

Inhaltsverzeichnis

1	Einleitung	5
1.1	Konfiguration verteilter komponentenbasierter Anwendungen	5
1.2	Eine Ausführungsumgebung für adaptive Anwendungen	8
1.3	Dynamische Rekonfiguration	9
1.4	Funktionsweise adaptiver Anwendungen	11
1.5	Struktur der Arbeit	12
1.6	Beitrag der Arbeit	13
1.7	Hinweise zur Schreibweise	15
2	Terminologie und Grundlagen	17
2.1	Terminologie	17
2.2	Auf dem Weg zu adaptiver Software	18
2.2.1	Allgegenwärtige Rechner	20
2.2.2	Kontextsensitive Systeme	21
2.2.3	Autonomic Computing	22
2.2.4	Organic Computing	23
2.3	Komponentenplattformen für adaptive Anwendungen	24
2.3.1	OMG CORBA und CCM	24
2.3.2	Sun Java und Enterprise JavaBeans	24
2.3.3	Microsoft DCOM und .NET	25
2.3.4	Komponentenplattformen für mobile Geräte	26
2.4	Basistechnologien für adaptive Anwendungen	27
2.4.1	Komponentenorientierung	27
2.4.2	Metaprogrammierung und Reflexion	27
2.4.3	Dynamisches Laden von Softwarekomponenten	28
2.4.4	Aspektorientierte Programmierung	29
2.5	Virtuelle Laufzeitumgebungen	31
2.6	Zusammenfassung	32
3	Dynamische Rekonfiguration	33
3.1	Ein Modell verteilter, komponentenbasierter Anwendungen	33
3.1.1	Die Einheiten des Deployment - Assemblies	35
3.1.2	Die Komponenten zur Laufzeit - Kapseln	36
3.1.3	Die Anwendungskonfiguration	40
3.1.4	Ausführungsmodell	41
3.2	Anforderungen an die dynamische Rekonfiguration	42
3.2.1	Erhaltung von Konsistenz	42
3.3	Dynamische Rekonfiguration in einer Komponentenplattform	46
3.3.1	Rekonfiguration mit azyklischen Verbindungsgraphen	48
3.3.2	Rekonfiguration mit zyklischen Verbindungsgraphen - ReDAC	50

3.3.3	Eigenschaften von ReDAC	54
3.3.4	Optimierung bei bekannten BlockSets	56
3.3.5	Erweiterung bei Synchronisation zwischen Anwendungs-Threads	56
3.3.6	Erweiterungen	57
3.4	Konfiguration und ihre Beschreibung	58
3.4.1	Die Konfigurationsspezifikation	58
3.4.2	Beschreibung von Konfigurationsänderungen	58
3.5	Eine Ausführungsplattform für adaptive Anwendungen	61
3.5.1	Der Konfigurationsmanager	62
3.5.2	Deployment	64
3.5.3	Konfigurierbare Kapseln	66
3.5.4	Der Konfigurationsaspekt	68
3.5.5	Der Kapselkonfigurator	73
3.6	Erreichen eines rekonfigurierbaren Zustands	76
3.6.1	Synchronisationspunkte und Programmierschnittstellen	76
3.6.2	Synchronisation bei azyklischen Verbindungsgraphen	78
3.6.3	Effizientes Blockieren von Konnektoren	79
3.6.4	Synchronisation bei zyklischen Verbindungsgraphen	82
3.6.5	Implementierung logischer Thread-IDs	85
3.7	Migration	86
3.7.1	Migration von Kapseln mit eigenen Threads	87
3.8	Dynamische Integration von Diensten	87
3.9	Zusammenfassung	87
4	Dynamische Aktualisierung - Spezialfall der dynamischen Rekonfiguration	89
4.1	Modellerweiterung	90
4.2	Implementierung	94
4.2.1	Auslösen dynamischer Aktualisierungen	94
4.2.2	Die Traversierungsphase	95
4.2.3	Vorabgenerierung der Aspektlogik	99
4.2.4	Typspezifische Zustandsanpassung	100
4.2.5	Anpassung graphischer Nutzerschnittstellen	102
4.2.6	Dynamische Aspekte	105
4.3	Einschränkungen	106
4.4	Zusammenfassung	106
5	Profile, Werkzeuge und Muster	109
5.1	Auswahl geeigneter Anwendungskonfigurationen	109
5.1.1	Dienstgüte und Ressourcen	109
5.1.2	Dienstgüte und Ressourcen im Modell adaptiver Anwendungen	110
5.1.3	Adaption im Modell	111
5.2	Profilbasierte Konfigurationszuordnung	113
5.2.1	Reglerbasierte Parameteradaption	113
5.3	Der Adaptionsmanager	114
5.3.1	Monitoring	115
5.3.2	Definition von Profilen	116
5.3.3	Auswahl der optimalen Konfiguration	118
5.4	Ein Werkzeug zur Erstellung adaptiver Anwendungen	118
5.4.1	Der AdaptationProfileCreator	118

5.4.2	Erstellung von Anwendungskonfigurationen	119
5.4.3	Konfiguration der Monitoring-Umgebung	120
5.4.4	Erstellen von Profilen	120
5.4.5	Die Generierungsphase	120
5.5	Architekturmuster für adaptive Anwendungen	121
5.5.1	Das Architekturmuster Filter	122
5.5.2	Das Architekturmuster Intelligenter Proxy	123
5.5.3	Das Architekturmuster Lastverteilung	124
5.5.4	Das Architekturmuster Replikation	125
5.5.5	Das Architekturmuster Online/Offline Mobilität	125
5.5.6	Das Architekturmuster Sichere Aktualisierung	126
5.5.7	Das Architekturmuster Simplex	126
5.5.8	Das Architekturmuster Migration	126
5.5.9	Das Muster Dynamische Aspektaktivierung	127
5.5.10	Das Muster Alternativer Algorithmus	127
5.5.11	Das Muster Parameteranpassung	128
5.6	Zusammenfassung	128
6	Adaption in Steuerungssystemen	129
6.1	Das Distributed Control Lab	129
6.1.1	Die Architektur des Distributed Control Lab	129
6.1.2	Experimente im Distributed Control Lab	130
6.2	Sicherheit und Fehlertoleranz durch Adaption	133
6.2.1	Dynamische Rekonfiguration und Zustand	134
6.2.2	Analytische Redundanz	136
6.3	Ausblick: Vorhersagbare Dynamische Rekonfiguration	138
6.3.1	Vorhersagbare Ausführung von CLI-Programmen	141
6.4	Zusammenfassung	142
7	Fallstudien und Leistungsbewertung	143
7.1	Zusatzkosten der Rekonfigurationsalgorithmen	143
7.1.1	Unterbrechungsdauer von CLI-Anwendungen	145
7.2	Rekonfiguration heterogener Anwendungen	146
7.3	Fallstudie: Foucaults Pendel	148
7.4	Fallstudien: Dynamische Aktualisierung	149
7.4.1	PictureViewer - Ein Bildbetrachter	149
7.4.2	Lumisoft-Mail-Server	149
7.4.3	PaintDotNet 3.0	150
7.5	Zusammenfassung	151
8	Verwandte Arbeiten	153
8.1	Forschungsprojekte zum Thema „Adaptive Software“	153
8.1.1	Die Demeter Methode	154
8.1.2	Middleware-basierte Ansätze	155
8.1.3	Datenorientierte Adaption	160
8.1.4	Softwarearchitekturbasierte Adaption	163
8.1.5	Sprachbasierte Ansätze zur Adaption	165
8.1.6	Adaption mit aspektorientierter Programmierung	168
8.1.7	Adaption in Betriebssystemen	171

8.2	Dynamische Rekonfiguration	171
8.2.1	Der knotenpassivierende Ansatz	172
8.2.2	Der knotenblockierende Ansatz	172
8.2.3	Der verbindungsblockierende Ansatz	173
8.2.4	Dynamische Rekonfiguration in CORBA I	173
8.2.5	Dynamische Rekonfiguration in CORBA II	174
8.2.6	Rekonfiguration in POLYLITH	174
8.2.7	Rekonfiguration in OSA+	175
8.2.8	Rekonfiguration mit Port-based Objects	175
8.3	Zusammenfassung	176
9	Zusammenfassung und Ausblick	177
	Literaturverzeichnis	181