

Sascha Vogel

Eine Methode zur Entwicklung  
flexibler Dienste auf der Basis  
von Interaktionsmustern



Herbert Utz Verlag · München

Bibliografische Information Der Deutschen Bibliothek  
Die Deutsche Bibliothek verzeichnet diese Publikation  
in der Deutschen Nationalbibliografie; detaillierte  
bibliografische Daten sind im Internet über  
<http://dnb.ddb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt. Die  
dadurch begründeten Rechte, insbesondere die der  
Übersetzung, des Nachdrucks, der Entnahme von Ab-  
bildungen, der Wiedergabe auf photomechanischem  
oder ähnlichem Wege und der Speicherung in Daten-  
verarbeitungsanlagen bleiben – auch bei nur auszugs-  
weiser Verwendung – vorbehalten.

Copyright © Herbert Utz Verlag GmbH · 2004

ISBN 3-8316-0334-0

Printed in Germany

Herbert Utz Verlag GmbH, München

089-277791-00

[www.utzverlag.de](http://www.utzverlag.de)

# Inhaltsverzeichnis

<b>I</b>	<b>Einführung und Grundlagen</b>	<b>1</b>
<b>1</b>	<b>Motivation</b>	<b>3</b>
1.1	Flexibilitäts-Eigenschaften . . . . .	5
1.2	Charakteristik einer Methode . . . . .	7
1.3	Aufgabenstellung . . . . .	7
1.4	Aufbau und Ergebnisse der Arbeit . . . . .	9
<b>II</b>	<b>Eingrenzung des Themenfeldes und verwandte Arbeiten</b>	<b>13</b>
<b>2</b>	<b>Abgrenzung unterschiedlicher Dienst- und dienstähnlicher Konzepte</b>	<b>15</b>
2.1	Perspektiven auf Dienste . . . . .	15
2.2	Modellkonzepte . . . . .	18
2.3	Konzeptbewertung und Einordnung . . . . .	42
2.3.1	Gemeinsamkeiten . . . . .	43
2.3.2	Unterschiede . . . . .	43
2.3.3	Folgerungen . . . . .	43
<b>3</b>	<b>Lebenszyklus von Diensten</b>	<b>45</b>
3.1	Dienstrollen . . . . .	45
3.2	Kategorisierung von Entwicklungstypen . . . . .	46
3.3	Phasen des Lebenszyklus . . . . .	48
3.4	Betrieb von Diensten . . . . .	50
3.4.1	Negotiation . . . . .	50
3.4.2	Provisioning . . . . .	54
<b>III</b>	<b>Modell, Konstruktion und Beschreibung</b>	<b>59</b>
<b>4</b>	<b>Realisierung komplexer Interaktions- und Kommunikationsstrukturen</b>	<b>61</b>
4.1	Interaktionsmuster . . . . .	61
4.2	Charakteristische Kommunikationsstrukturen . . . . .	64

4.3	Interpretation von Transitionen . . . . .	65
4.3.1	Syntax . . . . .	65
4.3.2	Ausführungen . . . . .	69
4.3.3	Prozesse . . . . .	71
4.3.4	Verhalten von Mustern . . . . .	73
4.4	Ähnlichkeit und Äquivalenz . . . . .	73
4.5	Konstruktionsoperatoren . . . . .	75
4.6	Geflechte von Interaktionsmustern . . . . .	77
4.6.1	Komposition . . . . .	77
4.6.2	Sequenzialisierung . . . . .	86
4.6.3	Interaktionsgeflechte . . . . .	94
<b>5</b>	<b>Dienstmodell</b>	<b>97</b>
5.1	Atomare Interaktionen . . . . .	98
5.2	Höherwertige Interaktionen . . . . .	98
5.3	Kontext und Kommunikationsmechanismen . . . . .	99
5.4	Aktivierungen bzw. Navigationen . . . . .	100
5.5	Aktivierungsstrukturen . . . . .	101
5.6	Dynamik von Interaktionsmustern . . . . .	103
5.6.1	Erzeugung von Nachfolgern . . . . .	104
5.6.2	Aufrufbeziehungen . . . . .	105
5.7	Dienstdefinition . . . . .	106
5.8	Netzbasierende Dienste . . . . .	109
<b>6</b>	<b>Aspektbezogene Beschreibung von Diensten</b>	<b>111</b>
6.1	Wohlgeformte Spezifikationen . . . . .	113
6.2	Aspekt – Struktur und Dynamik . . . . .	114
6.3	Aspekt – Verhalten . . . . .	117
6.4	Aspekt – Kooperation . . . . .	122
6.5	Aspekt – Integration . . . . .	126
<b>7</b>	<b>Partitionierung</b>	<b>129</b>
7.1	Bewertung der Dienstkommunikation . . . . .	130
7.2	Bewertung der Ausführungs-Performance . . . . .	132
7.3	Verteilung von Interaktionsgeflechtem . . . . .	133
7.3.1	Nebenbedingungen . . . . .	133
7.3.2	Algorithmus . . . . .	135
7.3.3	Verteilungsalternativen . . . . .	137
<b>IV</b>	<b>Prototyp und Entwicklungsprozess</b>	<b>141</b>
<b>8</b>	<b>Ausführungsmodelle</b>	<b>143</b>
8.1	Mischen von Nachrichten . . . . .	146
8.2	Modellierung von Zeit . . . . .	149

8.3	Bewertung der Ausführungsmodelle . . . . .	152
8.4	Port-Integration und -Unifikation . . . . .	153
8.5	Skalierbarkeit von Agenten . . . . .	154
8.6	Integration der Konzepte . . . . .	155
<b>9</b>	<b>Prototyp</b>	<b>157</b>
9.1	Architektur des Prototyps . . . . .	158
9.2	Realisierungsplattform . . . . .	158
9.3	Agenten . . . . .	160
9.4	Werkzeug . . . . .	162
9.5	Vorgehensempfehlung . . . . .	162
9.6	Fallstudie - IGS-Dienst . . . . .	165
9.7	Werkzeug-bezogene Integration der Vorgehensaktivitäten . . . . .	166
9.7.1	Definition der Trägermenge von Interaktionsmustern . . . . .	166
9.7.2	Spezifikation der Aktivierungsstruktur . . . . .	168
9.7.3	Spezifikation von Schablonen . . . . .	170
9.7.4	Spezifikation der Muster . . . . .	173
9.7.5	Abhängigkeiten zwischen globalen Zuständen . . . . .	176
9.8	Integration der aspektorientierten Notationen . . . . .	177
9.9	Deployment und Simulation . . . . .	180
<b>V</b>	<b>Resümee</b>	<b>183</b>
<b>10</b>	<b>Zusammenfassung und Diskussion</b>	<b>185</b>
<b>11</b>	<b>Ausblick</b>	<b>189</b>

# Kapitel 1

## Motivation

In jüngster Zeit konnte im Bereich der Informations- und Kommunikationstechnik zunehmend der Trend zur Dienstorientierung beobachtet werden. Softwareanwendungen werden nicht mehr überwiegend in einem Unternehmen selbst erstellt bzw. betrieben, sondern immer öfter als Dienstleistung von anderen Softwareanbietern eingekauft oder abonniert (vgl. [BGHV00]). Dies wird oft unter dem Begriff des Application Service Provisioning beschrieben. Ein Dienst realisiert definierte Prozesse, die über Interaktionen zwischen verschiedenen Software-Bausteinen, aus denen er sich zusammensetzt, entstehen. Voraussetzung für diesen Trend ist ein hohes Maß an Flexibilität der Dienste. Nur so können sie in sehr heterogenen Umgebungen eingesetzt werden.

Der Dienstbegriff in der Informatik ist vielfach durch Client-Server-Architekturen charakterisiert. Dabei dient der Client als Dienstanutzer und der Server als Dienstanbieter. Dieses Konzept kommt heute in vielen Anwendungen zum Einsatz. Ein Beispiel ist die Nutzung von Internet-Diensten, bei denen die Anfragen über das Internet von einem Browser in der Funktion des Client an einen Webserver gestellt werden. Die Anfragen werden schließlich vom Server verarbeitet und entsprechende Antworten werden an den Browser zurückgegeben. Die Dienstlogik wird dabei zu großen Teilen im Server realisiert.

Dieses Konzept ist allerdings für moderne Anwendungen zu unflexibel, bei denen eine Vielzahl von Diensten (auch unterschiedlicher Anbieter) verwaltet werden müssen. Eine Lösung für dieses Problem ist die Integration von Vermittlungskomponenten wie beispielsweise Tradern. Diese ermöglichen eine Verwaltung von Schnittstellen. Dienstanbieter können ihre Schnittstellen an einen Trader exportieren und Dienstanutzer können nach dieser Schnittstelle suchen bzw. fragen. Dieses Konzept wird auch als Importer/Exporter/Trader-Modell beschrieben.

Aber auch dieses Modell weist einige Schwächen auf. Stets gibt es ein Ungleichgewicht zwischen dem Dienstanutzer, der lediglich Anfragen stellt und Antworten erhält und dem Dienstbringer, der nahezu alleine die komplexe Dienstlogik realisiert. Dienste, die kooperative und verteilt erbracht werden, können nur mit hohem Aufwand implementiert werden. Dieses Problem kann über Agen-

tensysteme gelöst werden, indem die Kommunikation nicht mehr zwischen einem Dienstbringer und einem Dienstnutzer erfolgt, sondern zwischen gleichberechtigten Partnern, die in diesem Zusammenhang mit Agenten bezeichnet werden. Ein Agent kann nicht nur Anfragen stellen, sondern auch Anfragen anderer beantworten. Somit entsteht ein System interagierender Agenten. Über dieses Konzept können Dienste entwickelt werden, die unabhängig voneinander und eigenständig, d.h. autonom, ausführbar sind. Diese Autonomie erlaubt eine flexible Nutzung der Agenten in unterschiedlichen Umgebungen, da sie sich selbständig zurechtfinden können und keinen assoziierten Server benötigen.

Zudem kann darauf aufbauend über geeignete Kompositionsmechanismen eine kooperative Erbringung von Funktionen der jeweiligen Agenten realisiert werden. Trotz dieser Autonomie ist es dennoch erforderlich, die unterschiedlichen Agenten bei der Funktionserbringung in einen Prozess einzubinden. Nur so ist es möglich, Dienste zu entwickeln, welche über die Interationen mit dem Dienstnutzer bzw. anderen Akteuren des Dienstes, wie Sensoren, Aktoren oder Steuergeräte, in ihrer Funktionserbringung gesteuert werden können. Einige Szenarien, in denen die Flexibilität als Folge der heterogenen Umgebung sowie die Interaktionsintensität essentiell und offensichtlich ist, werden im Folgenden kurz aufgezeigt:

- **Szenario Integrierte Gebäudesysteme:** Ein Szenario, welches die Notwendigkeit von Flexibilität aufzeigt, ist im Bereich der integrierten Gebäudesysteme denkbar. Vorausgesetzt wird hierbei ein Bürogebäude, welches mit elektronischen Geräten wie Sensoren (z.B. Windsensor, Temperaturfühler, Luftfeuchtigkeitsmesser, etc.) und Aktoren (elektronisches Heizungsventil, elektronische Jalousiesteuerung, etc.) ausgestattet ist. Die Geräte können über spezifische Dienste genutzt, gesteuert und konfiguriert werden. Während die Sensoren die Erfassung von Umweltdaten, wie die Zimmertemperatur, die Helligkeit oder die Windgeschwindigkeit ermöglichen, erlauben die Aktoren die Beeinflussung des Raumklimas, beispielsweise die Regelung der Raumtemperatur durch die Steuerung des Heizungsventils oder die Regelung der Helligkeit über die elektronische Jalousiesteuerung.

Basierend auf diesen Geräten und den dazu gehörigen Diensten können Mehrwertdienste entwickelt werden, die den Komfort für die Nutzer eines Gebäudes erhöhen, den Energieverbrauch optimieren oder das Leben in einem Gebäude vereinfachen. Letzteres wird beispielsweise erreicht, indem Dienstanbieter, wie Energieversorger, selbständig über entsprechende Sensoren den Energieverbrauch auslesen und aufgeschlüsselt nach Nutzerparteien abrechnen können. Der Betreiber eines Gebäudes würde bei einem solchen Szenario ausschließlich als Dienstvermittler zwischen den Nutzerparteien und dem Dienstanbieter erscheinen.

- **Szenario E-Business:** Ein anderes Szenario wäre im Bereich des E-Business denkbar. Das Geschäftsmodell soll dabei den Verkauf von Musikdateien realisieren. Die Bestandteile dieses Modells sind durch die Dienstbausteine: Warenkorb, Bezahlung, Katalog einschließlich Suchfunktion, sowie Buchhaltung beschrieben. Jeder dieser Teildienste kann von einem anderen Dienstleister angeboten werden. So könnte beispielsweise die Suchfunktion von einem Anbieter wie Google gekauft und in das beschriebene

## 1.1. FLEXIBILITÄTS-EIGENSCHAFTEN

---

Geschäftsmodell integriert werden. Über eine Kopplung der Schnittstellen und die Abbildung auf das gewünschte Produkt, in diesem Fall Musikdateien, oder andere, mit Musik assoziierte Produkte, wie Musik-CDs oder Wiedergabegeräte, kann das System schließlich realisiert werden.

- **Szenario Telekommunikationsdienst:** Ein letztes Beispiel soll die Notwendigkeit der Flexibilität veranschaulichen. Hierbei wird ein Telekommunikationsdienst betrachtet, der eine Reihe von Dienstmerkmalen zur Verfügung stellt, wie die Weiterleitung einer Verbindung (Call Forwarding), die Filterung von Verbindungen (Call Screening) oder der Aufbau von (Telefon-)Konferenzen (Conferencing), und die vom Dienstanbieter separat abonniert werden können. Der Anwender kann diese Merkmale zur Definition individueller Abläufe bei der Nutzung und Verwaltung seiner Kommunikations-Verbindungen verwenden. Bei diesem Szenario ist die Anforderung gegeben, dass die Nutzung des Dienstes unabhängig von dem zugrunde liegenden Übertragungsnetz und der -technologie möglich ist. Als Beispiel hierzu soll es dem Dienstanutzer möglich sein, seine individuellen Verbindungseinstellungen in seinem Mobiltelefon, seinem ISDN-Gerät oder seinem PC gleichermaßen zu nutzen.

Die Realisierung der beschriebenen Szenarien verursacht derzeit einen enormen Integrationsaufwand (vgl. [BGHV00, BGV02]). Daher sind neue Konzepte erforderlich, welche die geforderte Flexibilität bieten werden. Was unter dem Begriff der Flexibilität zu verstehen ist, wird im Folgenden auf der Basis der Szenarien beschrieben.

### 1.1 Flexibilitäts-Eigenschaften

Aus den eingangs beschriebenen Szenarien lassen sich die Eigenschaften an Diensten wie folgt ableiten (vgl. [GHH<sup>+</sup>01, FSV01]):

- **Autonomie:** Diese Eigenschaft besagt, dass Instanzen von Diensten oder Dienstbausteinen bei ihrer Funktionserbringung eigenständig und abgeschlossen sind. Dennoch sollte es möglich sein, auf der Basis dieser Bausteine, komplexe Funktionen kooperativ zu realisieren. Das Beispiel der integrierten Gebäudesysteme verdeutlicht die Notwendigkeit und Existenz dieser Eigenschaft. Die zur Realisierung des Szenarios erforderlichen Elemente wie Sensoren oder Aktoren sind eigenständige, integrierte Bausteine im System.
- **Komplexe Kommunikationsstrukturen:** Unter dieser Eigenschaft versteht man, dass bei der Erbringung von Dienstfunktionen vielfältige Kommunikationsstrukturen bei der Kooperation einzelner Dienstbausteine oder der Dienstanutzung etabliert werden können. Die Kommunikationsstrukturen beziehen sich auf die Abbildung von Verbindungsstrukturen wie Broad- oder Multicast. Ein Beispiel hierzu bietet das Szenario der integrierten Gebäudesysteme. Es wäre hierbei ein Dienst denkbar, der über die Funktion *Exit* alle elektrischen Geräte, die nicht zum Unterhalt eines Gebäudes erforderlich sind, gleichzeitig ausschaltet.