

Karin Küffmann

**Software-
Wiederverwendung**

Programm Angewandte Informatik

hrsg. von Paul Schmitz und Norbert Szyperski

Die Reihe hat es sich zum Ziel gesetzt, Studenten, Ingenieure und DV-Praktiker mit zentralen Fragestellungen der Angewandten Informatik vertraut zu machen. Auch wenn in Werken dieser Reihe theoretische Grundlagen vermittelt werden, so stehen sie doch stets in Zusammenhang mit konkreten Anwendungen.

Die Reihe umfaßt sowohl grundlegende Einführungen, die den State-of-the-Art eines aktuellen Fachgebietes zur Darstellung bringen, wie auch speziellere Monographien, sofern sie der o.g. Zielsetzung entsprechen.

Unter anderem sind bisher folgende Titel erschienen:

Agentensysteme

Verteiltes Problemlösen mit Expertensystemen
von M. v. Bechtolsheim

Petri-Netze

Eine anwendungsorientierte Einführung
von B. Rosenstengel und U. Winand

Wissensbasiertes CASE

Theoretische Analyse – Empirische Untersuchung – Prototyp
von G. Herzwurm

Methoden verteilter Simulation

von H. Mehl

Echtzeitsysteme und Fuzzy Control

Konzepte, Werkzeuge, Anwendungen
von H. Rzehak (Hrsg.)

Software-Wiederverwendung

Konzeption einer domänenorientierten Architektur
von K. Küffmann

Vieweg

Karin Küffmann

Software- Wiederverwendung

Konzeption einer
domänenorientierten Architektur

Mit einem Geleitwort
von Ulrich Hasenkamp



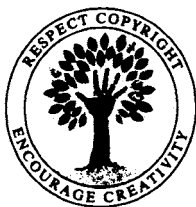
Das in diesem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor, die Herausgeber und der Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1994

Softcover reprint of the hardcover 1st edition 1994

Der Verlag Vieweg ist ein Unternehmen der Verlagsgruppe Bertelsmann International.



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.

ISBN 978-3-528-05457-1

ISBN 978-3-322-86226-6 (eBook)

DOI 10.1007/978-3-322-86226-6

Für
Michael,
Achim,
Bernd
und meine Eltern

Geleitwort

Die Softwareentwicklung ist heute durch die Probleme des Anwendungsstaus, verbunden mit einer geringen Produktivität der Anwendungsentwicklung und einer weitgehenden Bindung von Entwicklerkapazität in Wartungstätigkeiten gekennzeichnet. Die Anforderungen an die Anwendungsentwicklung wachsen hingegen stetig. Die Anwendungssysteme sollen die Aufgaben in den Unternehmen effizient unterstützen und hohen Qualitätsanforderungen standhalten. Eine mögliche Lösung stellt die Realisierung der vertikalen domänenspezifischen Wiederverwendbarkeit von Softwarebauteilen dar. Eine Domäne stellt einen Ausschnitt aus der realen Welt dar, der in einem Anwendungssystem abgebildet wird. Es zeichnet sich der Trend zur Wiederverwendung des anwendungsspezifischen Wissens in den Entwurfsprodukten des Anwendungssystems innerhalb einer Domäne ab.

Die vorliegende Arbeit beschäftigt sich mit der Wiederverwendbarkeit in einer domänenspezifischen und integrationsfördernden Architektur. Es wird zunächst der Entwurf wiederverwendbarer Software in den Mittelpunkt gestellt und Entwurfskriterien sowie Anwendungsvoraussetzungen erarbeitet, die die Wiederverwendbarkeit unterschiedlicher Objekte, von spezifischem Wissen über Teile der Spezifikation bis hin zu Modulen, ermöglichen. Durch die Konzeption einer Architektur, die auf diesen identifizierten Entwurfskriterien und Anwendungsvoraussetzungen aufbaut, wird eine Standardisierung der betrachteten Entwicklungsobjekte derart erreicht, daß sie wiederverwendbar, austauschbar und anpaßbar werden. Durch die Entwicklung getesteter, domänenspezifischer Modelle jeder Entwurfsstufe wird neben der Wiederverwendbarkeit der Modelle an sich, ihrer Folgeprodukte und des in ihnen dokumentierten Entwicklungswissens auch die Erhöhung der Qualität und Produktivität der Entwicklung erreicht. Darüber hinaus fördert die domänenspezifische vertikale Wiederverwendung die Integration von Anwendungssystemen in dieser Domäne und auch zwischen verschiedenen Domänen durch die Modellierung auf einer konzeptuellen Metaebene.

Diese Arbeit betrachtet nicht die einsetzbaren Techniken zur Realisierung von Wiederverwendungskonzepten; vielmehr konzentriert sie sich auf den Entwurf von wiederverwendbaren Bauteilen, um zu Aussagen zur Standardisierung des Entwurfs und der Anwendungsvoraussetzungen zu gelangen. Diese Standardisierung macht das Wiederverwendungskonzept praktisch einsetzbar.

Marburg, im Juni 1994

Prof. Dr. Ulrich Hasenkamp

Danksagung

Herrn Prof. Dr. Ulrich Hasenkamp möchte ich für die Betreuung und Korrektur der Arbeit danken. Auch Herrn Prof. Dr. Erich Priewasser sei für die Übernahme des Koreferats herzlich gedankt.

Herr Dr. Michael Syring hat den Part des Diskussionspartners übernommen. Ihm sind seine kritischen Fragen und seine Geduld hoch anzurechnen. Er hat damit zur ständigen Weiterentwicklung der Arbeit beigetragen.

Von Seite des Verlages sei Herrn Prof. Dr. Paul Schmitz gedankt, dessen Hinweise die Arbeit sicherlich verbessert haben.

Marburg, im Juni 1994

Karin Küffmann

Inhaltsverzeichnis

| | |
|--|----|
| Bilderverzeichnis..... | 13 |
| 1 Einleitung..... | 15 |
| 1.1 Einführung in das Themengebiet | 15 |
| 1.2 Zielsetzung | 17 |
| 2 Begriffs- und Themenabgrenzung..... | 18 |
| 2.1 Definition von Software | 18 |
| 2.2 Anwendungssysteme im Unternehmen..... | 19 |
| 2.2.1 Planung von Informationssystemen | 19 |
| 2.2.2 Wertkettenmodell | 21 |
| 2.2.3 Anwendungsportfolio..... | 22 |
| 2.2.4 Anwendungssystemplanung | 24 |
| 2.3 Domäne und Domänenanalyse | 25 |
| 2.4 Softwarewiederverwendung | 26 |
| 2.4.1 Begriffsvielfalt..... | 26 |
| 2.4.2 Aktivitätsausprägung | 28 |
| 2.4.3 Objektausprägung | 29 |
| 2.4.4 Vertikale Wiederverwendung..... | 31 |
| 2.5 Motivation für die Wiederverwendung..... | 32 |
| 2.5.1 Erhöhung der Produktivität | 32 |
| 2.5.1.1 Produktivitätspotential | 32 |
| 2.5.1.2 Reduzierung des Entwicklungsaufwandes | 33 |
| 2.5.1.3 Reduzierung des Wartungsaufwandes | 34 |
| 2.5.1.4 Erhöhung der Softwarequalität | 36 |
| 2.5.2 Größe der Objekte und Produktivität | 37 |
| 2.6 Probleme | 39 |
| 3 Wiederverwendungsorientierte Systementwicklung | 41 |
| 3.1 Systementwicklung | 41 |

| | | |
|-----------|---|----|
| 3.1.1 | Prinzipien | 41 |
| 3.1.2 | Wiederverwendungsfördernde Entwurfsprinzipien..... | 42 |
| 3.1.3 | Methoden | 44 |
| 3.2 | Vorgehensmodell der Entwicklung | 45 |
| 3.3 | Wiederverwendungsorientierter Softwareentwurf..... | 48 |
| 3.3.1 | Virtuelles Maschinenkonzept..... | 49 |
| 3.3.2 | Standardisierte Softwarearchitektur | 49 |
| 3.3.3 | Konzept der Programmfamilie..... | 51 |
| 3.3.4 | Entwurf von Modulen | 53 |
| 3.3.4.1 | Funktionen- und datenorientierte Abstraktion..... | 54 |
| 3.3.4.2 | Repräsentation..... | 55 |
| 3.3.4.3 | Allgemeine und spezielle Module | 56 |
| 3.3.4.4 | Entwurf funktionaler Module | 57 |
| 3.3.4.5 | Datenabstraktion..... | 57 |
| 3.3.4.5.1 | Abstrakte Datentypen..... | 57 |
| 3.3.4.5.2 | Objektorientierte Module | 58 |
| 3.3.4.6 | Interface-Probleme | 60 |
| 4 | Ansätze zur Wiederverwendung | 62 |
| 4.1 | Ähnlichkeit als Voraussetzung für Wiederverwendbarkeit..... | 62 |
| 4.1.1 | Funktionale Ähnlichkeit in betriebswirtschaftlichen Anwendungssystemen | 63 |
| 4.1.2 | Ähnlichkeiten transformationsorientierter Module | 64 |
| 4.1.3 | Ähnlichkeiten von Datenstrukturen | 65 |
| 4.2 | Wiederverwendungsobjekte und ihr Entwurf | 66 |
| 4.2.1 | Module | 66 |
| 4.2.1.1 | Regeln zum Aufbau von Modulen | 67 |
| 4.2.1.1.1 | Transformationsorientierte Module..... | 67 |
| 4.2.1.1.2 | Abstrakte Datentypen und objektorientierte Module | 70 |
| 4.2.1.2 | Sprachkonzepte | 73 |
| 4.2.1.3 | BB/LX-Ansatz | 80 |
| 4.2.1.4 | Zusammenfassung | 82 |
| 4.2.2 | Programmdesignorientierte Programmstrukturen | 83 |
| 4.2.3 | Wiederverwendbarkeit von Design | 87 |

| | |
|--|-----|
| 4.2.3.1 Designwiederverwendbarkeit mittels einer Softwarearchitektur | 87 |
| 4.2.3.2 Designwiederverwendbarkeit durch einheitliche Methoden | 91 |
| 4.2.3.3 Zusammenfassung | 92 |
| 4.2.4 Wiederverwendbarkeit von Spezifikationen..... | 93 |
| 4.2.4.1 Interaktive Ableitung neuer aus alten Spezifikationen..... | 94 |
| 4.2.4.2 Spezifikationswiederverwendung durch Analogiebildung | 100 |
| 4.2.4.3 Zusammenfassung | 105 |
| 4.2.5 Domänen- und familienspezifische Wiederverwendung..... | 106 |
| 4.2.6 Wiederverwendbarkeit von Wissen | 110 |
| 4.2.6.1 Erfahrungswissen..... | 110 |
| 4.2.6.2 Domänenspezifisches Wissen..... | 114 |
| 4.2.6.3 Prozeßwissen | 117 |
| 4.2.6.4 Zusammenfassung | 123 |
| 4.3 Anwendungsvoraussetzungen einer Wiederverwendungs- konzeption..... | 125 |
| | |
| 5 Anforderungen an eine wiederverwendungsfördernde Architektur..... | 138 |
| 5.1 Forderung nach einem domänenspezifischem Vorgehensmodell | 138 |
| 5.2 Vertikale Wiederverwendbarkeit | 139 |
| 5.3 Domänen- und Vorgehensmodell zur Erstellung wiederverwendbarer Bauteile | 141 |
| 5.3.1 Forderungen an ein Domänenmodell..... | 141 |
| 5.3.1.1 Entwurfsebenen | 142 |
| 5.3.1.2 Inhalt des Domänenmodells | 143 |
| 5.3.1.3 Abbildung der Domäne..... | 144 |
| 5.3.1.4 Integration von Domänenmodellen | 145 |
| 5.3.2 Forderungen an ein Vorgehensmodell | 146 |
| 5.4 Architekturrahmen | 147 |
| | |
| 6 Konzeption einer wiederverwendungsfördernden Architektur | 150 |
| 6.1 Architekturbegriff..... | 150 |
| 6.2 Architekturen für Informationssysteme | 151 |

| | |
|--|-----|
| 6.2.1 Information System Architecture | 151 |
| 6.2.2 Informationssystem-Architektur | 154 |
| 6.2.3 CIM-Open System Architecture | 156 |
| 6.2.4 Systems Application Architecture | 160 |
| 6.2.5 Softwarearchitekturen..... | 166 |
| 6.2.6 Zusammenfassung | 167 |
| 6.3 Architektur zur domänenorientierten und integrierten Wiederver- wendung von Software (ADONIS) | 170 |
| 6.3.1 Grundkonzeption einer wiederverwendungsfördernden Architektur..... | 171 |
| 6.3.2 Domänenreferenzmodell | 174 |
| 6.3.3 Ebenen des Domänenreferenzmodells | 177 |
| 6.3.4 Sichten im Domänenreferenzmodell..... | 182 |
| 6.3.5 Anpassung des generischen Referenzmodells | 186 |
| 6.3.6 Vorgehensmodell | 198 |
| 6.3.7 Informationsmodellierung in einem Unternehmen..... | 203 |
| 7 Ergebnisse der Arbeit und Ausblick..... | 207 |
| 7.1 Beitrag von ADONIS zur vertikalen Wiederverwendung..... | 207 |
| 7.2 Integration in das Anwendungsportfolio | 209 |
| 7.3 Umdenken in der Softwareentwicklung | 210 |
| 7.4 Markt für domänenspezifische Bauteile..... | 211 |
| 7.5 Ausblick | 212 |
| Literaturverzeichnis | 213 |
| Sachwortverzeichnis | 243 |

Bilderverzeichnis

| | |
|---|-----|
| Bild 1.1: Eingesetzte Techniken zur Wiederverwendung | 16 |
| Bild 2.1: Einfluß der Umwelt auf die Strategische Informationssystemplanung | 20 |
| Bild 2.2: Wertkettenmodell | 21 |
| Bild 2.3: Beispielhafte Einordnung von Anwendungssystemen in das Wertkettenmodell | 23 |
| Bild 2.4: Wertschöpfungskette und Informationsfluß..... | 24 |
| Bild 2.5: Zusammenfassung der Facetten der Wiederverwendung | 27 |
| Bild 2.6: Generierende und zusammensetzende Technik | 28 |
| Bild 2.7: Hierarchie der Softwareentwicklungsprodukte..... | 30 |
| Bild 2.8: Geschätzte Produktivitätseffekte der Wiederverwendung | 33 |
| Bild 2.9: Phasenaufwand zur Erstellung eines Softwareproduktes | 34 |
| Bild 2.10: Qualitätskriterien zur Erreichung bestimmter Qualitätsfaktoren..... | 35 |
| Bild 2.11: Größe, Häufigkeit und anteilige Wiederverwendung | 38 |
| Bild 3.1: Vorgehensmodell von Denert..... | 46 |
| Bild 3.2: Schichtenmodell der virtuellen Maschinen..... | 49 |
| Bild 3.3: Softwarearchitektur | 50 |
| Bild 3.4: Programmfamilien | 52 |
| Bild 4.1: Relative Häufigkeiten von Funktionen in kommerziellen Programmen | 63 |
| Bild 4.2: Wiederverwendungsfördernde und -hemmende Richtlinien bei EVB.. | 71 |
| Bild 4.3: Beispiel eines generischen Objektes in OBJ2 | 74 |
| Bild 4.4: Spezielles Interface | 78 |
| Bild 4.5: Allgemeines Interface | 78 |
| Bild 4.6: Templates Method Overview | 79 |
| Bild 4.7: Inline-Generierung von Building Blocks | 81 |
| Bild 4.8: Architektur der Software- und Datenbasis | 90 |
| Bild 4.9: Compute-Dependent-Revenue-Schema..... | 97 |
| Bild 4.10: Vorgehen von CORE | 99 |
| Bild 4.11: Vergleich der Syntax und der Beziehungen | 103 |
| Bild 4.12: Syntaktische Ähnlichkeiten und kausale Beziehungen..... | 103 |
| Bild 4.13: Vorschlag eines abstrakten Domänenmodells | 104 |
| Bild 4.14: Nutzung der Entwurfsstufen in der vertikalen Integration | 116 |

| | |
|---|-----|
| Bild 4.15: Zusammenhang zwischen Produkt und Vorgehen | 125 |
| Bild 4.16: Kostenreduktion in der Entwicklung..... | 129 |
| Bild 5.1: Abhängigkeit der wiederverwendbaren Bauteile von der Domäne .. | 138 |
| Bild 5.2: Abgrenzung von Domänenmodellen..... | 144 |
| Bild 5.3: Grobkonzept einer domänenspezifischen Architektur..... | 149 |
| Bild 6.1: Information System Architecture..... | 152 |
| Bild 6.2: Informationssystem-Architektur | 155 |
| Bild 6.3: CIM-OSA Framework | 157 |
| Bild 6.4: Ableitung des CIM-OSA Partial Level | 158 |
| Bild 6.5: General Software Structure..... | 161 |
| Bild 6.6: SAA-Architektur..... | 162 |
| Bild 6.7: Grundkonzept einer wiederverwendungsorientierten Architektur ... | 172 |
| Bild 6.8: Mögliche Hierarchie der Entwurfsstufen in einer Programmfamilie . | 174 |
| Bild 6.9: Verquickung der Entwurfsstufen und ihrer Sichten..... | 176 |
| Bild 6.10: Entwurfsebenen eines generischen Domänenreferenzmodells | 178 |
| Bild 6.11: Eingliederung eines Domänenreferenzmodells in die Entwicklung... | 181 |
| Bild 6.12: Allgemeine Sichten von Domänenreferenzmodellen in ADONIS | 183 |
| Bild 6.13: Perspektiven auf das Anwendungssystem | 184 |
| Bild 6.14: Instanziierung und Generalisierung | 187 |
| Bild 6.15: Teilvorgehensmodelle | 199 |
| Bild 6.16: Aufspaltung des Vorgehensmodells für eine Teilarchitektur | 201 |
| Bild 6.17: Ableitung von aufgabenspezifischer IT-Unterstützung..... | 204 |
| Bild 6.18: Eingliederung der domänenspezifischen Referenzmodelle | 205 |