

Inhalt

1 Einführung und Grundlagen	1
1.1 Einsatz verteilter DV-Systeme: Motivation und Entwicklungstendenzen	1
1.1.1 Generationswechsel von Rechnersystemen	1
1.1.2 Divergierende Wachstumsprofile	3
1.1.3 Wichtigste Folgen dieser Entwicklung	6
1.2 Verteilte Systeme und verteilte Anwendungen	7
1.2.1 Übersicht	7
1.2.2 Gegenstand der Verteilung	9
1.2.3 Zielsetzungen	10
1.2.4 Eigenschaften und Probleme	11
1.3 Beispiel einer verteilten Anwendung	13
2 Existierende Ansätze zur verteilten Programmierung	15
2.1 Betriebssystem-Ansatz	16
2.2 Datenbank-Ansatz	18
2.3 Sprachintegrierter Ansatz: Verteilte Programme	19
2.3.1 Verteilte Programmiersprachen	20
2.3.2 Softwareproduktionsumgebungen	22
2.4 Funktionaler Ansatz	25
2.5 Vergleichende Bewertung	27
3 Betriebssystemansatz	29
3.1 Netzbetriebssysteme und verteilte Betriebssysteme	29
3.2 Klienten-Server-Modell	31
3.3 Nameserver	31
3.4 Dateiserver	32

3.5	Entfernte Programmausführung	34
3.6	Prozeßmigration	35
3.7	Systembeispiel: Das Netzbetriebssystem DACNOS	37
3.7.1	Systemarchitektur	38
3.7.2	Das Kernsystem RSC	41
3.7.3	Der Task Setup Service	49
4	Sprachintegrierter Ansatz	55
4.1	Nachrichtenaustausch mit Sprachintegration	55
4.2	Remote Procedure Call: Überblick	61
4.3	Verteilte objektorientierte Ansätze: Überblick	62
4.4	Verteilte Konfigurationsverwaltung: Überblick	63
4.5	Verteilte Programmiersprachen	64
5	Remote Procedure Call	71
5.1	Grundlagen des RPC	72
5.1.1	Begriffsdefinition	72
5.1.2	Grundlegender Ablauf eines RPC	73
5.1.3	Abgrenzung zum einfachen Nachrichtenaustausch	75
5.2	Der Bindevorgang beim RPC	76
5.2.1	Beschreibung von RPC-Schnittstellen	76
5.2.2	Lokalisierung	81
5.2.3	Optimierung des Bindevorgangs	84
5.3	Abwicklung von RPCs zur Laufzeit	85
5.3.1	Kodierung von Übertragungsdaten	85
5.3.2	Zugrundeliegende Transportmechanismen	86
5.3.3	Prozeßverwaltung bei Klient und Server	90
5.4	Fehlersemantik des RPC	92
5.4.1	Fehlerursachen beim RPC	92
5.4.2	Fehlersemantik-Klassen	94
5.5	Beispiele für existierende RPC-Systeme	96

5.6	Probleme mit herkömmlichen RPC-Protokollen	98
5.7	Fehlerbehandlung in RPC-Systemen	101
5.7.1	Fehlerbenachrichtigung	102
5.7.2	Behandlung verwaister Aufrufe	103
5.7.3	Fehlertoleranz durch replizierte Aufrufe	105
5.7.4	Atomarität	110
5.8	Asynchrone RPCs und Massendatentransfer	115
5.8.1	RPC und Massendatentransfer: Vergleich	115
5.8.2	Integration der beiden Mechanismen	117
5.8.3	Asynchrone RPCs mit Ergebnismrückgabe	119
5.9	Sicherheitsaspekte des RPC	123
5.9.1	Mögliche Beeinträchtigungen der Sicherheit	123
5.9.2	Verschlüsselung	124
5.9.3	Identifizierung und Authentisierung	127
5.9.4	Mögliche Garantien und Kosten	128
5.10	RPC in offenen verteilten Systemen	130
5.10.1	RPC über Weitverkehrsnetze	130
5.10.2	Heterogene RPC-Systeme und Implementierungen... ..	133
5.11	Standardisierungsbestrebungen im Bereich des RPC	137
5.11.1	Der RPC der Open Software Foundation	137
5.11.2	Der RPC der ECMA und der ISO	139
6	Verteilte objektorientierte Ansätze	141
6.1	Grundlagen des objektorientierten Ansatzes	141
6.2	Zentrale Begriffsdefinitionen	142
6.3	Spezielle Eigenschaften	144
6.4	Existierende Sprachen und Systeme	146
6.5	Verteilte Erweiterungen	147
6.5.1	Zentrale Eigenschaften	148
6.5.2	Resultierende Vorteile	149
6.5.3	Anwendungsbeispiel	151
6.5.4	Existierende Systeme	154

6.6	Objektmobilität	157
6.6.1	Grundbegriffe	157
6.6.2	Realisierung von Objektmigrationen	158
6.7	Beispiel eines verteilten objektorientierten Systems	162
6.7.1	Prozesse in Emerald	165
6.7.2	Objektmobilität	166
6.7.3	Objektlokalisierung	169
6.7.4	Implementierung von Objektmigrationen	170
6.8	Programmiersprachliche Anwendung	173
6.9	Weitere Aspekte verteilter objektorientierter Ansätze	177
7	Konfigurationsverwaltung in verteilten Systemen	181
7.1	Motivation	181
7.2	Vorteile einer Konfigurationsverwaltung	183
7.3	Modell einer Konfigurationsverwaltung	185
7.4	Beispiele existierender Konfigurationsverwaltungen	187
7.5	Erforderliche Systemeigenschaften	196
7.5.1	Eigenschaften des zugrundeliegenden Systems	196
7.5.2	Eigenschaften der Konfigurationsverwaltung	199
8	Softwareproduktionsumgebungen	205
8.1	Einführung	206
8.1.1	Terminologie	206
8.1.2	Motivation	207
8.1.3	Bedeutung des Projektmanagements	209
8.2	Taxonomie	210
8.2.1	Sequentielle Ein-Entwickler SPUs	210
8.2.2	Sequentielle Mehr-Entwickler SPUs	211
8.2.3	Cross-Entwicklungsumgebungen	212
8.2.4	Verteilte SPUs	213
8.2.5	SPUs für verteilte Anwendungen	216
8.2.6	CASE: Computer Aided Software Engineering	217

8.3	Lifecycle-Modelle	218
8.3.1	Vorüberlegungen	219
8.3.2	Das Wasserfall-Modell	220
8.3.3	Ein Werkzeugmodell	223
8.3.4	Neuere Lifecycle-Modelle	227
8.4	SPU-Architekturen	230
8.4.1	Abstrakte Grobarchitektur	230
8.4.2	Interaktionsschicht	231
8.4.3	Datenschicht	233
8.4.4	Funktionsschicht - Shell	234
8.4.5	Funktionsschicht - Kern-Werkzeuge	235
8.4.6	Funktionsschicht - Entwicklungsaspekte	237
8.4.7	Funktionsschicht - Management-Aspekte	239
9	Methoden und Werkzeuge	241
9.1	Anforderungsanalyse	242
9.2	Spezifikation	244
9.2.1	Abgrenzung und Vergleich mit der Entwurfsphase	244
9.2.2	Spezifikation und verteilte Anwendungen	246
9.2.3	Spezifikationstechniken: Modellbildung	248
9.2.4	Kontrollschritte	259
9.2.5	Beispiel: Spezifikation vs. Intention	263
9.2.6	Beispiel: Spezifikation von Prozeßnetzen	266
9.3	Entwurf	270
9.3.1	Begriffsbildung	271
9.3.2	Sichten	272
9.3.3	Allgemeine Anforderungen	275
9.3.4	Anforderungen für verteilte Anwendungen	276
9.3.5	Beispiel: Issue-based Design	278
9.3.6	Beispiel: Object-oriented Structured Design	280
9.3.7	Beispiel: Higraphs und Statecharts	284
9.3.8	Zu weiteren Entwurfsmethoden	289
9.4	Beschreibung von Gruppenkommunikation	290
9.4.1	Script-Ansatz	291
9.4.2	Raddle	297

9.5	Verteiltes Debugging	298
9.5.1	Begriffsbestimmung und allgemeine Anforderungen ..	299
9.5.2	Spezielle Probleme verteilter Testhilfen	302
9.5.3	Ansätze für verteilte Testhilfen	307
9.5.4	Implementierungsfragen	320
9.6	Laufzeit-Codeverwaltung	323
9.6.1	Einleitung und Überblick	324
9.6.2	Übersetzungsverwaltung	326
9.6.3	Die Experimentdefinition	330
9.6.4	Die Experimentverteilung	333
10	Ausblick: Aktuelle Entwicklungen und Standardisierung	337
10.1	Hochgeschwindigkeitsnetze	337
10.1.1	Der FDDI Token Ring	338
10.1.2	Der DQDB-Ansatz	339
10.1.3	Breitband-ISDN	340
10.2	Perspektiven künftiger verteilter Anwendungen	341
10.2.1	Technologische Entwicklungstendenzen	341
10.2.2	Implikationen für verteilte Anwendungen	343
10.2.3	Software-Engineering	345
10.3	Standardisierungsansätze	350
10.3.1	ISO/OSI-Darstellungs- und Anwendungsschicht	350
10.3.2	Das OSF Distributed Computing Environment	353
10.3.3	Open Distributed Processing	354
10.3.4	Netzwerkmanagement	356
10.4	Zusammenfassung und Ausblick	362
	Literatur	365
	Index	385