

Inhalt

1	Einleitung	1
1.1	Softwaremuster	3
1.2	Verbesserung der Wartbarkeit	4
1.3	Ergebnisse der Arbeit	6
1.4	Aufbau der Arbeit	9
2	Grundlagen	11
2.1	Anti Patterns	11
2.2	Refactoring	15
2.3	Strukturbasierte Mustererkennung	18
2.3.1	Repräsentation eines Softwaresystems	19
2.3.2	Spezifikation von Strukturmustern	20
2.3.3	Erkennungsprozess	26
2.3.4	Bewertung der Ergebnisse	29
2.3.5	Erweiterung um quantitative Merkmale	32
2.4	Story Driven Modeling	35
2.4.1	Story Pattern	36
2.4.2	Story Diagramme	38
2.4.3	Verifikation von Story Pattern	40
3	Formale Spezifikation von Programmtransformationen	45
3.1	Anforderungen	45
3.2	Transformationsdiagramme	48
3.2.1	Pfade	54
3.2.2	Iteration	57
3.2.3	Transformationsaufrufe	61
3.2.4	Strukturmuster	63
3.3	Formalisierung von Transformationsdiagrammen	67
3.3.1	Vorbemerkungen	68
3.3.2	Typisierte Graphen	68

3.3.3	Transformationsdiagramme	71
3.3.4	Transformation Pattern	72
3.3.5	Zustand	76
3.3.6	Anwendungsstellen	77
3.3.7	Ausführung von Graphtransformationen	79
3.3.8	Ausführung von Transformation Pattern	84
3.3.9	Ausführung von Transformationsdiagrammen	88
3.3.10	Strukturmusterannotationen	89
3.4	Synthese von Transformationsspezifikationen anhand von Quell-	
	textbeispielen	90
3.4.1	Syntheseverfahren	92
3.4.2	Verwendung des Verfahrens	97
4	Verifikation von Transformationsspezifikationen	101
4.1	Verifikationskriterien	101
4.2	Überblick über das Verifikationsverfahren	104
4.2.1	Gegenbeispiele	108
4.3	Bestimmung von Ausführungspfaden	112
4.4	Verifikation für verbotene Graphmuster	114
4.4.1	Bestimmung problematischer iterierter Anteile	116
4.4.2	Berechnung initialer Regelsequenzen	119
4.4.3	Kennzeichnung optionaler Elemente	132
4.4.4	Vorwärtsüberprüfung und Vervollständigung	143
4.4.5	Generierung negativer Anwendungsbedingungen	144
4.4.6	Ergebnis	146
4.5	Verifikation für zu erhaltende Graphmuster	146
4.5.1	Bestimmung problematischer iterierter Anteile	147
4.5.2	Berechnung initialer Regelsequenzen	148
4.5.3	Kennzeichnung optionaler Elemente	149
4.5.4	Generierung negativer Anwendungsbedingungen	150
4.5.5	Ergebnis	151
4.6	Aussage des Verfahrens	153
5	Werkzeugunterstützung	157
5.1	Benutzungsschnittstelle	158
5.1.1	Spezifikation von Strukturmustern und Transformati-	
	onsdiagrammen	159
5.1.2	Verifikation von Transformationsdiagrammen	162
5.1.3	Strukturbasierte Mustererkennung	163

5.1.4	Ausführung von Transformationsdiagrammen	165
5.2	Architektur	166
6	Evaluierung	169
6.1	Strukturmuster zur Erkennung von Schwachstellen	170
6.1.1	Eingesetzte Metriken	170
6.1.2	Verkapselung und Verteilung von Verantwortung	171
6.1.3	Vererbungshierarchien	173
6.1.4	Bedingte Anweisungen	174
6.1.5	Objekterzeugung	175
6.1.6	Falsche Verwendung von Entwurfsmustern	176
6.1.7	Festlegung der Bewertungsfunktionen	177
6.2	Ergebnisse der Schwachstellenanalyse	181
6.2.1	Schwachstellen in SWT	181
6.2.2	Fazit	188
6.3	Transformationsspezifikationen	189
6.3.1	Fazit	192
6.4	Verifikation	193
6.4.1	Fazit	197
7	Verwandte Arbeiten	199
7.1	Erkennung von Schwachstellen	199
7.2	Restrukturierung von Software	204
7.2.1	Restrukturierung auf Basis von Entwurfsmustern	207
7.2.2	Vollständige Formalisierung von Refactorings	210
7.2.3	Refactoring formaler Modelle	212
7.2.4	Exogene Modelltransformationen	214
7.2.5	Transformationspezifikation anhand von Beispielen	215
7.3	Verifikation von Graphtransformationssystemen	215
7.3.1	Korrektheit per Konstruktion	216
7.3.2	Model Checking	217
7.3.3	Analyse mit Petrinetztechniken	218
7.4	Zusammenfassung	219
8	Zusammenfassung und Ausblick	221
8.1	Zusammenfassung	221
8.2	Ausblick	223
	Literatur	227

A Verkleben von Graphmustern mit Pfaden	245
A.1 Verkleben ohne Pfade	246
A.2 Abbilden von Pfaden	246
Abbildungen	251
Tabellen	255