Portlets und JavaServer Faces

JavaServer Faces und Portlets verbinden

Agenda

- Überblick Portlets und Portale
  - Was sind Portale?
  - Von Portalen zu Portlets
  - Funktionsweise eines Portlets
- Überblick JavaServer Faces und Ajax
  - Was ist JSF? Was ist Ajax?
  - UI-Entwicklung mit JSF
- Kombination von JSF und Portlets (JSR-301)
  - Wo liegt die Herausforderung?
  - „Frühe“ Ansätze
  - Der JSR-301: Portlet Bridge Specification for JSF

Agenda

- **Überblick Portlets und Portale**
  - Was sind Portale?
  - Von Portalen zu Portlets
  - Funktionsweise eines Portlets
- Überblick JavaServer Faces und Ajax
  - Was ist JSF? Was ist Ajax?
  - UI-Entwicklung mit JSF
- Kombination von JSF und Portlets (JSR-301)
  - Wo liegt die Herausforderung?
  - „Frühe“ Ansätze
  - Der JSR-301: Portlet Bridge Specification for JSF

Was ist eigentlich ein „Portal“?

Portalbegriff im weiteren Sinne:

- Ein Portal ist eine Webseite, auf der unterschiedliche Informationen zu einem Themengebiet zusammengefasst sind. z.B. Anglerportal, Zauberer-Portal, Jodel-Portal
- Ein Portal ist eine zentrale Einstiegsseite in das WWW, oftmals mit redaktionell betreutem Inhalt wie bei Yahoo oder iGoogle
- „Ein Portal ist eine Webseite“

Was ist eigentlich ein „Portal“?

Portalbegriff im engeren Sinne:

- Ein Portal ist eine Webapplikation, die mindestens folgende Eigenschaften besitzt:
  - Aggregation von (verschiedenen) Inhalten
  - Einheitliche Präsentation verschiedener Dienste
  - Personalisierung
  - Single Sign On
  - Security
  - …

Was ist eigentlich ein „Portal“?

Portalbegriff im engeren Sinne

- Ein Portal ist ei… …lgende Eigenschaften b…
  - Ag…
  - Einhe…
  - Persona…
  - …

Für diese Anforderungen gibt es Softwarelösungen wie z.B. IBM WebSphere Portal, BEA Portal, JBoss Portal uvm.

Von Portalen zu Portlets

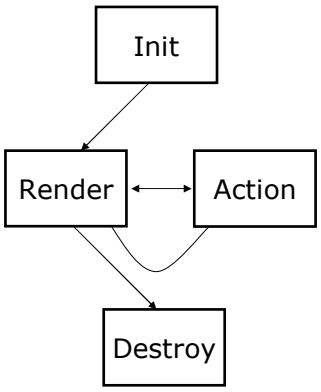
- Ein Portal **aggregiert** verschiedene Dienste, Services und Inhalte.
- Dazu wird eine Portalseite in verschiedene **Fragmente** zerlegt.
- Ein **Fragment** ist somit ein **Portlet**.

→ Portlets sind im Ggs. zu Portalen **standardisiert**. Im JSR-168 ist die Funktionsweise eine Portlets genau beschrieben.
Im aktuellen JSR-286 wurde eine weitere Version des Portletstandards erarbeitet.

Funktionsweise eines Portlets

- Ein Portlet bekommt durch den PortletContainer den Trigger, sich darzustellen (**Render-Phase**)
- Ausgaben mehrere Portlets einer Portalseite werden durch das Portal aggregiert und als (Html-)Seite ausgegeben.
- Erfolgt eine Aktion in einem Portlet, wird in diesem einen Portlet eine **Action-Phase** angestossen. Danach werden alle Portlets neu gerendert.

Lifecycle eines Portlets (JSR-168)



Der JSR-286 (Portlet V2) wird diesen Lifecycle erweitern.

Ein triviales Portlet …

```
public class HelloWorld extends GenericPortlet {

  protected void doView( ... ) {
    PrintWriter writer = response.getWriter();
    writer.println( "Hallo neue Portlet Welt" );
    writer.println( "<br><br>" );
    writer.println( "Hallo Welt !!!" );
    writer.println( "<br><br>" );
  }

}
```

Fazit Portal-/Portlet-Entwicklung

- Portale bieten bereits eine Vielzahl von Basisfunktionalitäten, um die sich eine Anwendung nicht mehr kümmern muss.
- Es gibt einen Standard für Portlets (JSR-168)
- Künftig wird es eine neue Version der Portlet-Spec geben (JSR-286)
- Dennoch ist eine Portlet-Entwicklung direkt an der API sehr aufwendig und wenig „komfortabel“.
- Ideal wäre es, die Portlet-Technologie mit einem modernen UI-Framework zu kombinieren …

Agenda

- Überblick Portlets und Portale
  - Was sind Portale?
  - Von Portalen zu Portlets
  - Funktionsweise eines Portlets
- **Überblick JavaServer Faces und Ajax**
  - Was ist JSF? Was ist Ajax?
  - UI-Entwicklung mit JSF
- Kombination von JSF und Portlets (JSR-301)
  - Wo liegt die Herausforderung?
  - „Frühe“ Ansätze
  - Der JSR-301: Portlet Bridge Specification for JSF

Was ist JavaServer Faces?

- JavaServer Faces ist ein UI-Framework
- JavaServer Faces ist komponentenorientiert
- JavaServer Faces ist ein Standard
- Für JavaServer Faces existieren mittlerweile eine (fast unüberschaubare) Anzahl von ergänzenden Komponenten, AddOns und Erweiterungen

Was bietet JavaServer Faces?

- Model-View-Controller
- Navigationskonzept
- Bean-Management
- Eventhandling
- Datenvalidierung und –konvertierung
- Große Auswahl an (OpenSource-) UI-Komponenten
- Ajax-Integrationsbibliotheken
- Templating-Engines
- …

JSF und Komponenten

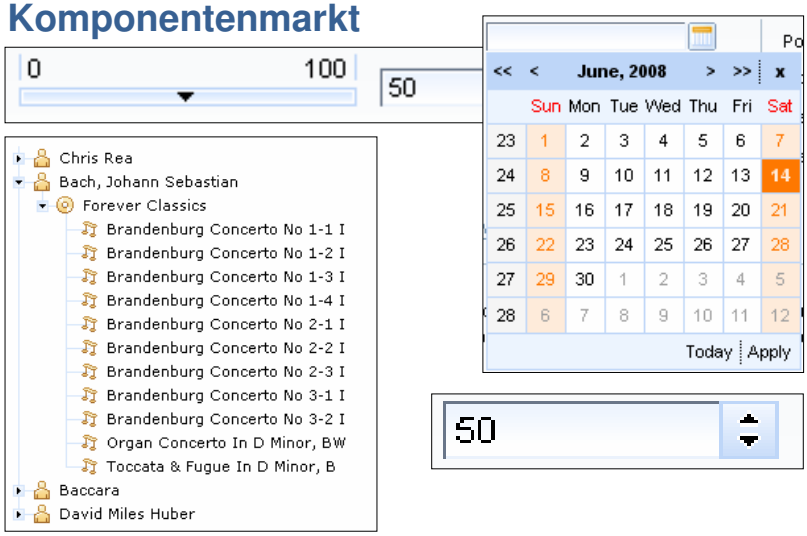
```
<h:form>
  <rich:calendar popup="true"/>

  <rich:inputNumberSpinner
          minValue="0"
          maxValue="100" step="1"/>
</h:form>
```


JSF und Ajax

- Ajax hat zunächst einmal nichts mit JSF zu tun, sondern ist eine eigenständige „Technologie“
- Es gibt jedoch Frameworks für JSF, die eine Integration von Ajax in JSF erleichern.
- Z.B. bei ajax4jsf muss der Anwendungsentwickler keinerlei JavaScript-Code mehr schreiben
- Viele Komponentenbibliotheken im JSF-Umfeld sind bereits „ajaxifiziert“ und nutzen somit die Vorteile von PPR und asynchroner Verarbeitung.

Komponentenmarkt



Fazit JSF und Ajax

- JavaServer Faces bieten einen großen Komponentenmarkt
- JSF liefert viele Konzepte für die UI-Entwicklung
- JSF hat sich für die UI-Entwicklung etabliert
- JSF ist auch ein Standard
- Ajax läßt sich sehr gut in JSF integrieren.
- Es gibt Tools und IDE‘s, um eine JSF-Entwicklung zu unterstützen

Agenda

- Überblick Portlets und Portale
  - Was sind Portale?
  - Von Portalen zu Portlets
  - Funktionsweise eines Portlets
- Überblick JavaServer Faces und Ajax
  - Was ist JSF? Was ist Ajax?
  - UI-Entwicklung mit JSF
- **Kombination von JSF und Portlets (JSR-301)**
  - Wo liegt die Herausforderung?
  - „Frühe“ Ansätze
  - Der JSR-301: Portlet Bridge Specification for JSF

Warum soll beides kombiniert werden?

Portlets (bzw. die Portlet-API) ist recht „basisnah“. Trotz des JSR-286 ist eine direkte Verwendung der API nicht gerade komfortabel.

Konzepte wie Pageflow, Konvertierung, Validierung, Eventhandling, UI-Komponenten fehlen komplett.

Es muss somit die Portlet-Technologie mit einem „reichhaltigen“ UI-Gerüst vermengt werden.

jsf-portlets.net

Die Lösung: JavaServer Faces

JSF kann all das! ☺

… wußten wir ja schon immer …

jsf-portlets.net

JSF + Portlets?

Das Problem jedoch ist der „Glue-Code“.

- Beide Technologien müssen integriert werden
- Jede Technologie soll ihre „ureigene“ Funktionalität beibehalten
- Integrationsaufwände sollen möglichst gering sein

Geht das überhaupt?

→ Ja, mit dem JSR-301

JSR-301: PortletBridge Specification for JavaServer Faces

- Start im Dezember 2006
- Spec Lead: Michael Freedman (Oracle)
- Ziel ist die Bereitstellung einer standardisierten Bridge, um JSF-Anwendungen im Portlet-Context ausführen zu können.
- Ziel ist es außerdem, die Handhabung für den Entwickler so einfach wie möglich zu gestalten.
- Innerhalb des 301 wird es zwei Spezifikationen geben: Eine für JSR-168 + JSF 1.2
  und eine für JSR-286 + JSF 1.2

Die Welt vor JSR-301

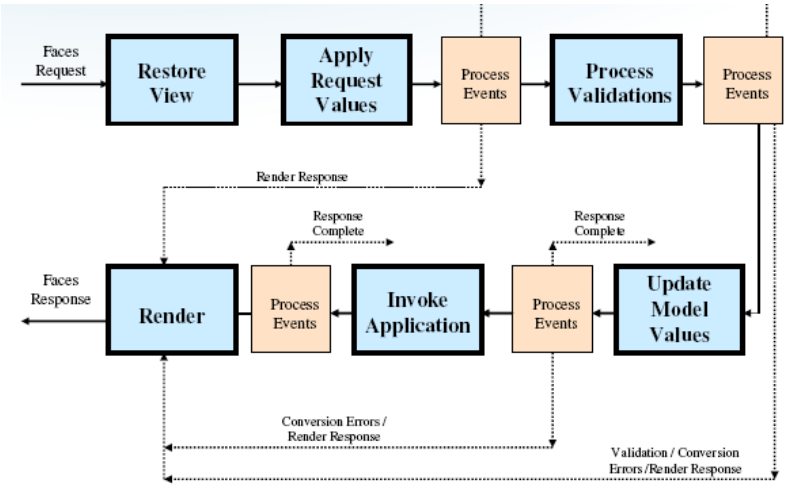
- Jeder Application Server hatte eine eigene Bridge-Implementierung
- Keine Bridge-Implementierung funktionierte wirklich ☹
- Komponentenbibliotheken waren nur mit viel „Fummel-Aufwand“ zum Laufen zu bekommen (wenn überhaupt)
- Es gab viele Projekte im Umfeld, um den fehlenden Standards Abhilfe zu schaffen (Filter-Portlets, Tomahawk-Bridge, …)
- Irgendwie machte dies keinen Spaß …

Aufgaben des JSR-301

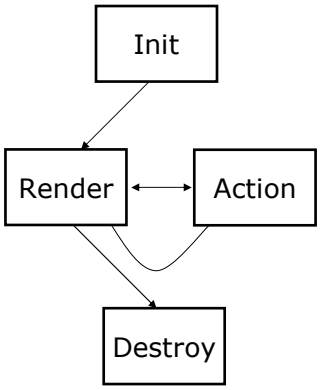
- **Standardisierung einer Bridge-Lösung, nach der sich möglichst alle Bridge-Implementierer richten können**

- Mapping der Lifecycles
- Einführung eines Bridge Request Scope
- URL-Adressierungen
- u.v.m.

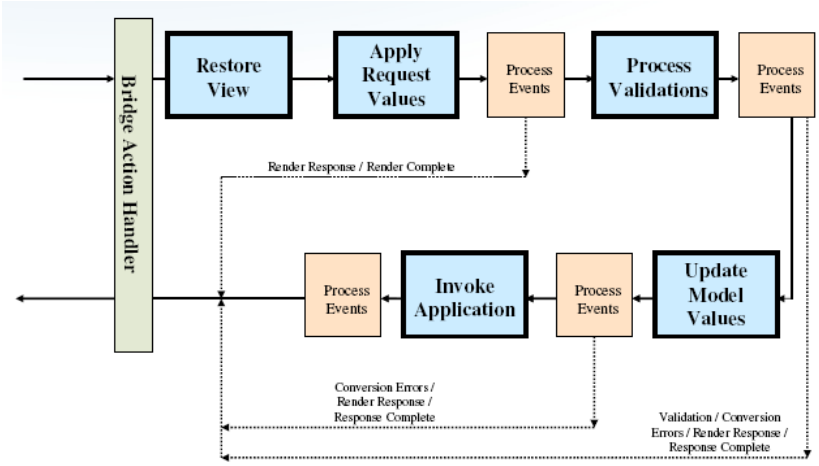
Mapping der Lifecycles: JSF



Zur Erinnerung: Portlet Lifecycle

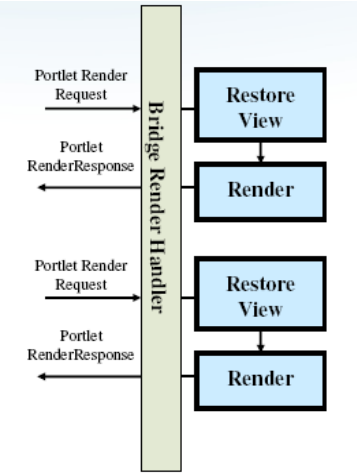


Mapping der Lifecycles: Bridge



jsf-portlets.net

Mapping der Lifecycles: Bridge



jsf-portlets.net

Bridge Request Scope

- JSF geht davon aus, dass der komplette Lebenszyklus in einem Request abgearbeitet wird.
- Bei Portlets wird die Render- und die Action-Phase in zwei komplett unabhängigen Requests durchgeführt

→ Die Bridge muss dafür sorgen, dass alle notwendigen Informationen für JSF in beiden Requests vorhanden sind.
→ Dazu werden die Attribute der Action-Phase durch die Bridge zwischengespeichert und für jede Render-Phase bereitgestellt.

JSR-301 Implementierungen

- Referenzimplementierung innerhalb der MyFaces-Community
  http://myfaces.apache.org/portlet-bridge/index.html
- Open Portal Implementierung
  https://jsfportletbridge.dev.java.net/
- JBoss Implementierung
  http://www.jboss.org/portletbridge/
- …

Einsatz der PortletBridge (1)

- Entwicklung einer „normalen“ JSF-Anwendung
- Entfernen „verbotener“ Tags wie <html>, <body>, <head> etc.
- Einbinden der JSR-301 Libs
- Einbinden der portlet.xml und Hinterlegen des GenericFacesPortlet
- Deploy and have fun ☺

Einsatz der PortletBridge (2)

```
<portlet>
   <portlet-name>HelloJSF</portlet-name>
   <portlet-class>
      javax.portlet.faces.GenericFacesPortlet
   </portlet-class>
   <init-param>
      <name>
        javax.portlet.faces.defaultViewId.view
      </name>
      <value>/helloJSF.jsp</value>
   </init-param>
   <supports>
      <mime-type>text/html</mime-type>
      <portlet-mode>VIEW</portlet-mode>
   </supports>
</portlet>
```

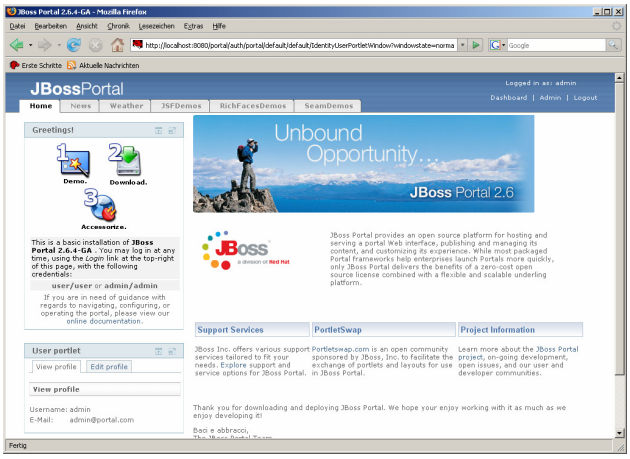

Beispiel JBoss Portal

JBoss hat eine eigene Implementierung des JSR-301.

Die Beispiele beruhen auf folgenden Versionen:

- JBoss Portal 2.6.4
- JBoss Application Server 4.2.2
- JBoss Portlet Bridge 1.0.0.B2
- RichFaces 3.1.4.SR1
- Seam 2.1.0.A1

Live Demo



WSRP?

- WSRP sind „Remote Portlets“, d.h. Portlets können aus einem anderen Portal in das eigene eingebunden werden.
- WSRP hat mit dem JSR-301 nichts zu tun, da es lediglich um die Einbindung eines Portlets geht.
- Folglich funktionieren auch Portlets über WSRP mit dem JSR-301

Zeitplan und Status des JSR-301

- Portlet 1.0 Bridge:
  - Aktuell: Public Review
  - RI verfügbar
- Portlet 2.0 Bridge:
  - Early Draft 1: Bald
  - RI für ED1 : Bald ☺

Fazit

- JavaServer Faces ist cool ☺
- Ajax ist auch cool (zumindest wenn man geeignete Frameworks dafür hat) ☺
- Portale und Portlets sind enorm mächtig ☺
- Durch Verwendung der JSR-301-Bridge hat man eine funktionierende und durchdachte Möglichkeit, Portlets und JSF zu kombinieren ☺
- Auch die Integration von UI-Bibliotheken funktioniert mit der 301-Bridge ☺
- Das Ergebnis ist durchaus sehenswert und bietet enorme Möglichkeiten !!! ☺ ☺ ☺

jsf-portlets.net

Achtung: Werbeblock

Erscheint (vermutlich ☺)
im Herbst 2008

jsf-portlets.net

Fragen?

Weitere Tutorials und
viel Material unter:

www.jsf-forum.de

oder auch:

www.jsf-portlets.net

Gerne auch:
andy.bosch@jsf-forum.de