

Modellgetriebene Entwicklung komponentenorientierter Systeme mit besonderer Berücksichtigung von Dienstgüteeigenschaften

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

im Fach

Informatik

eingereicht an der

Mathematisch-Naturwissenschaftlichen Fakultät II
der Humboldt-Universität zu Berlin

von

Dipl.-Inform. Tom Ritter geboren am 31. März 1974 in Hagenow

Präsident der Humboldt-Universität zu Berlin

Prof. Dr. Christoph Markschies

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II

Prof. Dr. Peter Frensch

Gutachter/Gutachterin

1. Prof. Dr. sc. Joachim Fischer
2. Prof. Dr.-Ing. Ina Schieferdecker
3. Prof. Dr. rer. nat. habil. Andreas Polze

Tag der Verteidigung: 14. Juli 2011

Inhaltsverzeichnis

- INHALTSVERZEICHNIS.....I
- ABBILDUNGSVERZEICHNIS V
- TABELLENVERZEICHNIS VII
- KAPITEL 1
 - EINLEITUNG1
 - 1.1 MOTIVATION UND ZIELSETZUNG.....1
 - 1.2 ÜBERBLICK ÜBER DIE ERGEBNISSE DER ARBEIT3
 - 1.3 STRUKTUR DER ARBEIT.....5
 - 1.4 KONVENTIONEN6
- KAPITEL 2
 - GRUNDLAGEN UND BEGRIFFE9
 - 2.1 KOMPONENTEN.....9
 - 2.1.1 Allgemeine Eigenschaften von Komponenten9
 - 2.1.2 Komponenten in der Softwareentwicklung11
 - 2.2 DIENSTGÜTE19
 - 2.2.1 Dienstgüteeigenschaften.....22
 - 2.2.2 Weitere Gesichtspunkte der Dienstgüte25
 - 2.2.3 Schnittstellen und Dienstgüte.....26
 - 2.2.4 Dienstgüte und Aspektorientierung.....30
 - 2.2.5 Dienstgüte und Softwarequalität32
 - 2.3 MODELLGETRIEBENE SOFTWAREENTWICKLUNG34
 - 2.3.1 Modellbildung und Konzeptraum35
 - 2.3.2 Modellbeziehungen und Modelltransformation.....37
- KAPITEL 3
 - STAND DES WISSENS UND VERFÜGBARE TECHNOLOGIEN.....41
 - 3.1 KONZEPTRÄUME FÜR DIE MODELLIERUNG VON DIENSTGÜTEEIGENSCHAFTEN41
 - 3.1.1 Dienstgüte-Metamodell für DotQoS.....43
 - 3.1.2 Dienstgüte-Metamodell für Middleware nach van Halteren45
 - 3.1.3 Dienstgüte-Metamodell für CCM47
 - 3.1.4 Das Dienstgüte-Metamodell der OMG49
 - 3.1.5 Fazit55
 - 3.2 SPRACHEN ZUR BESCHREIBUNG VON DIENSTGÜTEEIGENSCHAFTEN55
 - 3.2.1 IDL-Erweiterungen zur Dienstgütebeschreibung57
 - 3.2.2 CQML59
 - 3.2.3 Platform Independent QoS Modeling Language - PIQML62
 - 3.2.4 UML-Profil für Dienstgüte.....63
 - 3.2.5 Fazit66
 - 3.3 VERWALTUNG DER DIENSTGÜTEEIGENSCHAFTEN.....66

3.3.1	<i>QoS Provisioning Service - QPS</i>	68
3.3.2	<i>QuO - Quality Objects</i>	70
3.3.3	<i>Dienstgüte für Enterprise Java Beans</i>	72
3.3.4	<i>DotQoS</i>	74
3.3.5	<i>Fazit</i>	75
KAPITEL 4		
DAS CORBA-KOMPONENTENMODELL		77
4.1	COMMON OBJECT REQUEST BROKER ARCHITECTURE	77
4.2	DER ÜBERGANG ZUR KOMPONENTENTECHNOLOGIE	79
4.3	DIE KONZEPTE DES CORBA-KOMPONENTENMODELLS	81
4.3.1	<i>BaseIDL</i>	82
4.3.2	<i>ComponentIDL und Streams</i>	84
4.3.3	<i>CIF</i>	85
4.3.4	<i>Deployment</i>	87
4.4	DIE NOTATIONEN DES CORBA-KOMPONENTENMODELLS	89
4.4.1	<i>Interface Definition Language – IDL</i>	89
4.4.2	<i>Component Implementation Definition Language – CIDL</i>	90
4.4.3	<i>Installationsdeskriptoren</i>	90
4.4.4	<i>UML-Profil für CORBA-Komponenten</i>	93
4.4.5	<i>Weitere Ansätze</i>	96
4.5	LAUFZEITUMGEBUNG	98
4.6	INSTALLATION UND INBETRIEBNAHME	100
4.7	DISKUSSION	101
KAPITEL 5		
ERWEITERUNG DES CORBA-KOMPONENTENMODELLS		103
5.1	ANFORDERUNGEN	103
5.2	ERWEITERTE MODELLIERUNGSKONZEPTE	105
5.2.1	<i>Integration der Metamodelle</i>	105
5.2.2	<i>Integration der Notationen</i>	107
5.3	CONTAINERBASIERTE PORTABLE INTERCEPTOREN	108
5.3.1	<i>Einsatz von Interceptoren</i>	108
5.3.2	<i>CORBA Portable Interceptor</i>	109
5.3.3	<i>Identität von CO</i>	110
5.3.4	<i>Entwurfsprinzipien und Flusskontrolle</i>	111
5.3.5	<i>Einfache Interceptoren</i>	113
5.3.6	<i>Erweiterte Interceptoren</i>	115
5.3.7	<i>Integration in die Aufrufkette des Containers</i>	117
5.4	ERWEITERUNGEN DER LAUFZEITUMGEBUNG	119
5.4.1	<i>Registrieren der COPI</i>	119
5.4.2	<i>Dynamisches Laden von Dienstgüteeerweiterungen</i>	120
5.4.3	<i>Einflussnahme durch die Anwendungslogik</i>	123
5.5	VERHANDLUNG VON DIENSTGÜTEEIGENSCHAFTEN	123
5.5.1	<i>Verhandlungsschnittstelle</i>	124
5.5.2	<i>Verhandlungsfluss</i>	126
5.6	ZUSAMMENFASSUNG	130
5.6.1	<i>Berücksichtigung der Anforderungen</i>	130
5.6.2	<i>Diskussion</i>	131

KAPITEL 6

MODELLGETRIEBENE ENTWICKLUNG VON CORBA-KOMPONENTEN133

6.1 ÜBERBLICK133

6.2 ENTWICKLUNGSSCHRITTE IM DETAIL.....135

6.2.1 Einführung in das Beispiel.....137

6.2.2 Plattformunabhängiger Systementwurf.....137

6.2.3 Modelltransformation zwischen PIM und PSM1140

6.2.4 Plattformspezifischer Entwurf 1162

6.2.5 Modelltransformation zwischen PSM und PSM (Modellierung).....164

6.2.6 Plattformspezifischer Entwurf 1 (Modellierung)165

6.2.7 Modelltransformation zwischen PSM1 und PSM2167

6.2.8 Plattformspezifischer Entwurf 2178

6.2.9 Plattformspezifischer Entwurf 2 (Modellierung)180

6.2.10 Code Generierung.....182

6.2.11 Implementierung183

6.2.12 Paketierung184

6.2.13 Installation und Inbetriebnahme185

6.3 INKRAFTSETZUNG DES PROZESSES186

6.4 ZUSAMMENFASSUNG.....190

KAPITEL 7

RESÜMEE193

7.1 ZUSAMMENFASSUNG.....193

7.2 EINORDNUNG DER ERGEBNISSE194

7.3 AUSBLICK195

LITERATUR199

ABKÜRZUNGEN211

INDEX.....215

ANHANG A – IDL DEFINITIONEN217