

Bettina Schewe  
Kooperative Softwareentwicklung

**Bettina Schewe**

# **Kooperative Softwareentwicklung**

**Ein objektorientierter Ansatz**



**DeutscherUniversitätsVerlag**

GABLER · VIEWEG · WESTDEUTSCHER VERLAG

**Schewe, Bettina:**

Kooperative Softwareentwicklung : ein objektorientierter Ansatz /  
Bettina Schewe. — Wiesbaden : DUV, Dt.Univ.-Verl., 1996  
Zugl.: Hamburg, Univ., Diss., 1995

Der Deutsche Universitäts-Verlag ist ein Unternehmen  
der Bertelsmann Fachinformation.

© Deutscher Universitäts-Verlag GmbH, Wiesbaden 1996



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Gedruckt auf chlorarm gebleichtem und säurefreiem Papier

ISBN-13: 978-3-8244-2070-4

e-ISBN-13: 978-3-322-85820-7

DOI: 10.1007/978-3-322-85820-7

# Vorwort

Der 25. Geburtstag des Begriffs 'Software-Engineering' (SE) hat eine Reihe grundsätzlicher Einschätzungen und zukunftsweisender Forderungen initiiert (vgl. z. B. die Beiträge in 'Informatik-Spektrum', Band 17, Heft 1). So plädiert Christiane Floyd für eine gewandelte Sicht auf die Software-Entwicklung: Weg von der 'Produktionssicht', die das *Produkt* Software als Hauptthema des Software-Engineerings ansieht und hin zur 'Design-Sicht', die dem *Entwurfsprozeß* eine große Auswirkung auf das Produkt zubilligt und ihn als Thema des Software-Engineerings sehen möchte.<sup>1</sup> Viele der Argumente kann ich aus der Praxis bestätigen. Vielleicht hat sich sogar diese Sicht in der Praxis bereits viel weiter durchgesetzt als es niedergeschriebene SE-Konzepte vermuten lassen. Die Reflexion der Erfahrungen aus einem großen Projekt 'Leistungswesen' bei einer Krankenkasse, das ich in den letzten Jahren geleitet habe, lieferten die erste Idee zu dieser Arbeit. Der zweite Teil der Idee zu dieser Arbeit stammt aus der (theoretischen) Auseinandersetzung mit dem objektorientierten Datenmodell OODM, das Klaus-Dieter Scheue, Bernhard Thalheim und Ingrid Wetzel entwickelten und theoretisch fundierten [STW92, ST93, SST94]. Ich habe dieses Modell an den Inhalten 'meines Projekts' erprobt, gemeinsam haben wir erste methodische Entwurfsschritte erarbeitet [SST93]. Dies war für mich ein neuer und vielversprechender Ansatz einer Methode für konzeptionelle Phasen, gemessen an ISOTEC [Plo89], mit dem ich bisher überwiegend gearbeitet und das ich mitentwickelt habe. In dem Projekt 'Leistungswesen' spielte die Benutzungsschnittstelle eine herausragende Rolle. Wir entwickelten eine Dialogsteuerung, die sich an den bearbeiteten Objekten der BenutzerInnen orientierte und die die dritte Idee zu meiner Arbeit lieferte.

In dieser Arbeit möchte ich eine objektorientierte Methode für den Entwurf datenintensiver Dialoganwendungen (OMEDA) vorstellen. Ich möchte damit

- das OODM um ein Dialogmodell erweitern, so daß der Entwurf von datenintensiven Dialogsystemen vollständiger abgedeckt wird,
- die Modelle mit Verfeinerungsschritten und Entwurfshinweisen versehen, die sie erst zu einer Methode machen,
- die Methode in ein Software-Engineering-Konzept einfügen, das dem 'Design'-Gedanken näherkommt als ältere Konzepte, und schließlich

---

<sup>1</sup> Dieser Artikel enthält auch wichtige Hinweise auf weitere Beiträge zu dieser Idee.

- den ‘Design-Gedanken’ anhand meiner Erfahrungen konkreter fassen und diskutieren.

Datenintensive Dialoganwendungen sind Dialoganwendungen, in denen die Präsentation und Bearbeitung von persistenten Daten im Mittelpunkt stehen. Sie finden Anwendung zur Unterstützung von Arbeitsvorgängen hochqualifizierter SachbearbeiterInnen, die die präsentierten Daten zur Kenntnis nehmen, einen Vorgang daraufhin beurteilen, die weiteren Dialogschritte selber auswählen und dabei die Daten verändern.

Solche Anwendungen verlangen eine Entwurfsmethode, die den Entwurf des Schemas mit dem Entwurf von Dialogen integriert. Zudem setzt der Entwurf die Einbeziehung der zukünftigen Anwender voraus, was an die Entwurfsmethode hohe Anforderungen in Bezug auf Handhabbarkeit, Verständlichkeit und Verwendung im iterativen Entwurfsprozeß stellt.

Basis für eine solche Entwurfsmethode ist ein objektorientiertes Daten- und Dialogmodell, in dem Grundbegriffe wie Objekt und Klasse benutzergerecht definiert werden. Das beinhaltet die grundsätzliche Unterscheidung von Werten und Objekten bzw. Typen und Klassen im Datenmodell und eine eingängige Modellierung von Objekten und ihrem Verhalten. Der gleiche Objektbegriff findet sich im Dialogmodell, wo Objekte einerseits ausgewählt werden, um auf ihnen Aktionen auszuführen und andererseits ihre Werte am Bildschirm dargestellt werden. Auswahlklassen und Dialogklassen bilden Sichten auf das Datenmodell.

Als Vorgehensweise beim Entwurf werden je nach Projektsituation zunächst Daten modelliert, aus ihnen Auswahlklassen und Dialogklassen abgeleitet und ihre Nutzbarkeit in Arbeitsvorgängen untersucht oder aber umgekehrt aus Arbeitsvorgängen die nötigen Dialogklassen ermittelt und aus den zugehörigen Sichten das Schema nach und nach aufgebaut. Der Entwurfsprozeß folgt grundsätzlich keinen starren Regeln; es werden stattdessen Verfeinerungsschritte angeboten, mit denen in einem Entwurf neue fachliche Informationen berücksichtigt werden können. Entwurfshinweise erlauben die Qualität eines Entwurfs zu überprüfen.

Den Rahmen der Entwurfsschritte bildet ein Vorgehensmodell für kooperative Softwareentwicklung. Eine Aufarbeitung weit verbreiteter SE-Konzepte einerseits, die i. a. aus Vorgehensmodell, Methoden, Projektmanagement-Konzept und Werkzeugen bestehen, sowie verwandter Konzepte andererseits runden die Arbeit ab.

Ich verzichte in dieser Arbeit auf die Formulierung männlicher und weiblicher Form, da dies die Lesbarkeit verschlechtert. Wo die männliche Form verwendet wird, sind immer auch weibliche Personen eingeschlossen und umgekehrt.

Diese Arbeit ist eine überarbeitete Fassung meiner unter dem Titel ‘Eine objektorientierte Methode für den Entwurf von datenintensiven Dialoganwendungen’ von der Universität Hamburg angenommenen Dissertation.

## Überblick über die vorliegende Arbeit

Die vorliegende Arbeit konzentriert sich auf Dialogsysteme in Dienstleistung und Verwaltung.

In Kapitel 1 erläutere ich die grundsätzliche Perspektive von Software als Werkzeug im Arbeitsprozeß, an der ich in späteren Kapiteln benutzergerechten Entwurf von Software messe. Ferner stelle ich das genannte Projekt vor.

In Kapitel 2 gebe ich einen praxisbezogenen, sicherlich nicht vollständigen Überblick über den Stand von SE-Konzepten.

Kapitel 3 stellt das STEPS-Vorgehensmodell [FRS89] vor und bettet OMEDA darin ein.

Kapitel 4 setzt sich mit verwandten Arbeiten zur Objektorientierung, zum Entwurf von Dialogsystemen und zu Softwareentwicklung mit Einbeziehung der Benutzerinnen auseinander.

Kapitel 5 stellt die Modelle zur Entwurfsmethode OMEDA vor, die Datenentwurf, Entwurf der Arbeitsvorgänge, Dialogentwurf und Operationentwurf beinhaltet. Es handelt sich dabei um die Definition der Strukturierungs-, Modellierungs- und Beschreibungsmittel, mit denen der Entwurf eines Anwendungssystems dokumentiert wird.

Kapitel 6 enthält die Beschreibung der Verfeinerungsschritte und Entwurfshinweise. Sie bilden das Regelwerk zur Ausführung des Entwurfs. Ich beschränke mich dabei auf das softwaretechnische Regelwerk. Die benötigten Organisations- und Koordinationsmethoden im Projekt liefert STEPS, Lern- und Arbeitsmethoden sind teilweise noch auszuarbeiten.

In Kapitel 7 ziehe ich ein abschließendes Fazit.

## Danksagung

Viele Menschen haben bewußt oder unbewußt zu dieser Arbeit beigetragen. Ihnen möchte ich an dieser Stelle danken: An erster Stelle Christiane Floyd, deren Gedanken zur Softwareentwicklung mir Orientierung für die Einordnung meiner Erfahrungen und meines Wissens in die Forschungsbereiche der Informatik gegeben haben. Ferner gilt mein Dank meinem Mann Klaus-Dieter sowie seinen Kollegen Bernhard Thalheim und Ingrid Wetzel, dessen gemeinsam verfaßte theoretische Ausarbeitungen es erlauben, in der Praxis entwickelte Lösungen nicht nur intuitiv richtig zu finden, sondern sie zu verallgemeinern und damit vermittel- und lernbar zu machen. Mein Mann war es auch, der mich zu dieser Arbeit ermutigte und immer wieder meine Entwürfe kritisch kommentierte und damit zur nächsten Verbesserung beitrug.

---

Weiter danke ich dem Team des Projekts 'Leistungswesen' einer hier nicht genannten Krankenkasse, das unter meiner methodischen Anleitung einen 'Prototyp' im Sinne der ersten Anwendung einer Variante der hier vorgestellten Methode entwickelte. Dazu gehörten ein Programmrahmen als Realisierung des Dialogmanagementsystems, Dialogspezifikationen und -programme, die unter dem Programmrahmen liefen, Datenschema und Dateien sowie Standards der Benutzungsschnittstelle, die in ausführlichen Diskussionen und Erprobungen anhand von Arbeitsvorgängen und einzelnen Dialogen gemeinsam festgelegt wurden.

# Inhaltsverzeichnis

<b>1 Die Rolle von Software im Arbeitsprozeß</b>	<b>1</b>
1.1 Software-Engineering und Arbeitswissenschaft . . . . .	1
1.2 Praktische Erfahrungen . . . . .	5
1.2.1 Gesetzliche Krankenkassen . . . . .	5
1.2.2 Das Projekt 'Leistungswesen' . . . . .	6
<b>2 Software-Engineering-Konzepte</b>	<b>11</b>
2.1 Terminologie . . . . .	13
2.2 Komponenten . . . . .	14
2.3 Notwendige Differenzierungen . . . . .	15
2.4 Kritiken und Weiterentwicklungen . . . . .	18
2.4.1 Grundsätzliche Kritik . . . . .	19
2.4.2 Kritik am Phasenmodell . . . . .	20
2.4.3 Kritik an Methoden . . . . .	22
2.4.4 Kritik an Projektmanagementkonzepten . . . . .	26
2.4.5 Kritik an der Werkzeugunterstützung . . . . .	26
<b>3 Ein Vorgehensmodell für OMEDA</b>	<b>29</b>
3.1 Produktions- und Designansicht . . . . .	30
3.2 Das STEPS-Konzept . . . . .	35
3.2.1 Das Projektmodell . . . . .	36
3.2.2 Aktivitäten der Systemgestaltung . . . . .	38
3.3 SE-Methoden in der Systemgestaltung . . . . .	38
3.3.1 Vorgehensweise im PETS-Projekt . . . . .	39
3.3.2 Beschreibungsmittel in PETS . . . . .	40
3.3.3 Theoriebildung und Modellierung . . . . .	40
3.3.4 Kritik der Trennung von Beschreibung und Modellierung . . . . .	42
3.3.5 Forderungen an geeignete SE-Methoden . . . . .	44
3.4 Systemgestaltung im Projekt 'Leistungswesen' . . . . .	46
3.4.1 Teilprojekt 'Stammdatenverwaltung und Neuzugänge' . . . . .	46



3.4.2	Teilprojekt 'Leistungsfälle' . . . . .	47
3.4.3	Teilprojekt 'übergreifende Anforderungen' . . . . .	48
3.5	SE-Methoden im STEPS-Projektrahmen . . . . .	49
3.5.1	Aktivitäten bei der Systemgestaltung . . . . .	50
3.5.2	Die Wahl der Vorgehensweise . . . . .	54
<b>4</b>	<b>Verwandte Konzepte</b>	<b>61</b>
4.1	Objektorientierter Dialogentwurf . . . . .	61
4.2	Objektorientierte Datenmodellierung . . . . .	64
4.3	Objektorientierte Programmierung . . . . .	66
4.4	Andere objektorientierte SE-Konzepte . . . . .	67
4.4.1	Object Modeling Technique (OMT) . . . . .	67
4.4.2	Objektorientierte Anwendungsentwicklung (OA) . . . . .	71
4.5	Andere Dialogentwurfsmethoden . . . . .	76
4.5.1	Der TASK-Ansatz . . . . .	76
4.5.2	Andere Ansätze . . . . .	80
4.6	Andere Konzepte mit Benutzerbeteiligung . . . . .	81
<b>5</b>	<b>Die Modelle in OMEDA</b>	<b>85</b>
5.1	Typen und Typoperationen . . . . .	86
5.1.1	Typen . . . . .	86
Basistypen.	. . . . .	86
Konstruktion von Typen.	. . . . .	87
5.1.2	Typ-Operationen . . . . .	92
5.1.3	Parametrisierte Typen . . . . .	94
5.1.4	Subtypen . . . . .	95
5.2	Das Datenmodell . . . . .	98
5.2.1	Objekte . . . . .	98
5.2.2	Klassen . . . . .	99
Einführung.	. . . . .	99
Klassen versus Objekte.	. . . . .	102
Identifikatoren.	. . . . .	102
Spezialisierung, Klassen und Superklassen.	. . . . .	103
5.2.3	Schema und Instanz. . . . .	104
5.2.4	Integritätsbedingungen. . . . .	108
5.2.5	Operationen . . . . .	110

---

Der Kopf einer Operation. . . . .	111
Der Rumpf einer Operation. . . . .	111
Anfragen. . . . .	120
Vererbung von Operationen. . . . .	120
5.3 Das Dialogmodell . . . . .	120
5.3.1 Vorbemerkungen . . . . .	121
Aufbau dieses Abschnitts. . . . .	121
Datenmodell und Dialogsystem. . . . .	121
Dialogmanagementsysteme. . . . .	122
Die Darstellung der Beispiele. . . . .	123
5.3.2 Dialogobjekte . . . . .	124
5.3.3 Dialogklassen . . . . .	128
Definition und Beispiel. . . . .	128
Auswahlklassen. . . . .	130
Die formale Beschreibung einer Dialogklasse. . . . .	134
Darstellungen von Dialogklassen. . . . .	135
Felder. . . . .	137
Super-Dialogklassen. . . . .	138
Dialogboxen. . . . .	142
Dialogsysteme. . . . .	143
5.3.4 Aktionen und Operationen . . . . .	144
Aktionen. . . . .	145
Das Zusammenspiel von DMS und Dialogsystem. . . . .	152
Vererbung von Aktionen. . . . .	154
Die Aufruf-Operation. . . . .	157
Navigationsoperationen. . . . .	159
Bearbeitungsoperationen. . . . .	160
Auswahloperationen. . . . .	162
5.4 Sichten: Zusammenhang von Daten- und Dialogmodell . . . . .	165
5.4.1 Einführung . . . . .	165
5.4.2 Definitionen und Beispiele . . . . .	167
5.5 Das Aufgabenmodell . . . . .	172
5.5.1 Ziele der Aufgabenmodellierung . . . . .	172
5.5.2 Das Aufgabenmodell . . . . .	174

---

<b>6 Verfeinerungsschritte und Entwurfshinweise</b>	<b>179</b>
6.1 Verfeinerungsschritte . . . . .	180
6.2 Verfeinerung von Typen . . . . .	181
6.2.1 Verfeinerung durch Instanziierung . . . . .	181
6.2.2 Verfeinerung durch Erweiterung . . . . .	182
6.2.3 Restrukturierung von Typen . . . . .	183
6.2.4 Operationale Verfeinerungen . . . . .	184
6.3 Entwurfshinweise für Typen . . . . .	185
6.3.1 Fachliche Relevanz von Typen . . . . .	185
6.3.2 Mehrfachverwendung von Typen . . . . .	185
6.4 Verfeinerung des Datenschemas . . . . .	186
6.4.1 Instanziierung von Klassen und Operationen . . . . .	187
6.4.2 Verfeinerung durch Abspaltung . . . . .	188
6.4.3 Verfeinerung durch Zusammensetzen . . . . .	190
6.4.4 Verfeinerung durch Erweiterung . . . . .	190
6.4.5 Verfeinerung durch Spezialisierung von Klassen . . . . .	191
6.4.6 Verfeinerung durch Spezialisierung von Strukturausdrücken	194
6.4.7 Verfeinerung durch Generalisierung . . . . .	195
6.4.8 Weitere operationale Verfeinerungen . . . . .	196
6.4.9 Ergänzen von Integritätsbedingungen . . . . .	196
6.5 Entwurfshinweise zum Schema . . . . .	196
6.5.1 Fachliche Relevanz und Eigenständigkeit von Klassen . . .	197
6.5.2 Minimierung von Integritätsbedingungen . . . . .	197
6.5.3 Redundanzfreiheit des Schemas . . . . .	199
6.6 Verfeinerungen im Dialogschema . . . . .	200
6.6.1 Verfeinerung durch Instanziierung . . . . .	201
6.6.2 Verfeinerung durch Abspaltung . . . . .	202
6.6.3 Verfeinerung durch Zusammensetzen . . . . .	203
6.6.4 Verfeinerung durch Erweiterung . . . . .	204
6.6.5 Verfeinerung durch Spezialisierung . . . . .	204
6.6.6 Verfeinerung durch Generalisierung . . . . .	205
6.6.7 Horizontale Zerlegung . . . . .	206
6.6.8 Ausarbeiten von Sichten . . . . .	207
6.6.9 Weitere Verfeinerungen von Operationen und Aktionen . .	208
6.7 Entwurfshinweise zum Dialogschema . . . . .	208

---

6.7.1 Fachliche Relevanz und Eigenständigkeit von Dialogklassen	208
6.7.2 Auswahlklasse und Datenschema . . . . .	209
6.7.3 Andere Arbeitsweisen als Objekt-Aktions-Ansätze . . . . .	210
6.7.4 Modellierung von Aktionen . . . . .	211
6.7.5 Extensive Verwendung von Dialogboxen . . . . .	212
6.8 Entwurfshinweise zum Aufgabenmodell . . . . .	213
Arbeitsabläufe sind komplex. . . . .	213
6.9 Ein Beispiel . . . . .	215
6.9.1 Entwurf des Schemas . . . . .	215
6.9.2 Entwurf des Dialogschemas . . . . .	222
<b>7 Kritische Bewertung</b>	<b>227</b>