beide basieren auf Servlet API
- Struts
 - PlugIn-Mechanismus ab Version 1.1
- Cocoon
 - Jede Avalon-Komponente ist austauschbar (Generator, Transformer, Serializer,...)
 - Cocoon mit idealer XML-Integration

Wartbarkeit und Erweiterbarkeit



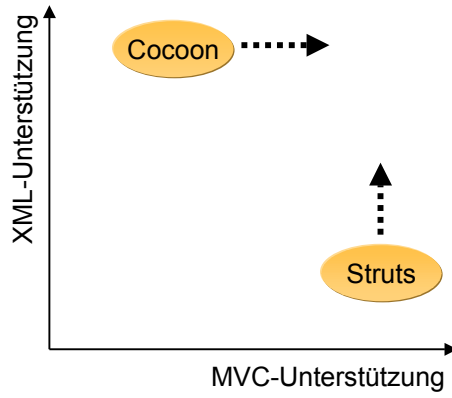
- Deklaratives Hinzuschalten von Prozessen / Actions
- Struts:
 - Seiten deklarativ austauschbar (Mappings)
- Cocoon:
 - Austausch von Avalon Komponenten
 - Anwendungsdesign ist entscheidend
 - Trennung der SoC muss klar sein
 - Welche Logik kommt XSLT zu?
 - XML muss darstellungsfrei sein
 - Actions und XSPs sollten klar abgegrenzt sein (Redirect)

Effektivität (nach Erlernen)



- Durch Modularisierung der Arbeit steigt Effektivität im Team
- Plugins für JBuilder, Eclipse, etc...
- Struts:
 - Struts kann JSP-Tools nutzen
- Cocoon:
 - Verwendung von XML-Tools
 - Kein XSP-Editor

Positionierung von Struts und Cocoon



Fragen?

