Xpert.press

Springer-Verlag Berlin Heidelberg Gmbh

Die Reihe **Xpert.press** des Springer-Verlags vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Bertram Neubauer · Tom Ritter · Frank Stoinski

CORBA Komponenten

Effektives Software-Design und Programmierung

Mit 45 Abbildungen

Springer

Bertram Neubauer
Humboldt-Universität Berlin, Institut für Informatik
Unter den Linden 6, 10099 Berlin

Tom Ritter
Fraunhofer Institut FOKUS
Kaiserin-Augusta-Allee 31, 10589 Berlin

Frank Stoinski
Humboldt-Universität Berlin, Institut für Informatik
Unter den Linden 6, 10099 Berlin

Additional material to this book can be downloaded from http://extras.springer.com.
ISSN 1439-5428
ISBN 978-3-642-62161-1 ISBN 978-3-642-18566-3 (eBook)
DOI 10.1007/978-3-642-18566-3

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, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Der Springer-Verlag ist nicht Urheber der Daten und Programme. Weder der Springer-Verlag noch der Autor übernehmen Haftung für die CD-ROM und das Buch, einschließlich ihrer Qualität, Handels- oder Anwendungseignung. In keinem Fall übernehmen der Springer-Verlag oder der Autor Haftung für direkte, indirekte, zufällige oder Folgeschäden, die sich aus der Nutzung der CD-ROM oder des Buches ergeben.

springer.de

© Springer-Verlag Berlin Heidelberg 2004
Ursprünglich erschienen bei Springer-Verlag Berlin Heidelberg New York 2004
Softcover reprint of the hardcover 1st edition 2004

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Umschlaggestaltung: KünkelLopka, Heidelberg
Satz: Belichtungsfertige Daten von den Autoren
Gedruckt auf säurefreiem Papier 33/3142 GF 543210

Vorwort

Die Übertragung der allgemeinen Idee von Komponenten auf den Bereich von Softwareanwendungen konnte sich in den letzten Jahren verstärkt durchsetzen. Hauptursache dafür ist die Möglichkeit der raschen und kosteneffizienten Implementierung neuer Anwendungen auf der Grundlage vorgefertigter, konfigurierbarer Bausteine, der Komponenten. Obwohl CORBA seit Jahren eine Schlüsselstellung bei der Realisierung verteilter Anwendungen einnimmt, war bis vor kurzem eine komponentenorientierte Herangehensweise bei der Entwicklung von CORBA-Anwendungen nicht standardisiert Mit dem CORBA-Komponentenmodell schließt die OMG diese Lücke.

Das CORBA-Komponentenmodell führt einen Komponentenbegriff ein, der mit CORBA und relevanten CORBA-Standarddiensten integriert ist. Eine CORBA-Komponente erweitert dabei den Begriff des Objekts um die Reflektion von Informationen zur Erzeugung von Verbindungen zwischen Komponenten. Ähnlich zu entsprechenden Konzepten in der jüngsten Sprachrevision der *Unified Modeling Language*, UML 2.0, können Anschlussstellen definiert werden, die die Implementierung oder Verwendung von Schnittstellen, sowie das Senden oder Empfangen von Ereignissen ausdrücken. Zum Betrieb benötigt eine CORBA-Komponente eine erweiterte Infrastruktur, deren zentraler Bestandteil der Container ist. Der Vorteil für den Entwickler besteht darin, dass große Teile der Implementierung automatisch generiert werden können und er somit von immer wiederkehrenden und fehlerträchtigen Implementierungsaufgaben entbunden wird. Darüber hinaus kann auch die Inbetriebnahme kompletter Anwendungen nahezu automatisch auf der Grundlage der Spezifikation der Anwendung erfolgen.

Dieses Buch unternimmt den Versuch, die durch *die Object Management Group* (OMG) veröffentlichte Spezifikation des CORBA-Komponentenmodells in der Version 1.1 für den Software-Entwickler zu erschließen. Diskutiert wird der komplette Entwicklungsweg von der Spezifikation einer CORBA-Komponente in der *Interface Definition Language* und im UML-Profil für das CORBA-Kom-

ponentenmodell über die Spezifikation einer Implementierung für die Komponente in der *Component Implementation Definition Language* und die werkzeugunterstützte Generierung und Implementierung der Komponente bis zur Komposition von Anwendungen aus vorgefertigten CORBA-Komponenten und deren Inbetriebnahme.

Dieses Buch zu schreiben war für uns eine große Herausforderung. Darüber hinaus verlangte es auch von Freunden und Verwandten, sich in dieser Zeit in Geduld und Nachsicht zu üben. Unser besonderer Dank gilt unseren Familien, die uns die vielen Arbeitsabende stets wohlwollend verziehen haben und uns damit die Zeit gaben, dieses Buch zu schreiben. Unser Dank gilt auch Harald Böhme, mit dem wir gemeinsam in vielen kritischen Diskussionen in die Tiefen des Standards des CORBA-Komponentenmodells vorgestoßen sind. Des Weiteren bedanken wir uns bei den Mitarbeitern des Lehrstuhls Systemanalyse der Humboldt-Universität zu Berlin und des Fraunhofer Instituts Fokus, insbesondere Klaus-Peter Eckert und Klaus-D. Engel, für die Unterstützung und die kritische Durchsicht des Manuskripts dieses Buches.

Berlin, im März 2004

Bertram Neubauer
Tom Ritter
Frank Stoinski

Leitfaden zum Lesen

Das vorliegende Buch ist in vier große Teile gegliedert. Der erste Teil ist der Theorie gewidmet. Diskutiert werden zuerst die Theorie der komponentenbasierten Softwareentwicklung im Allgemeinen und anschließend speziell die theoretischen Grundlagen des CORBA-Komponentenmodells. Im zweiten Teil geht es um die praktische Komponentenentwicklung. Gezeigt wird an mehreren Beispielen, wie die Modellierungskonzepte und Implementierungskonzepte eingesetzt werden, um zu fertig implementierten und komponierbaren Komponenten zu kommen. Im dritten Teil wird die Komposition von Anwendungen aus CORBA-Komponenten diskutiert. Dabei wird auch auf die spätere Inbetriebnahme der Anwendung in einer IT-Umgebung eingegangen. Im vierten Teil geht es schließlich mit der Software Qedo um eine konkrete Implementierung des CORBA-Komponentenmodells. Die auch auf der beiliegenden CD zu findende Version dieser Software kann für erste eigene Schritte mit dem CORBA-Komponentenmo-dell genutzt werden.

Im Buchtext werden häufig verschiedene Schriftarten eingesetzt, um bestimmte Wörter und Wortgruppen vom umgebenden Text abzuheben. Erstmalig eingeführte Begriffe werden *kursiv* geschrieben. Das gilt auch für die Erwähnung englischer Wörter, die sich ansonsten in den normalen Text einbetten sollen und nicht übersetzt werden. Treten Bezeichner und Elemente aus einem Programmtext oder einer textuellen Spezifikation im laufenden Text auf, werden sie immer in `Courier` gesetzt. Längere Fragmente eines Programmtextes oder einer textuellen Spezifikation sind ebenfalls in `Courier` gesetzt. Bei diesen Fragmenten ist zu beachten, dass an Stellen, an denen Teile des Originaltextes ausgelassen wurden, mit dem Auslassungszeichen „. . .“ in einer eigenen Zeile gearbeitet wird. Zur Vereinfachung wurde an einigen Stellen im weiteren Verlauf des Buches anstelle von Standard des CORBA-Komponentenmodells nur vom Standard gesprochen.

Inhaltsverzeichnis

Teil I – Theorie des CORBA-Komponentenmodells

1 CORBA 3

1.1 Überblick CORBA 3
1.2 Object Request Broker 7
1.3 Interface Definition Language 10
1.3.1 Sprachabbildungen 11
1.4 Portable Object Adapter 13
1.4.1 Servant Manager 14
1.5 General Inter-ORB Protocol 14
1.5.1 Common Data Representation 16
1.5.2 Interoperable Object Reference 17
1.6 Standarddienste in CORBA 18

2 Das CORBA-Komponentenmodell 21

2.1 Komponentenbegriff 21
2.1.1 Was ist eine Komponente? 24
2.1.2 Was ist ein Objekt? 26
2.1.3 Komponenten und Objekte 27
2.1.4 Instanzen 27
2.1.5 Komponententyp 28
2.1.6 Sichtbarkeit und Wiederverwendung 29
2.1.7 Eine abschließende Definition 31
2.2 CORBA-Komponenten 31
2.2.1 Der Metatyp CORBA-Komponente 32
2.2.2 Ports 33
2.2.3 Das Komponenteninterface 34
2.2.4 Facets 35
2.2.5 Receptacles 36
2.2.6 Ereignisproduzenten 37

2.2.7 Ereigniskonsumenten ... 38
2.2.8 Attribute ... 39
2.3 Homes ... 40

3 Der Container ... 43

3.1 Laufzeitumgebung einer CORBA-Komponente ... 43
3.2 Standarddienste des Containers ... 46
3.2.1 Transaktionen ... 49
3.2.2 Sicherheit ... 50
3.2.3 Automatische Speicherung ... 52
3.3 Komponentenkategorien ... 55
3.4 Containertypen ... 60
3.4.1 Der Komponenten-Kontext ... 61
3.4.2 Callback-Interfaces ... 62
3.4.3 Allgemeine Eigenschaften des Containers ... 63
3.4.4 Eigenschaften des Session-Containers ... 64
3.4.5 Eigenschaften des Entity-Containers ... 64
3.5 Der Komponentenserver ... 65
3.6 Der Komponentenserver-Aktivator ... 67

4 Implementierungsstruktur ... 69

4.1 Komposition ... 69
4.2 Executors ... 72
4.3 Segmente ... 75
4.4 Dauerhafter Zustand ... 78
4.5 Proxy Homes ... 80
4.6 Komponentenkategorien ... 81

5 Komponentenbasierte Anwendungen ... 83

5.1 Was ist eine komponentenbasierte Anwendung? ... 84
5.2 Beispielanwendungen ... 85
5.2.1 Hello World ... 86
5.2.2 Philosophen ... 86
5.3 Bereitstellungsformate ... 88
5.3.1 Metainformationen ... 88
5.3.2 Archive ... 89
5.4 Bereitstellung von Komponenten ... 90
5.5 Bereitstellung von Anwendungen ... 92
5.6 Konfiguration ... 94
5.7 Infrastruktur für Anwendungen ... 97
5.7.1 Werkzeugunterstützung ... 98
5.7.2 Plattformunterstützung ... 102
5.8 Inbetriebnahme von Anwendungen ... 105

5.8.1 Prozess der Inbetriebnahme 105
5.8.2 Szenario 107

6 Rollen und Entwicklungsprozesse 109

6.1 Rollen im CORBA-Komponentenmodell 109
6.1.1 Der Nutzer 110
6.1.2 Der Komponentenentwickler 110
6.1.3 Der Anwendungsentwickler 112
6.1.4 Der Installateur 112
6.1.5 Der Containerhersteller 113
6.1.6 Der Systemadministrator 115
6.2 Der Komponentenentwicklungsprozess 115
6.2.1 Entwurf des Komponententyps 115
6.2.2 Implementierung der Komponenten 119
6.2.3 Bereitstellung 120
6.3 Der Anwendungsentwicklungsprozess 121
6.3.1 Entwurf und Spezifikation 121
6.3.2 Implementierung und Bereitstellung 123
6.3.3 Inbetriebnahme 123

Teil II – Entwicklung von Komponenten

7 Modellierungskonzepte 127

7.1 Modellierung 127
7.1.1 Metamodell des CORBA-Komponentenmodells 128
7.1.2 Modellierungssprache 130
7.2 Verfeinerung der Beispiele 132
7.2.1 Hello World 132
7.2.2 Speisende Philosophen 133

8 Komponentendesign 137

8.1 Entwurf eines Komponententyps 137
8.1.1 Datentypen 137
8.1.2 Interfaces 143
8.1.3 Valuetypes 145
8.1.4 Komponenten 148
8.1.5 Facets 150
8.1.6 Receptacles 152
8.1.7 Ereignistypen 154
8.1.8 Ereignisproduzenten 156

8.1.9 Ereigniskonsumenten ... 158
8.1.10 Attribute ... 159
8.1.11 Home ... 160
8.2 Beispielhafte Komponentendefinition ... 163
8.2.1 Hello World ... 163
8.2.2 Speisende Philosophen ... 165

9 Äquivalente Typumwandlung ... 171

9.1 Motivation ... 171
9.2 Komponenteninterface ... 172
9.2.1 Umwandlung eines Facet ... 173
9.2.2 Umwandlung eines Receptacle ... 173
9.2.3 Umwandlung eines Ereignistyps ... 176
9.2.4 Umwandlung von Ereignisproduzenten ... 178
9.2.5 Umwandlung von Ereigniskonsumenten ... 180
9.2.6 Das Interface CCMObject ... 181
9.2.7 Das Interface Navigation ... 182
9.2.8 Das Interface Receptacles ... 184
9.2.9 Das Interface Events ... 187
9.3 Home-Interface ... 191
9.3.1 Das implizite Home-Interface ... 192
9.3.2 Das explizite Home-Interface ... 195
9.4 Typumwandlung für die Beispielszenarien ... 198
9.4.1 Hello-World ... 199
9.4.2 Philosophen ... 200

10 Design der Implementierung ... 203

10.1 Entwurf der Komposition ... 203
10.2 Speichertypen ... 204
10.2.1 Speicherschnittstellen ... 205
10.2.2 Speicherfabrikschnittstellen ... 206
10.2.3 Speicherobjekttypen ... 208
10.2.4 Speicherfabriktypen ... 209
10.3 Komposition ... 210
10.3.1 CIDL ... 210
10.3.2 UML ... 211
10.4 Home Executor ... 212
10.4.1 CIDL ... 213
10.4.2 UML ... 214
10.5 Proxy Home Executor ... 215
10.5.1 CIDL ... 216
10.6 Komponenten-Executor ... 216
10.6.1 CIDL ... 217

10.6.2 UML 218
10.7 Definition der Implementierungsstruktur der Beispiele 219
10.7.1 Hello World 219
10.7.2 Philosophen 221

11 Implementierung 225

11.1 Der Sinn der Sprachabbildung 225
11.2 Grundlegende Strategie 227
11.3 Basis einer Komponentenimplementierung 228
11.3.1 Der Executor Locator 229
11.3.2 Fehlerzustände im Executor 230
11.4 Interfaces für den Session-Containertyp 231
11.5 Interfaces für den Entity-Containertyp 238
11.6 Typspezifische Implementierungsregeln 243
11.6.1 Regeln für Interfaces 244
11.6.2 Regeln für den Komponententyp 245
11.6.3 Regeln für Ereignistypen 247
11.6.4 Facets 248
11.6.5 Receptacles 250
11.6.6 Ereignisproduzenten 251
11.6.7 Ereigniskonsument 252
11.6.8 Homes 253
11.7 Abbildung der CIDL-Konstrukte 255
11.7.1 Abbildung der Modulnamen 255
11.7.2 Abbildung des Kompositionsnamens 256
11.7.3 Abbildung des Hauptsegments 258
11.7.4 Abbildung für weitere Segmente 260
11.7.5 Abbildung für den Home-Executor 263
11.8 Implementierungsaspekte beim locator-basierten Ansatz 265
11.9 Bemerkungen 265

12 Bereitstellung von Komponenten 267

12.1 Komponentenbeschreibung 267
12.1.1 Dateikopf 268
12.1.2 Allgemeine Informationen 269
12.1.3 Transaktionen, Sicherheit und Persistenz 270
12.1.4 Eigenschaften des Home 270
12.1.5 Eigenschaften der Komponente 270
12.2 Implementierungsbeschreibung 272
12.2.1 Dateikopf 272
12.2.2 Allgemeine Informationen 273

12.2.3 Implementierungsvarianten 275
12.3 Komponentenarchive 279
12.3.1 Beispiel 279

Teil III – Entwicklung von Anwendungen

13 Entwurf und Spezifikation von Anwendungen 283

13.1 Entwurf von Anwendungen 284
13.1.1 Beispiel 285
13.2 Spezifikation von Anwendungen 286
13.2.1 Umfang der Spezifikation 287
13.2.2 Spezifikation mit XML 288
13.2.3 Anwendungsbeschreibung 289
13.2.4 Konfigurationsbeschreibung 302
13.3 Ergebnis 305

14 Implementierung und Bereitstellung von Anwendungen 307

14.1 Implementierung von Anwendungen 308
14.1.1 Vorgehensweise 309
14.1.2 Beispiel 310
14.2 Bereitstellung von Anwendungen 312
14.2.1 Anwendungsarchive 312
14.2.2 Beispiel 313

15 Infrastruktur für die Inbetriebnahme 315

15.1 Installation von Komponenten 315
15.1.1 Bemerkungen 320
15.2 Erzeugung von Laufzeitinstanzen 321
15.2.1 Instantiierung eines Servers 322
15.2.2 Das Interface des Komponentenservers 324
15.2.3 Das Container-Interface 326
15.2.4 Einsprungpunkte 329
15.3 Laufzeitrepräsentationen von Anwendungen 331
15.3.1 Erzeugung von Laufzeitrepräsentationen 333
15.3.2 Bemerkungen 336
15.4 Sichtbarkeit von Laufzeitinstanzen 336
15.4.1 Durchsuchen der Plattform 338
15.4.2 Verwendung von Standarddiensten 339
15.4.3 Home Finder 340

16 Inbetriebnahme von Anwendungen 345

16.1 Bestimmung der Verteilung 346
16.1.1 Beispiel 347
16.2 Übergabe an die Plattform 350
16.2.1 Erzeugung der Laufzeitrepräsentation der Anwendung 350
16.2.2 Installation 351
16.2.3 Instantiierung 352
16.2.4 Konfiguration 355

Teil IV – Entwickeln mit Qedo

17 Die Software Qedo 361

17.1 Geschichte von Qedo 361
17.2 Eigenschaften von Qedo 363
17.2.1 Allgemeine Eigenschaften 363
17.2.2 Eigenschaften der Laufzeitumgebung 365
17.2.3 Eigenschaften der Werzeuge 367

18 Installation von Qedo 371

18.1 Die aktuellen Quellen 371
18.1.1 CVS-Benutzung 372
18.1.2 Web-Benutzung 373
18.1.3 FTP-Benutzung 374
18.2 Kompatible ORB-Produkte 374
18.3 Weitere benötigte Software 375
18.3.1 XML-Unterstützung 375
18.3.2 Archivierungsunterstützung 376
18.3.3 Softwarepakete für den Generator 376
18.4 Installationsprozess 377
18.4.1 Installation unter Windows 378
18.4.2 Installation unter Linux 381
18.5 Inbetriebnahme von Qedo 382

19 Ein Beispiel mit Qedo 385

19.1 Die Benutzung des Generatorwerkzeugs 385
19.2 Die Spezifikation der Komponente 388
19.3 Die Implementierung der Komponente 390
19.3.1 Die Beschreibung mit CIDL 390
19.3.2 Implementierung mit Visual Studio .NET 393

19.4 Die Erstellung des Komponentenarchivs 404
19.4.1 Die Komponentenbeschreibung 404
19.4.2 Die Implementierungsbeschreibung..... 405
19.5 Fertigstellung des Komponentenarchivs 406
19.6 Ausblick 407

Glossar 409

Literatur 417

Sachverzeichnis 419