

Inhaltsverzeichnis

1 Einführung	1
1.1 Einleitung	1
1.2 Beispiel einer verteilten Anwendung	1
1.3 Vorteile verteilter Anwendungen	3
1.4 Resultierende Probleme und Anforderungen	4
1.5 Problemstellung: Systemunterstützung	5
2 Client/Server-Ansatz und verteilte Objekte: Systemmodelle im Vergleich	7
2.1 Client/Server-Modell	7
2.1.1 Grundstruktur	7
2.1.2 Charakteristische Eigenschaften	8
2.1.3 Bewertung	9
2.2 Verteiltes objektorientiertes Modell	10
2.2.1 Grundstruktur	10
2.2.2 Charakteristische Eigenschaften	12
2.2.3 Bewertung	12
2.3 Abschließender Vergleich	13
3 Remote Procedure Call	15
3.1 Grundlagen	15
3.1.1 Übersicht und Definition	15
3.1.2 Ablauf	16
3.2 Schnittstellenbeschreibung und Laufzeitunterstützung	17
3.2.1 Schnittstellenbeschreibung	18
3.2.2 Binden von Client und Server	21
3.2.3 Basiskommunikation	23
3.2.4 Prozeßverwaltung	24
3.2.5 Fehlersemantik	25
3.3 Vergleich existierender Systeme und Standards	27
3.3.1 Systemvergleich	27
3.3.2 Leistungsmessungen zum DCE RPC	28
3.3.3 Standardisierungsbestrebungen der ISO und ECMA	29
3.3.4 Kritik des RPC	30
3.4 Erweiterte Kommunikationsstrukturen	31
3.4.1 Effizienzverbesserung durch asynchrone RPCs	32
3.4.2 RPC und Massendatentransfer	34
3.4.3 Effiziente RPCs zwischen Adreßräumen eines Rechners	36
3.4.4 Strukturelle Verbesserungen durch rückwärtige Aufrufe	37
3.4.5 Flexibilität durch dynamische Server-Codeinstallation	38

3.5	Mechanismen der Fehlerbehandlung in RPC-Systemen	39
3.5.1	Integration von RPCs und Transaktionskonzepten	39
3.5.2	Replizierte RPCs	43
3.5.3	Behandlung verwaister Aufrufe	44
3.6	RPC in offenen verteilten Systemen	46
3.6.1	Sicherheitsaspekte des RPC	46
3.6.2	Behandlung der Heterogenitätsproblematik	50
3.7	Unterstützung durch verteilte Namensverwaltung	52
3.7.1	Kontexte und Namensinterpretation	53
3.7.2	Implementierung verteilter Directory Services	55
3.7.3	Standards	56
3.7.4	Systemvergleich	57
3.8	Zusammenfassung	60
4	Verteiltes objektorientiertes Modell	61
4.1	Objektorientierte Programmierung	61
4.1.1	Charakteristische Eigenschaften	61
4.1.2	Nebenläufigkeit	63
4.1.3	Verteiltes Objektmodell	67
4.2	Entfernte Kommunikation zwischen Objekten	69
4.2.1	Grundlagen	69
4.2.2	Lokalisierung fixierter Objekte	69
4.2.3	Lokalisierung mobiler Objekte	71
4.2.4	Lokalisierung mittels Vorwärtsadressierung	73
4.2.5	Parameterübergabe bei entfernten Aufrufen	76
4.3	Objektmigration	77
4.3.1	Steuerung von Migrationen durch die Anwendung	78
4.3.2	Realisierung von Objektmigrationen	80
4.3.3	Behandlung von Kontrollflüssen	83
4.3.4	Migration bei heterogenen Klassenstrukturen	85
4.4	Teilautomatische Steuerung von Migrationen	88
4.4.1	Kollokationsdefinitionen und Kollokationen	89
4.4.2	Laufzeitprotokoll zur Initiierung von Kollokationen	90
4.4.3	Simulative Analyse des Laufzeitprotokolls	91
4.5	Automatische Speicherverwaltung	93
4.5.1	Referenzzähler	93
4.5.2	Markierungsverfahren	94
4.5.3	Verteilte Realisierung von Markierungsverfahren	95
4.6	Systemrealisierung: DC++	96
4.6.1	Systemimplementierung	96
4.6.2	Leistungsanalyse und Erfahrungen	99
4.7	Systemvergleich	100
4.7.1	Überblick über die einzelnen Systeme	101
4.7.2	Abgrenzung des eigenen Ansatzes	103

4.8	Weiterführende Konzepte	104
4.8.1	Integration mit persistenten Objektmodellen	104
4.8.2	Replizierte verteilte Objekte	106
4.8.3	Duale Objekte: Temporäre Replikation bei Stellvertretern	107
4.8.4	Distributed Shared Memory als Realisierungsbasis	109
4.8.5	Objekte in kooperativen verteilten Anwendungen	110
4.9	Zusammenfassung	111
5	Verteilte Büroabläufe als Anwendungsgebiet	112
5.1	Einführung und Anforderungen	112
5.2	Objektorientierte Modellierung	115
5.2.1	Repräsentation der Systemeinheiten durch verteilte Objekte	115
5.2.2	Vergleich mit dem Client/Server-Modell	117
5.3	Sprachunterstützung	118
5.3.1	Beschreibung von Büroabläufen	118
5.3.2	Beschreibung der Basisdienste und der Server	120
5.4	Laufzeitunterstützung	122
5.4.1	Systemarchitektur	123
5.4.2	Ablaufphasen	123
5.4.3	Ablaufüberwachung	127
5.5	Vergleich existierender Systeme	128
5.5.1	Überblick über die einzelnen Systeme	128
5.5.2	Abgrenzung des eigenen Ansatzes	131
5.5.3	Systemvergleich	131
5.6	Erweiterung: Dynamische Rekonfiguration der Serverstruktur im Fehlerfall	132
5.6.1	Grundkonzept und Basisalgorithmus	133
5.6.2	Beispiel	134
5.6.3	Weiterentwicklungen	135
5.7	Zusammenfassung	135
6	Bezug zur aktuellen Standardisierung	137
6.1	Das OSF Distributed Computing Environment	137
6.2	Anwendungsorientierte ISO/OSI-Standards	139
6.3	Object Management Architecture	142
6.4	Open Distributed Processing	145
6.5	Bewertung und Kritik	146
6.6	Zusammenfassung	147
7	Zusammenfassung und Ausblick	148
7.1	Ergebnisse der Arbeit	148
7.2	Ausblick auf weiterführende Projekte	149
8	Literaturverzeichnis	150