

Patrick Ruhkopf
Wilhelm Nüßer (Hrsg.)
Stefan Nieland (Hrsg.)

**Software Reuse zur Beschleunigung der
Entwicklung Java EE basierter
Webanwendungen –
Ansätze & Potentiale**

FHDW Ostwestfalen
Fachbericht Nr. 1/2008

FHDW-Fachbericht

Band 1/2008

Patrick Ruhkopf
Wilhelm Nüßer (Hrsg.)
Stefan Nieland (Hrsg.)

**Software Reuse zur Beschleunigung der
Entwicklung Java EE basierter Webanwendungen –
Ansätze & Potentiale**

Shaker Verlag
Aachen 2008

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Copyright Shaker Verlag 2008

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8322-7621-8

ISSN 1861-3292

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • E-Mail: info@shaker.de

Vorwort

Die Wiederverwendung von Software hat eine lange Tradition in der Informatik. Sie ist aus den unterschiedlichsten Gründen immer wieder in den Fokus der Diskussion geraten. Waren es zu Beginn einfache modularisierte Systeme, die eine Wiederverwendung nahelegten, so stand bei der Entwicklung Objektorientierter Ansätze diese Frage schon im Zentrum des Designs. Die Bestrebungen nach komponentenorientierter Software-Entwicklung, die in den 80er und 90er Jahren des letzten Jahrhunderts ansetzten, gingen hier sogar noch einen Schritt weiter. Eine Komponente konnte geradezu durch ihre Fähigkeit zur Wiederverwendung definiert werden.

Komponentenbasierte Ansätze haben Technologien bzw. Produktlinien wie JEE und .NET beflügelt und haben sich damit faktisch in der heutigen IT-Landschaft etabliert. Auch die heute immer noch intensiv untersuchten serviceorientierten Architekturen schreiben sich Wiederverwendung auf die Fahnen. Ist damit eine Publikation wie die vorliegende noch notwendig?

Wir sind davon überzeugt, da Wiederverwendung bei allen Erfolgen doch immer noch einige grundlegende Fragen offen lässt. Dazu zählt zum einen die Frage nach der faktischen Wiederverwendung von einmal geschriebener Software. Wie so oft, scheint hier eine Kluft zwischen konzeptioneller Vision und praktischer Umsetzung zu existieren. So sind z.B. die Java Beans einer JEE-Anwendung als Komponenten zu verstehen und damit für die Wiederverwendung geschaffen. In der Praxis findet sich eine erneute Verwendung einmal geschriebener Beans aber nur in besonderen Fällen. Dazu gehören z.B. Unternehmen, die immer wieder ähnliche Software erstellen müssen und deshalb passende Frameworks entwickelt haben. Die Vision einer Software-Entwicklung, die Komponenten Dritter einfach zusammensteckt, hat sich weder im Bereich klassischer komponentenbasierter Software noch bei SOA bislang erfüllt. In den weitaus meisten Fällen findet immer noch sehr viel Neuentwicklung statt.

Eine zweite Frage betrifft die Art der Software, die wiederverwendet werden soll. Eine Wiederverwendung von serverseitigen oder Backend-Komponenten hat eine deutlich höhere Verbreitung gefunden als die Wiederverwendung von Frontend-Komponenten. Ausnahmen sind hier etablierte Tag-Libraries etc. Grundsätzlich stellt sich aber häufig die Frage, in welcher Weise Wiederverwendung bei der Dialog-Gestaltung selbst zum Einsatz kommen kann. Welche Elemente eines GUIs sind wiederverwendbar und in welcher Weise können diese einfach zu vollständigen Dialogen zusammengesetzt werden?

Diese zweite Frage steht im Fokus der vorliegenden Ausarbeitung. Ausgehend von aktuellen Technologien, wie JEE, JSF, Seam, zeigt die Arbeit, wie Software-Reuse bei der Entwicklung von Web-Anwendungen sinnvoll eingesetzt werden kann. Eine gründliche Analyse moderner Dialoge ist dazu notwendig. Die Ergebnisse dieser Analyse lassen hoffen, die Entwicklung moderner Web-Anwendungen mittels wiederverwendeter Software zu beschleunigen. Wenn dies der Fall ist, hat die Arbeit ihren Zweck erfüllt.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	V
Abkürzungsverzeichnis	VI
1 Einführung.....	1
1.1 Einleitung und Motivation	1
1.2 Zielsetzung & Vorgehensweise	2
2 Software Reuse.....	4
2.1 Ursprung & Begriffsbestimmung.....	4
2.2 Aspekte der Softwarewiederverwendung	7
2.3 Ziele, Nutzen und Hindernisse.....	11
2.4 Softwaretechnische Maßnahmen	14
3 Java EE Webanwendungen	16
3.1 Webanwendungen.....	16
3.1.1 Definition.....	16
3.1.2 Kategorien von Webanwendungen.....	18
3.1.3 Besondere Charakteristika von Webanwendungen	19
3.1.4 Die Benutzerschnittstelle	22
3.2 Java Platform, Enterprise Edition	24
3.2.1 Java EE Technologien	25
3.2.2 JavaServer Faces.....	27
3.2.3 Dritthersteller Frameworks.....	35
4 Analyse.....	42
4.1 Ist-Situation	42
4.1.1 Software Engineering	42
4.1.2 Web Engineering	43

4.1.3	Cliententwicklung.....	46
4.1.4	Die Benutzerschnittstelle.....	48
4.2	Sollkonzept.....	59
4.2.1	Allgemeine Anforderungen.....	59
4.2.2	Anforderungen an den Dialograhmen.....	59
4.2.3	Anforderungen an Dialoge.....	61
4.3	Evaluation.....	63
4.3.1	Dialogflusssteuerung.....	63
4.3.2	Kommunikation zwischen Dialogen.....	64
4.3.3	Daten-, Aktions- und Zustandsverwaltung.....	66
4.3.4	JBoss Seam.....	68
4.4	Ergebnis.....	69
5	Entwurf.....	71
5.1	Szenario-Übersicht.....	71
5.1.1	Programmabläufe.....	72
5.2	Architektur.....	73
5.2.1	Geschäfts- und Datenschicht.....	74
5.2.2	Webclient.....	77
6	Implementierung.....	84
6.1	Der Bestellvorgang.....	85
6.2	Hindernisse.....	89
7	Ergebnisse und Ausblick.....	93
7.1	Ergebnisse.....	93
7.2	Ausblick.....	94
	Anhangsverzeichnis.....	i
	Quellenverzeichnis.....	xvii

Abbildungsverzeichnis

Abbildung 1 Software Reuse Ansätze	7
Abbildung 2 Besondere Charakteristika von Webanwendungen.....	20
Abbildung 3 Arbeitsweise einer Template Engine	23
Abbildung 4 JSP Model 2 Architektur.....	24
Abbildung 5 Java Multitier Application	25
Abbildung 6 JavaServer Faces Model 2-Architektur.....	28
Abbildung 7 Verwaltung von JSF-Widgets in einem Komponentenbaum.....	29
Abbildung 8 JavaServer Faces Request-Response Life Cycle	31
Abbildung 9 Quelltextausschnitt, Deklaration einer Backing Bean	32
Abbildung 10 Quelltextausschnitt, Deklaration von JSF-Komponenten (Template).....	33
Abbildung 11 Quelltextausschnitt, JSF-Konfiguration von Navigationsregeln	34
Abbildung 12 Quelltextausschnitt, Statische Navigation in JSF	34
Abbildung 13 Quelltextausschnitt, Dynamische Navigation mit JSF.....	35
Abbildung 14 Quelltextausschnitt, Seam Bijection	39
Abbildung 15 Quelltextausschnitt, Definition einer Seite in jPDL.....	40
Abbildung 16: Typische Aufgaben eines Clients.....	47
Abbildung 17 Darstellung eines einfachen datenorientierten Dialogflusses	50
Abbildung 18 Quelltextausschnitt, Transparentes Lesen von Daten aus der Session mittels JavaServer Faces' Dependency Injection	52
Abbildung 19 Quelltextausschnitt, Nicht-transparentes Schreiben von Daten in die Session mittels JavaServer Faces Framework-Klassen	53
Abbildung 20 Quelltextausschnitt, Injektion einer Eigenschaft einer Backing Bean in die Eigenschaft einer anderen Backing Bean	53
Abbildung 21 Seite mit hierarchischen Dialogstrukturen.....	54
Abbildung 22 Dialog als Container für steuerbare Dialoge.....	55
Abbildung 23 Quelltextausschnitt, Präsentationssteuerung in JSF.....	57
Abbildung 24 Bestandteile des Dialograhmens in der Übersicht	60

Abbildung 25 Aufgabenverteilung in Dialogen	62
Abbildung 26 Quelltextausschnitt aus einem Template, Benutzerdefiniertes Tag mit Anbindung an einen Dialog	66
Abbildung 27 Abgeänderte Aufgabenverteilung in Dialogen	67
Abbildung 28 Use Cases des Beispielszenarios.....	71
Abbildung 29 Programmablauf, Produktauswahl	72
Abbildung 30 Programmablauf, Bestellvorgang.....	73
Abbildung 31 Verteilungsdiagramm des Beispielszenarios	73
Abbildung 32 Klassendiagramm, Geschäftsklassen für die Suchfunktion	74
Abbildung 33 Klassendiagramm, Bestellung.....	76
Abbildung 34 Klassendiagramm, Präsentationszustände	77
Abbildung 35 Klassendiagramm, Dialoge	78
Abbildung 36 Klassendiagramm, Suchdialoge	80
Abbildung 37 Klassendiagramm, DeliveryOptionsDialog	80
Abbildung 38 Klassendiagramm, Dialoge des Use Case „Produkte auswählen“	81
Abbildung 39 Klassendiagramm, Dialoge des Use Case „Produkte bestellen“	82
Abbildung 40 Dialogfluss Programmablaufs „Produktauswahl“	83
Abbildung 41 Dialogfluss des Programmablaufs „Produktauswahl“	84
Abbildung 42 Quelltextausschnitt, Beginn und Ende des Bestellvorgangs.....	86
Abbildung 43 Quelltextausschnitt, jPDL „checkoutFlow“, Teil 1	87
Abbildung 44 Quelltextausschnitt, jPDL „checkoutFlow“, Teil 2	88
Abbildung 45 Quelltextausschnitt, jPDL „checkoutFlow“, Teil 3	89
Abbildung 46 Quelltextausschnitt, Validierung eines Geschenkgutscheinodes	90

Tabellenverzeichnis

Tabelle 1 Aspekte der Softwarewiederverwendung	8
Tabelle 2 Software Reuse basierend auf Komponenten und Generierung im Vergleich.....	9
Tabelle 3 Vorteile von Software Reuse	12
Tabelle 4 Kategorien von Webanwendungen basierend auf Funktionalität	18
Tabelle 5 Kategorien von Webanwendungen basierend auf Entwicklungshistorie und Architekturmerkmalen	19
Tabelle 6 Java EE Technologien der Webschicht.....	27
Tabelle 7 Phasen des JavaServer Faces Request-Response Life Cycle	30
Tabelle 8 Seam Kontexte	37

Abkürzungsverzeichnis

CSS	Cascading Style Sheets
EJB	Enterprise JavaBeans
HTML	HyperText Markup Language
JAX-WS	Java API for XML Web Services
JPQL	Java Persistence Query Language
MDBs	Message-driven Beans
MVC	Model-View-Controller
RIA	Rich Internet Application
SFSB	Stateful Session Bean
SLSB	Stateless Session Bean
UI	User Interface
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language