

Dipl.-Inform. Thomas Wolff, Berlin

Transparente Verteilung objektorientierter Programme

Reihe **10**: Informatik/
Kommunikationstechnik Nr. **457**

Inhalt

1	Übersicht	I
1.1	Einleitung	1
1.1.1	Verteilung und Nebenläufigkeit	1
1.1.2	Verteilung und Objektorientierung	1
1.1.3	Transparenz als Prinzip	2
1.1.4	Projektziele im Vergleich mit anderen Verteilungssystemen	4
1.1.5	Programmiersprache	7
1.1.6	Implementierung der Verteilungstechnik	7
1.1.7	Vorgehensweise beim Entwurf	7
1.2	Inhalt und Darbietung	8
1.2.1	Hervorhebungen	8
1.2.2	Textlicher Ausdruck	9
2	Vorüberlegungen zur Verteilungstechnik	10
2.1	Fernbeziehungen und Transparenz	10
2.2	Wann kann Objektfernerzeugung transparent sein?	11
2.3	Wie kann Objektfernbenutzung transparent sein?	11
2.4	Transparenz der Parameterübergabesemantik	12
2.4.1	Fortsetzung des Stellvertreterprinzips durch Parameterübergabe	13
2.5	Transparenz, Effizienz und Sprachstruktur	13
2.6	Implementierungstechniken	14
3	Erste Entwicklung der Stellvertreterobjekte	15
3.0	Terminologie	15
3.1	Verteilungsbedingter Informationsfluß	16
3.2	Verteilter Informationstransport und die Rolle von RPC	17
3.3	Aufbau der Stellvertreterobjekte	17
3.3.1	Aufbau der Objektvertreter-Klassen	18
3.3.2	Vorgang des Fernaufrufs bzw. der Fernerzeugung	22
3.3.3	Aufbau der Objekthalter-Klassen	25
4	Verteilung und Vererbung	31
4.1	Vererbung und Verteilung	31
4.1.1	Isomorphe Namensbindung	32
4.1.2	Auffinden umbenannter und undefinierter Operationen	32
4.1.3	Alternativen zur Erbschaftsisomorphie	34
4.2	Lokaler Weg des Aufrufs in das Originalobjekt	37
4.2.1	Einstieg in den Objekthalter	37
4.2.2	Auffinden der korrespondierenden Operationen im Objekthalter	39
4.2.3	Aufruf des originalen Objekts	45
5	Spezielle Aspekte der Verteilung	47
5.1	Parameterübergabe bei Mehrfachverteilung	47
5.2	Behandlung lokaler bzw. entfernter Parameter	48
5.3	Nullfachverteilung	49
5.3.1	Realisierung	50
5.3.2	Alternativen	50
5.3.3	Namenskonflikt	51
5.3.4	Selbstbezug in bilokalen Klassen	52
5.3.5	Objektmobilität (Migration)	53
5.4	Verteilungsstruktur	54
5.4.1	Verteilungsstruktur und lokale Klassen	56

5.4.2	Verteilungsstruktur bei verteilter Erbschaft	58
5.4.3	Konfiguration der Verteilungsstruktur	58
5.4.4	Verteilungsstruktur und Parameterübergabe	59
5.5	Objektidentität	60
5.5.1	Ein weiteres Problem mit gleicher Ursache und Konsequenz	62
5.5.2	Diskussion eines alternativen Lösungsansatzes	62
5.6	Aufgeschobene Erzeugung („deferred creation“)	64
5.6.1	Nicht-Transparenz des Objekterzeugungszeitpunkts	68
5.7	Umgang mit Objektwerten	69
5.7.1	Struktur der Objektpacker	71
5.7.2	Mehrfaches Kopieren von Parameterobjekten?	72
5.7.3	Weitere Probleme mit Objektwerten	73
5.8	Remote Object Handling (ROH)	75
5.8.1	Fernkopie und Fernduplizierung	75
5.8.2	Fernvergleich	79
5.8.3	Realisierung der ROH-Implementierung	79
5.9	Verteilte Garbage Collection	80
5.10	Remote Exception Propagation	81
5.11	Generische Klassen	82
5.11.1	Einfacher Ansatz	82
5.11.2	Aspekte der transparenten Lösung	85
5.11.3	Implementierungsaspekte einer transparenten Lösung	87
5.12	Globale Objekte	88
5.13	Datenstrukturen	91
5.14	Öffentliche Server-Kompostanzen	92
5.14.1	Nebenläufigkeitsaspekt öffentlicher Kompostanzen	93
5.14.2	Gemeinsamer Zugriff in öffentliche Kompostanzen	93
5.14.3	Verschiedene Klassensysteme mit öffentlichen Kompostanzen	94
5.14.4	Zugriffsschutz für Schnittstellen und Objekte	96
6	Verteilte Erbschaft	97
6.1	Diskussion einer Alternativstruktur	99
6.2	Verteilungsstruktur zerteilter Objekte	99
6.3	Vertreter für entfernte zerteilte Objekte	100
6.4	Erzeugung zerteilter Objekte	100
6.5	Implementierung der Segmente	103
7	Vorgehensweisen bei der Implementierung	105
7.1	Unterscheidung lokaler und entfernter Objekte	105
7.2	Unterscheidung lokaler und entfernter Objekterzeugung	105
7.3	Nebenläufige Objekterzeugung	107
7.4	Geerbte Erzeugungsoperationen	107
7.5	Selektive Zugriffsrechte	108
8	Abschließende Gesamtübersicht der Stellvertreter	109
8.1	Zusammenfassung der strukturbestimmenden Regeln	109
8.2	Inhalt der Objektvertreter	112
8.3	Klassenstruktur des Verteilungssystems	125
9	Systembeschränkungen	127
9.1	Transparenzdefizite	127
9.2	Einschränkungen	128
9.3	Implementierungsbeschränkungen	128

10	Objektorientiertes Programmiermodell und Verteilung	130
11	Einbettung des Verteilungssystems	133
11.1	Konfigurierung der Objekterzeugung	133
11.1.1	Realisierung der Residenzbestimmungsfunktion	134
11.1.2	Nicht-transparente Residenzbestimmung	135
11.1.3	ROC im Systemvergleich	137
11.2	Schnittstelle zum Laufzeitsystem	137
11.2.1	Handhabung und Aufbau der Botschaften	138
11.2.2	Kontrollfluß zwischen Botschaften und Operationsaufrufen	140
11.2.3	Kompostanzhandhabung	141
11.2.4	Wahl eines Kommunikationssystems	143
11.2.5	Eingesetzte Funktionalität des Kommunikationssystems PVM	144
12	Realisierung des Vertretergenerators	147
12.1	Parser	147
12.2	Generator	147
12.2.1	Benutzung des Generators	148
13	Heterogene Erweiterung	149
14	Abschließende Bemerkungen	151
14.1	Zusammenfassung wesentlicher Aspekte und Resultate	151
14.2	Gesamtwürdigung des Systems und Zukunftsaspekte	154
14.3	Danksagungen	155
14.4	Erklärung	156
ANHANG 1	Elemente von Eiffel	157
ANHANG 2	Die Klassen des Laufzeitsystems	160
ANHANG 2.1	Die variablen Klassen der Kompilenten	160
ANHANG 2.2	Die festen Klassen des Verteilungssystems	162
ANHANG 2.3	Die Kommunikationskomponenten des Laufzeitsystems	169
ANHANG 3	Einrichtung des LZS und eines verteilten Programms	187
ANHANG 4	Zusammenfassung	190
	Index	191
	Regeln	194
	Referenzen	195