

Inhaltsverzeichnis

Vorwort	ix
Einleitung	xi
Teil 1 Motivation, Grundlagen, Beispielanwendung	1
1 Motivation für komponentenbasierte Softwareentwicklung	3
1.1 Flexible Verteilung und Anwendungsnahe	5
1.2 Die Legosteine-Metapher	13
1.3 Migration zu komponentenbasierten Softwaresystemen	15
1.4 Grundbegriffe der komponentenbasierten Softwareentwicklung	15
2 Migration zu komponentenbasierten Anwendungsarchitekturen	25
2.1 Ist-Analyse	27
2.2 Definition der fachlichen Architektur	30
2.3 Definition der softwaretechnischen Architektur	31
2.4 Definition der systemtechnischen Architektur	33
2.5 Anpassung des Entwicklungsprozesses	33
2.6 Definition der organisatorischen Konsequenzen	36
2.7 Pilotprojekte	36
3 Vorstellung der Beispielanwendung	39
3.1 Motivation	39
3.2 Anforderungen, Use Cases	40
3.2.1 Beschreibung der Use Cases	40
3.2.2 Priorisierung und Auswahl der Use Cases	42
3.3 Analysemodell	42
Teil 2 Die Umsetzung der Szenarien	45
4 DCOM-Szenario	47
4.1 COM-Übersicht	47
4.1.1 Historie von COM	47
4.1.2 Was ist COM?	51
4.1.3 Schnittstellen	52

4.1.4	Referenzzählung	56
4.1.5	Erzeugen von Komponenteninstanzen	58
4.1.6	Weltweit eindeutige Bezeichner – GUID	61
4.1.7	IDL	61
4.1.8	Automation und die IDispatch-Schnittstelle	64
4.1.9	Komponentenkategorien	66
4.1.10	Versionierung	67
4.1.11	Containment und Aggregation	68
4.1.12	Komponenten-Server	71
4.1.13	Nebenläufigkeit	75
4.1.14	Persistenz	77
4.1.15	Monikers	80
4.1.16	Verbindungspunkte	82
4.1.17	ATL – Die ActiveX Template Library	86
4.2	Übergang zu DCOM	87
4.2.1	Erzeugen von entfernten Komponenteninstanzen	88
4.2.2	Referenzzählung bei DCOM	90
4.2.3	Sicherheit	91
4.2.4	Konfiguration	92
4.3	Anwendung auf das Beispiel	94
4.3.1	Eingesetzte Bibliotheken und Werkzeuge, Umgebung	94
4.3.2	Architektur	95
4.3.3	Design und Implementierung	96
5	JavaBeans/CORBA-Szenario	143
5.1	JavaBeans-Übersicht	143
5.1.1	Struktur einer JavaBeans-Komponente	144
5.1.2	Ereignisse	145
5.1.3	Eigenschaften	149
5.1.4	Nebenläufigkeit	153
5.1.5	Metainformationen	154
5.1.6	Persistenz	157
5.1.7	Unterstützung visueller Programmierung	160
5.1.8	Weitergabe von JavaBeans	161
5.1.9	Laufzeitunterstützung	163

5.2	CORBA-Übersicht	164
5.2.1	Der Object Request Broker (ORB)	164
5.2.2	OMG-IDL	168
5.2.3	Implementierung von CORBA-Objekten	169
5.3	Anwendung auf das Beispiel	171
5.3.1	Eingesetzte Bibliotheken und Werkzeuge, Umgebung	171
5.3.2	Architektur	172
5.3.3	Design und Implementierung	176
6	JavaBeans/Enterprise-JavaBeans-Szenario	209
6.1	Enterprise-JavaBeans-Übersicht	209
6.1.1	Ziele von EJB	210
6.1.2	EJB-Architektur	211
6.1.3	Entwicklung einer Enterprise Bean	213
6.1.4	Session Beans	217
6.1.5	Entity Beans	218
6.1.6	Transaktionen	219
6.1.7	Sicherheit	221
6.1.8	Einsatz von Enterprise Beans	222
6.2	Anwendung auf das Beispiel	222
6.2.1	Eingesetzte Bibliotheken und Werkzeuge, Umgebung	222
6.2.2	Architektur	223
6.2.3	Design und Implementierung	225
Teil 3	Vergleich und Ergebnisse	255
7	Zusammenfassender Vergleich	257
7.1	Auswahl der Vergleichskriterien	257
7.2	Vergleich	260
7.2.1	Standardisierung	260
7.2.2	Plattformunabhängigkeit	261
7.2.3	Binärstandard	262
7.2.4	Kapselung	263
7.2.5	Identität	263
7.2.6	Versionierung	264
7.2.7	Skalierbarkeit	265

7.2.8	Beschreibung von Schnittstellen	266
7.2.9	Eigenschaften einer Komponente	266
7.2.10	Ereignismodell	267
7.2.11	Datenaustausch	268
7.2.12	Kontextbeschreibung	269
7.2.13	Kompositionsmöglichkeiten	269
7.2.14	Unterstützung visueller Programmierung	270
7.2.15	Komponentenstruktur	271
7.2.16	Persistenz	272
7.2.17	Transaktionen	273
7.2.18	Sicherheit	274
7.2.19	Nebenläufigkeit	275
7.2.20	Speicherverwaltung	275
7.2.21	Fehlerbehandlung	276
7.2.22	Frameworks	277
7.2.23	Metadaten	278
7.2.24	Dynamisches Laden	278
7.2.25	Polymorphie	279
7.2.26	Skriptsprachen	280
7.2.27	Installation	280
7.2.28	Aktivierung	281
7.2.29	Verbreitung	282
7.3	Erfahrungen	283
	EBNF der Projektimportdateien	289
	Glossar	291
	Abkürzungsverzeichnis	295
	Literaturverzeichnis	299
	Stichwortverzeichnis	305