

Spezifikation und Steuerung Ajax-basierter Nutzerinteraktion im Dialog Control Framework

Michael Becker

Professur für Angewandte Telematik / e-Business

09.12.08 / DuD Seminar

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

Demoanwendung

Fazit und Fragen

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

Demoanwendung

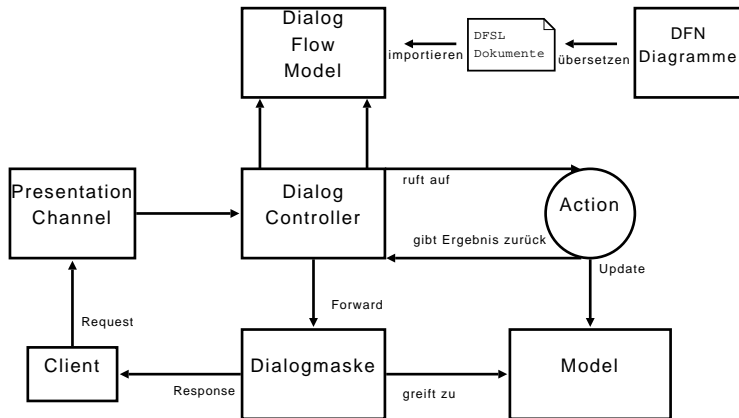
Fazit und Fragen

Evolution von Web-Anwendungen

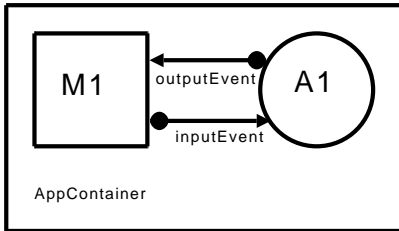
- ▶ Trend von statischen zu dynamischen Web-Anwendungen: e-Commerce, Online Banking, Auktionsplattformen
- ▶ Komplexität von Anwendungen managen: DCF, JSF
- ▶ Nutzer sind an interaktive, bedienungsfreundliche GUIs gewöhnt
- ▶ dynamisch auf Aktionen der Benutzer reagieren: Ajax

Dialog Control Framework

- ▶ liest DFSL ein
- ▶ übernimmt Dialogflusssteuerung in MVC-Anwendungen
- ▶ wählt anhand Eventnamen das nächste Dialogelement aus



DFN / DFSL



```
<in-appcon>
  <regular-elements>
    <mask name="M1">
      <file channel=html>M1.jsp</file>
    </mask>
    <action name="A1">
      <class>org.dialogcontrol.dcf.demo.A1
    </action>
    ..
    <ex-mask name="M1">
      <on-event name="inputEvent">
        <call-action>A1</call-action>
      </on-event>
    </ex-mask>
    <ex-action name="A1">
      <on-event name="outputEvent">
        <call-mask>M1</call-mask>
      </on-event>
    </ex-action>
```

...

Ziele

- ▶ Möglichkeiten der Verbesserung der Nutzerinteraktion durch die Verwendung von Ajax
- ▶ DCF-Integration mit JSF vorantreiben
- ▶ Erweiterung des DCF, die auf Ajax-Anfragen reagiert

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

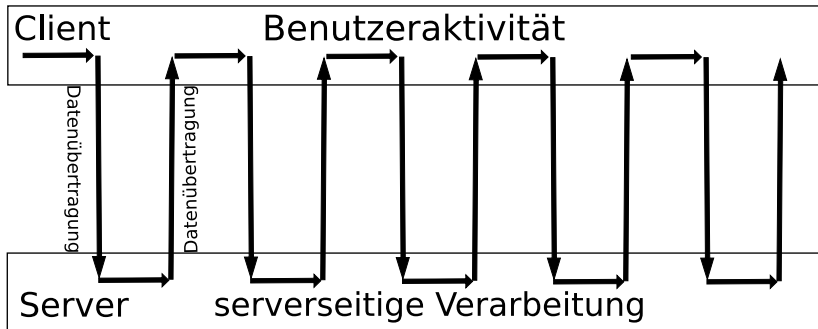
Demoanwendung

Fazit und Fragen

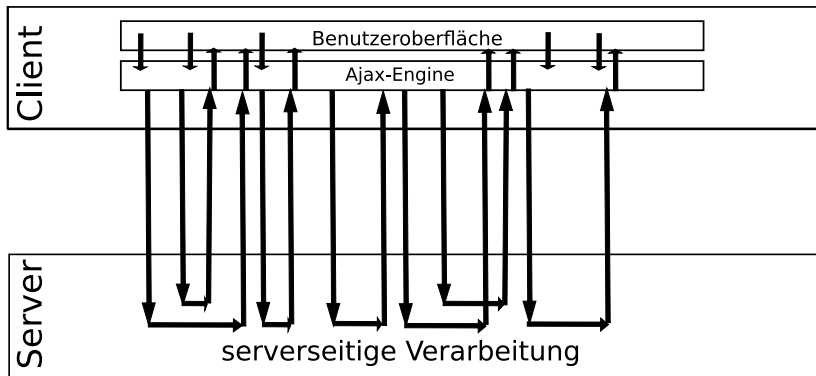
Web 2.0 und noch mehr Buzzwords

- ▶ Konglomerat standardisierter und bekannter Techniken
- ▶ Asynchronous Javascript and XML
- ▶ asynchrone HTTP-Requests
- ▶ Response wird durch Javascript / DOM verarbeitet
- ▶ Daten liegen (meist) im XML-Format vor
- ▶ Grundidee: Daten erst bei Bedarf laden, Benutzer soll während dem Laden weiterarbeiten
- ▶ Ajax ist aus technischer Sicht Teil von Web 2.0

Klassischer Request-Response-Zyklus



Ajax Request-Response-Zyklus



Vorteile von Ajax

- ▶ Standardtechnologien, keine Plugins, keine Vorgabe an Serverseite
- ▶ bessere Reaktionsmöglichkeiten auf Benutzeraktionen
- ▶ Entlastung Serverseite durch Verlagerung der Darstellungslogik auf den Client
- ▶ Verringerung des Traffics, da Daten statt Dokumente übertragen werden

Probleme bei der Verwendung von Ajax

- ▶ benötigt zwingend Javascript, gerade in Firmen nicht immer aktiviert
- ▶ Browserinkompatibilitäten
- ▶ Aushebeln der History-Funktion (ist das wirklich ein Nachteil?)
 - ▶ Gewöhnung der Benutzer an Seiten-Paradigma
- ▶ fehlendes Feedback - muss selbst implementiert werden
- ▶ barrierearme Websites

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

Demoanwendung

Fazit und Fragen

Zu lösende Fragen

- ▶ Kann das DCF so erweitert werden, dass Ajax-Anfragen gehandhabt werden können?
- ▶ Wie müssen die Spezifikationsmöglichkeiten DFN / DFSL erweitert werden?
- ▶ Ermöglicht die Verwendung eines Frameworks Zeitersparnisse?

DCF und JSF - Clash of Cultures?

- ▶ JSF: Entwicklung von Benutzeroberflächen für Web-Anwendungen
- ▶ weg von JSPs vermischt mit Java-Code hin zu einer MVC-Architektur
- ▶ wiederverwendbare Komponenten für die View
- ▶ Abarbeitung eines Request mittels Lebenszyklus, der Aufgaben wie Validierung, Konvertierung, . . . auf einzelne Phasen verteilt
- ▶ Navigation mittels faces-config
- ▶ DCF: Dialogflusssteuerung

DCF und JSF - So passt's

- ▶ DCF agiert als JSF-Navigationhandler
- ▶ Event-Name: action-Wert von `commandButton` / -Links
- ▶ durch DCF weitaus mächtigere Navigationssteuerung als über `faces-config` möglich

Ajax im JSF-Lebenszyklus

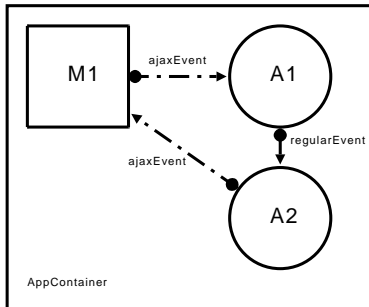
- ▶ momentan keine generische Unterstützung, geplant mit JSF 2.0
- ▶ bei jedem Request komplette Abarbeitung des Lifecycles
- ▶ durch Ajax: viel mehr Request, erhöhte Belastung des Servers
- ▶ durch erweiterbare Architektur Komponenten mit Ajax-Unterstützung

Ajax im JSF-Lebenszyklus, II

- ▶ unterschiedliche Integrationsmöglichkeiten
- ▶ am praktikabelsten JSF PhaseListener: direkt im JSF Lifecycle integriert, Zugriff auf Komponentenbaum
- ▶ Response per Partial Page Rendering oder Delta DOM Rendering

Erweiterung der DFN / DFSL

- ▶ DFN: neue Kante für Ajax-Events
- ▶ Eigenschaften wie Kante für reguläre Events, Strichpunktspfeil



- ▶ DFSL: neues Element on-ajax
- ▶ am Ende der Verarbeitungskette on-ajax an die Maske
- ▶ Anstoß clientseitiger Aktualisierungen, kein Neuladen der Maske

Regeln für Ajax-Events

- ▶ Ajax-Event als Reaktion auf Benutzereingabe: auslösendes Element muss eine Maske sein
- ▶ am Ende muss wieder die auslösende Maske angezeigt werden, alles andere wäre ein Seitenwechsel
- ▶ innerhalb der Abarbeitungskette dürfen keine anderen Masken erreichbar sein
- ▶ ansonsten: wie reguläre Events, also auch Einschachtelung von Modulen / Containern möglich

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

Demoanwendung

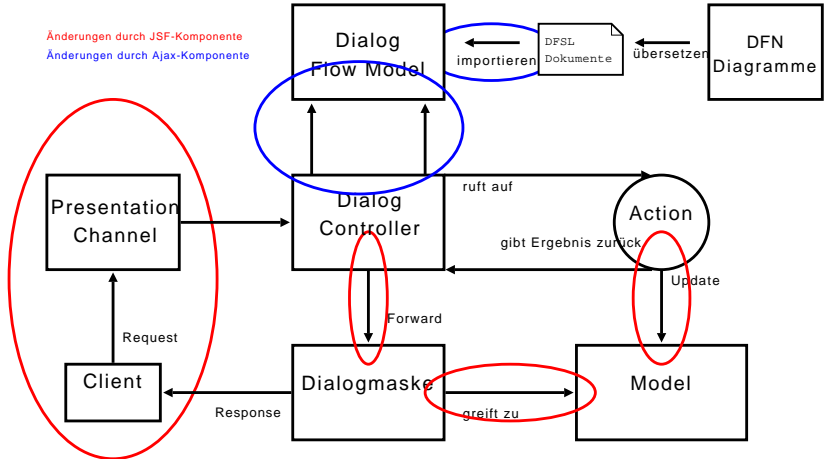
Fazit und Fragen

Verhalten des DCF bei Ajax-Anfragen

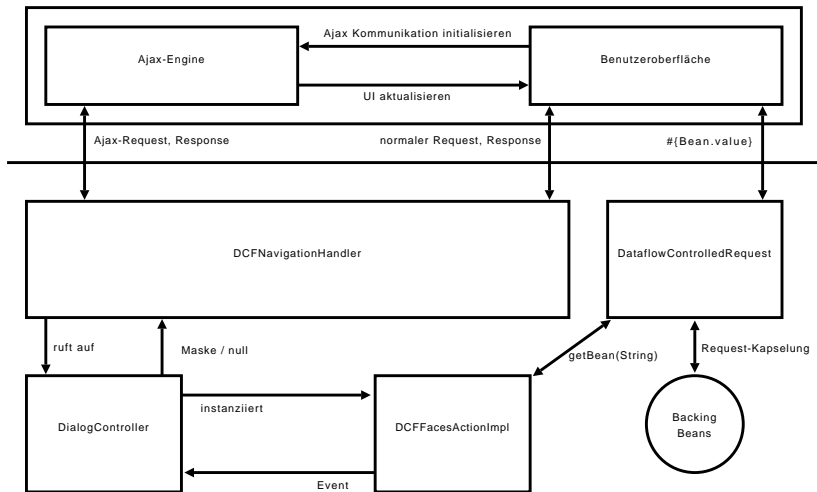
- ▶ Benutzer schickt Request ab, DCF erkennt Ajax-Anfrage anhand eingelesener Spezifikation
- ▶ vorhandene Events können leicht in Ajax-Events umgewandelt werden, wodurch die Erweiterung in bestehende Anwendung erleichtert wird; es muss nur die DFSL angepasst werden
- ▶ Abarbeitung des Events wie bisher
- ▶ das letzte Event in der Eventkette ist wieder ein Ajax-Event, wodurch keine neue Maske geladen wird sondern die alte Maske bestehen bleibt

Änderungen an der DCF Architektur

Änderungen durch JSF-Komponente
Änderungen durch Ajax-Komponente



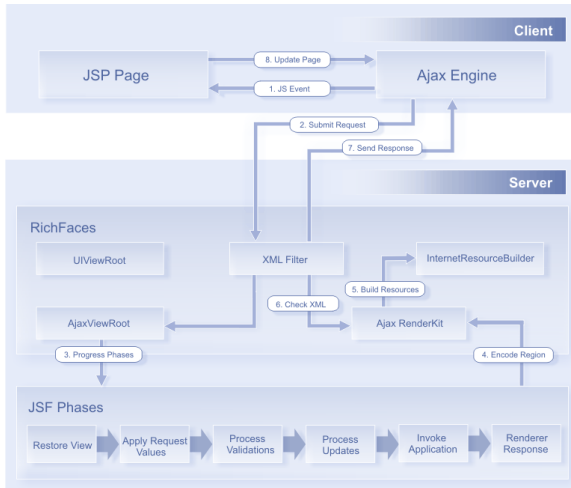
DCF, JSF, Ajax



- Integration eines JSF-Ajax-Frameworks

- ▶ Erweiterung bestehender JSF-Komponenten um Ajax-Funktionalität
- ▶ RichFaces als Komponentenbibliothek
- ▶ Ajax4JSF zur Erweiterung bestehender Komponenten, ermöglicht seitenbasierte Ajax-Integration
- ▶ problemlose Konfiguration
- ▶ Corporate Design durch verschiedene Skins

RichFaces Architektur



- Schnittstelle zum DCF

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

Demoanwendung

Fazit und Fragen

Demoanwendung

- ▶ Benutzerregistrierung der vorhandenen Demoanwendung mittels Ajax

Inhalt

Einleitung

Evolution von Web-Anwendungen

Vorstellung DCF

Ziele der Arbeit

Ajax

Vorstellung

Vorteile / Probleme

Integrationsmöglichkeiten Ajax - JSF - DCF

DCF und JSF

JSF und Ajax

Erweiterung der DFN / DFSL

Implementierungsdetails

Verhalten des DCF bei Ajax-Anfragen

Anpassungen am DCF

Demoanwendung

Fazit und Fragen

Fazit

- ▶ Integration mit JSF: komplexere UIs auf Komponentenbasis
- ▶ Erweiterung des DCF um Ajax möglich
- ▶ viele neue Interaktionsmöglichkeiten: Zwischenspeichern, Validieren beim Fokuswechsel, schnellere Fehlermeldungen, ...

Was noch zu tun bleibt, Ausblick

- ▶ Implementierung säubern, Fehler entfernen
- ▶ Ausgefeilterte UIs durch clientseitige Javascriptbibliotheken
- ▶ Integration on-compound-ajax, on-common-ajax
- ▶ mittels Cepheus automatisch Ajax-fähige Dialogflüsse ermitteln
- ▶ periodische Events in DFN / DFSL

Fragen, Anregungen, Diskussion

- ▶ Danke für die Aufmerksamkeit.

alle Jahre wieder
Weihnachtet Es

und die Infos und Kugeln Feiern
18. Dezember 2008
2 Bands 3 DJ's

21 Uhr
im Club Lagerhof

Welcomedrinks so lange der Vorrat reicht

Unkostenbeitrag 2€
in echter Kostüemierung 1€

Club Lagerhof
Lagerhofstr.2
04103 Leipzig

Hauptbahnhof Leipzig

Willy-Brandt-Platz

B6/B67
Richtung Torgau

Tram 1, 3, 8, 13
N6
"Friedrich List Platz"

de Gasse jooanz weit hinter laufen

Bands:

1. Band: "The Legend so Far"
2. Band: "Kiloherz"

DJ's:

- DJ Madtrickx (Dancehall, Reggae)
- DJ Warpcut (Funk, Oldschool, House)
- DJ Inertial (Drum n Bass)

UNIVERSITÄT LEIPZIG

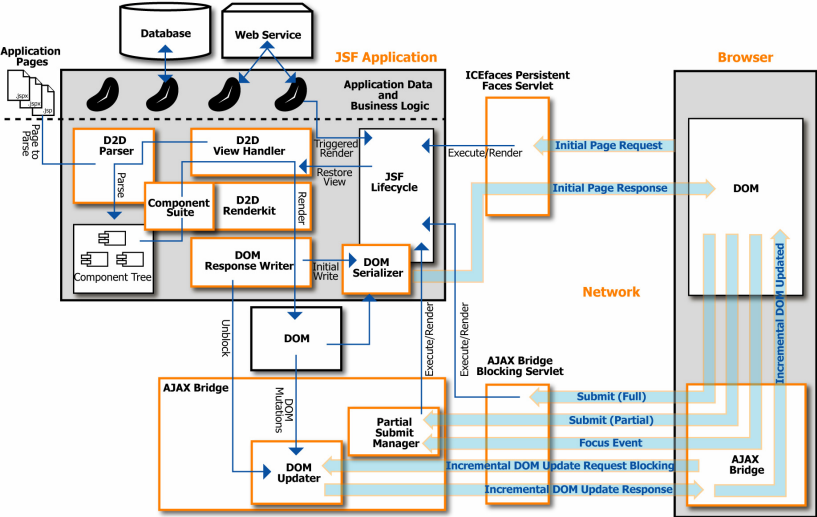
LEIPZIGER UNIVERSITÄT

KUGEL

IMMER HOCH VERWANDT?
Folgt einfach dem Beschilderungen/teilen oder Richtungweisern

- ▶ JSF basierendes Ajax-Framework
- ▶ Standard JSF Renderer wird durch Direct to DOM Renderer ersetzt
 - ▶ Serverseitiger DOM-Baum, Änderungen mittels Delta DOM
 - ▶ rendert JSF-Komponentenbaum direkt in DOM Struktur
- ▶ Persistent Faces Servlet: Initiales Rendern von Seiten, vor den Lifecycle geschaltet; Initialisierung Renderer, Weiterleitung Request
- ▶ Blocking Servlet: Nimmt alle weiteren Requests entgegen

ICEFaces Architektur



- ▶ RichFaces: durch Ajax4JSF sehr leichte Integration in vorhandene Anwendungen
- ▶ problemlose Konfiguration
- ▶ Corporate Design durch verschiedene Skins

- ▶ ICEFaces Serverseitige Aktualisierungen: Reverse Ajax weit fortgeschritten
- ▶ etwas komplexere Konfiguration durch Persistent Faces Servlet und Blocking Servlet
- ▶ JSPs müssen in validem XML-Format vorliegen, damit D2D-Renderer Delta-DOM generieren kann

Auswahl eines Frameworks

- ▶ zur Zeit RichFaces aufgrund der leichteren Integrierbarkeit mit MyFaces
- ▶ ICEFaces hat komplexere Architektur durch eigenen ViewHandler, Renderkit, FrontController; daher unter Umständen schwierigere Anbindung an bestehende Anwendungen
- ▶ Erweiterung soll möglichst generisch werden, so dass Framework austauschbar