

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>v</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Aktuelle Web-Service-Technologien für serviceorientierte Architekturen . . . . .	6
1.2 Aufgabenstellung . . . . .	7
<b>2 Kontraktbasierte Modellierung, Implementierung und Suche in serviceorientierten Architekturen</b>	<b>9</b>
2.1 Design by Contract . . . . .	11
2.2 OO und DbC . . . . .	13
2.3 Modelle für Web Services . . . . .	16
2.4 Visuelle Kontrakte . . . . .	19
2.5 Model-Driven Monitoring . . . . .	21
2.6 Model-Driven Matching . . . . .	26
2.7 Fazit . . . . .	33
2.8 Aufbau der Arbeit . . . . .	34
2.9 Bibliographische Hinweise . . . . .	36
<b>3 Visuelle Kontrakte</b>	<b>39</b>
3.1 Verwandte Arbeiten . . . . .	39
3.2 Modellierung von Kontrakten . . . . .	41
3.2.1 Online Shop . . . . .	41
3.2.2 Design-Klassendiagramm des Online Shops . . . . .	44
3.2.3 Visuelle Kontrakte . . . . .	45
3.3 Metamodell für visuelle Kontrakte . . . . .	50
3.3.1 Verwendete UML 2.0 Pakete und Klassen . . . . .	51
3.3.2 Erweiterungen des UML 2.0 Metamodells . . . . .	54
3.4 Semantik von visuellen Kontrakten . . . . .	58
3.4.1 Getypte, attributierte Graphen . . . . .	60

3.4.2	Graphtransitionen . . . . .	66
3.4.3	Negative Bedingungen . . . . .	72
3.4.4	Allquantifizierung . . . . .	72
3.5	Fazit . . . . .	75
<b>4</b>	<b>Model-Driven Monitoring</b>	<b>77</b>
4.1	Verwandte Arbeiten . . . . .	78
4.2	Übersetzung von Klassendiagrammen nach Java . . . . .	81
4.3	Übersetzung visueller Kontrakte nach JML . . . . .	85
4.3.1	Teilgraphensuche . . . . .	86
4.3.2	Struktur von JML-Annotationen . . . . .	89
4.3.3	Übersetzung visueller Kontrakte nach JML: Intuitiver Ansatz . . . . .	91
4.3.4	Detaillierung der intuitiven Übersetzung . . . . .	96
4.4	Visuelle Kontrakte in der Implementierung . . . . .	106
4.5	Fazit . . . . .	109
<b>5</b>	<b>Spezifikation der Transformation</b>	<b>111</b>
5.1	Formale Spezifikationen . . . . .	111
5.2	Existierende Ansätze . . . . .	113
5.2.1	Transformation von Modellen in Code . . . . .	114
5.2.2	Transformation von Modellen in Modelle . . . . .	116
5.2.3	Hybride Ansätze: Kombination von Graphtransfor- mationen mit templatebasierten Ansätzen . . . . .	119
5.3	Spezifikation der Transformation nach JML . . . . .	120
5.3.1	Modelltransformationen mit Compound Rules . . . . .	121
5.3.2	Kontrollstrukturen . . . . .	124
5.4	Transformationsregeln . . . . .	127
5.5	Fazit . . . . .	133
<b>6</b>	<b>Model-Driven Matching</b>	<b>135</b>
6.1	Verwandte Arbeiten . . . . .	136
6.1.1	Web-Service-Standards . . . . .	137
6.1.2	Wissenschaftliche Ansätze . . . . .	139
6.2	Semantische Beschreibungen . . . . .	142
6.2.1	Ontologien . . . . .	142
6.2.2	Kombination von Ontologien und visuellen Kontrakten	144
6.3	Matching von visuellen Kontrakten . . . . .	148
6.3.1	Matching — intuitive Definition . . . . .	149
6.3.2	Operationale Interpretation des Matchings . . . . .	154
6.4	Matching — formale Definition . . . . .	156

6.5	Fazit . . . . .	162
<b>7</b>	<b>Matching, Semantic-Web-Technologien</b>	<b>165</b>
7.1	Semantic Web . . . . .	166
7.2	Semantic-Web-Sprachen . . . . .	167
7.2.1	RDF . . . . .	168
7.2.2	RDF Vocabulary Description Language . . . . .	168
7.2.3	DAML+OIL . . . . .	170
7.2.4	RDQL . . . . .	171
7.3	Ontologien, visuelle Kontrakte, Semantic Web . . . . .	173
7.3.1	Repräsentation von Ontologien . . . . .	173
7.3.2	Repräsentation von visuellen Kontrakten . . . . .	174
7.4	Matching-Algorithmus . . . . .	175
7.5	Fazit . . . . .	179
<b>8</b>	<b>Visual Contract Workbench</b>	<b>181</b>
8.1	Eclipse . . . . .	182
8.1.1	Eclipse Modeling Framework . . . . .	182
8.1.2	Graphical Editor Framework . . . . .	183
8.1.3	Java Emitter Templates . . . . .	184
8.2	Produktfunktionen . . . . .	185
8.2.1	Modellierung von Klassendiagrammen . . . . .	186
8.2.2	Modellierung von visuellen Kontrakten . . . . .	189
8.2.3	Codegenerierung . . . . .	195
8.2.4	Generierung von Semantic-Web-Repräsentationen . . . . .	197
8.2.5	Modelle als Vorlage verwenden . . . . .	197
8.2.6	Bearbeiten des Java-Codes . . . . .	198
8.2.7	Starten des Compilers . . . . .	199
8.3	Architektur der Visual Contract Workbench . . . . .	199
8.4	Fazit . . . . .	205
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>207</b>
9.1	Zusammenfassung der Ergebnisse der Arbeit . . . . .	207
9.2	Evaluation . . . . .	208
9.3	Ausblick . . . . .	211
9.4	Abschluss . . . . .	213
<b>A</b>	<b>Beschreibung Metamodellklassen</b>	<b>215</b>
A.1	Klasse Constraint . . . . .	215
A.2	Klasse DependentParameter . . . . .	218
A.3	Klasse NegativeCondition . . . . .	219

A.4	Klasse Parameter . . . . .	221
A.5	Klasse Postcondition . . . . .	221
A.6	Klasse Precondition . . . . .	223
A.7	Klasse VCElement . . . . .	225
A.8	Klasse VisualContract . . . . .	227
<b>B</b>	<b>Transformationsregeln VC2JML</b>	<b>229</b>
B.1	Grundgerüst einer JML-Spezifikation . . . . .	229
B.1.1	Erstellung der Operationen . . . . .	230
B.1.2	Erstellung der Parameterliste . . . . .	231
B.2	Erstellung der Vorbedingung . . . . .	236
B.2.1	Überprüfung der Variablen des Elements self . . . . .	236
B.2.2	Abhängige Elemente des Objekts self überprüfen . . . . .	237
B.2.3	Abhängige Elemente überprüfen . . . . .	240
B.2.4	Links zu besuchten Elementen überprüfen . . . . .	248
B.2.5	Abschluss der Suche . . . . .	255
B.2.6	Überprüfung der Attributinhalt — Vorbereitung . . . . .	258
B.3	Attributinhalt in Vorbedingung . . . . .	259
B.4	Bearbeitung von Variablen . . . . .	262
B.4.1	Variable ist Attributinhalt eines Operationsparameters . . . . .	262
B.4.2	Variable ist kein Attributinhalt eines Operationsparameters . . . . .	266
<b>C</b>	<b>DAML+OIL</b>	<b>271</b>
C.1	DAML+OIL Ontologie . . . . .	271
C.2	DAML+OIL Kontrakt . . . . .	277
	<b>Literaturverzeichnis</b>	<b>279</b>