

Inhaltsverzeichnis

1. Einleitung	1
1.1. Kontext der Arbeit	3
1.2. Ziele und Ergebnisse der Arbeit	4
1.3. Aufbau der Arbeit	5
2. Grundlagen modellgetriebener Softwareentwicklungsprojekte	7
2.1. Modellbasierte und modellgetriebene Softwareentwicklung	7
2.1.1. Sprach- und Modellkomposition	8
2.1.2. Modelltransformationen	9
2.2. Modellgetriebene Softwareentwicklungsprojekte	10
2.3. MontiCore	12
2.3.1. Überblick	12
2.3.2. Symboltabelle	13
2.3.3. Templates	14
2.3.4. Reports	15
2.4. Ausgewählte Modellierungssprachen und modellverarbeitende Werkzeuge	16
2.4.1. UML/P	16
2.4.2. MontiArc	18
2.4.3. DEx	19
3. Analyse von Softwareentwicklungsprojekten	21
3.1. Software Reverse Engineering	21
3.2. Traceability und Artefaktabhängigkeiten	22
3.3. Extraktion von Traceabilityinformationen	25
3.3.1. Statische Extraktion	25
3.3.2. Dynamische Extraktion	26
3.4. Nutzung von Traceabilityinformationen	27
3.4.1. Abhängigkeitsanalyse	27
3.4.2. Change-Impact-Analyse	27
3.4.3. Abdeckungsanalyse	28
3.5. Architekturekonstruktion und Architekturkonformität	28
3.6. Metriken in der Softwareentwicklung	30

3.7.	Visualisierung von Abhängigkeiten in der Softwareentwicklung	32
3.7.1.	Abhängigkeitsmatrix	32
3.7.2.	Graphdarstellung	33
3.7.3.	Softwarestadt	34
3.7.4.	Architekturvisualisierung	35
4.	Artefaktbasierte Analyse	39
4.1.	Artefakte in modellgetriebenen Softwareentwicklungsprojekten	39
4.2.	Szenario	41
4.3.	Lösungsansatz	44
4.4.	Grundlegende Begriffe	47
4.5.	Anforderungen	48
4.5.1.	Anforderungen an Analyseverfahren	49
4.5.2.	Anforderungen an die Modellierungstechnik	49
4.5.3.	Anforderungen an die Werkzeugunterstützung	50
5.	Modellierungstechnik für die Artefaktmodellierung	53
5.1.	Verwendung der UML/P zur Modellierung von Artefaktzusammenhängen . .	53
5.1.1.	Klassendiagramme	54
5.1.2.	Objektdiagramme	55
5.1.3.	Object Constraint Language	57
5.1.4.	Semantik	58
5.2.	Artefaktmodellkern	61
5.2.1.	Artefakte und Artefaktcontainer	61
5.2.2.	Systeme und Werkzeuge	67
6.	Artefaktmodell für modellgetriebene Softwareentwicklungsprojekte mit MontiCore	71
6.1.	Java Artefakte	71
6.1.1.	Java-Quellcode- und .class-Dateien	72
6.1.2.	Beziehungen zwischen Java-Artefakten und Java-Typen	75
6.2.	Definition von Modellierungssprachen	80
6.2.1.	Modellierungssprachen	81
6.2.2.	Grammatiken	82
6.2.3.	Modelldateien	83
6.3.	Klassendiagramm Artefakte	83
6.4.	Generatorartefakte	86
6.4.1.	Templates	86
6.4.2.	Generatoren	88
6.5.	Dynamische Beziehungen zur Generierungszeit	90
6.5.1.	Aktionen und Events	90

6.5.2.	Aktionen des Generierungsprozesses	92
6.5.3.	Ein- und Ausgaben von Werkzeugen	94
6.5.4.	Beteiligung von Java und Templatedateien an der Generierung	96
6.6.	Beispielhafte Analysespezifikationen	97
6.6.1.	Architekturrekonstruktion	98
6.6.2.	Differenzarchitektur	99
6.6.3.	Indirekte Templatebeziehungen	101
6.6.4.	Ungenutzte Importe	102
6.6.5.	Inkrementelle Ausführung von Werkzeugketten	104
6.7.	Verwandte Ansätze	106
6.7.1.	Megamodelle	106
6.7.2.	Artefaktorientierung und Artefaktmodelle	107
7.	Methodik zur artefaktbasierten Analyse modellgetriebener Software- entwicklungsprojekte	111
7.1.	Methodik zur Erstellung projektspezifischer Artefaktmodelle	112
7.2.	Methodik zur Spezifikation projektspezifischer Analysen	115
7.2.1.	OCL-basierte Analysen	116
7.2.2.	Abhängigkeitsanalyse und Change-Impact-Analyse	116
7.2.3.	Architekturanalysen	117
7.2.4.	Metriken	118
7.2.5.	Nutzungsart von OCL-Ausdrücken spezifizieren	120
7.3.	Datenextraktion	121
7.4.	Merge von Artefaktdaten	123
7.5.	Datenprüfung und -anreicherung	127
7.5.1.	Auswertung von OCL-Prüfbedingungen	127
7.5.2.	Anreicherung	127
7.6.	Analyse von Artefaktdaten	129
8.	MontiArT: Eine Werkzeugkette zur Durchführung artefaktbasierter Ana- lysen	137
8.1.	Werkzeuge	138
8.1.1.	Ein- und Ausgaben	139
8.1.2.	Nutzung	142
8.1.3.	Implementierung	142
8.2.	Datenextraktion	145
8.2.1.	Aufbau statischer Extraktoren	146
8.2.2.	Artifact Container Extractor	148
8.2.3.	Template Extractor	150
8.2.4.	Java Extractor	151

8.3.	Merge von Artefaktdaten	152
8.3.1.	OD Merger	152
8.3.2.	Verwendung des OD Mergers in MontiArT	154
8.4.	Datenprüfung und Anreicherung	154
8.4.1.	Fehlerarten bei der Konsistenzprüfung	155
8.4.2.	Strukturprüfung	156
8.4.3.	OCL Generator	157
8.4.4.	Auswertung von OCL-Prüfbedingungen	160
8.4.5.	Anreicherung der abgeleiteten Eigenschaften	162
8.5.	Datenanalyse	163
8.5.1.	Artifact Data Explorer	163
8.5.2.	OCL-basierte Analysen	164
8.5.3.	OD Filter	165
8.5.4.	Architekturanalyse	166
8.5.5.	Visualisierung von Artefaktdaten	168
8.6.	Konverterwerkzeuge	177
8.6.1.	OD Flattener	177
8.6.2.	OD2JSON Transformation	178
8.6.3.	CD OCL Splitter	179
8.6.4.	CD Merger	180
9.	Methodik zur Erstellung von MontiArT-Instanzen	183
9.1.	Auswahl mitgelieferter Extraktoren	184
9.2.	Implementierung projektspezifischer Werkzeuge	184
9.2.1.	Implementierung neuer Werkzeuge	185
9.2.2.	Extraktion von Artefaktbeziehungen mit Hilfe der Symboltabelle	185
9.2.3.	Verwendung und Anpassung der Reportinginfrastruktur	188
9.3.	Fehlererkennung bei skriptbasierter Verkettung von Werkzeugen	190
9.4.	Konfiguration von ArtViz	192
9.5.	Einbinden von MontiArT in eine Software-Engineering-Infrastruktur	196
9.5.1.	Software-Engineering-Infrastruktur	196
9.5.2.	Einbinden der Werkzeugkette	197
10.	Artefaktbasierte Analysen in der Anwendung	199
10.1.	Der DEx-Generator	199
10.1.1.	DEx Transformation Extractor	200
10.1.2.	DEx Action Reporter	202
10.1.3.	Durchführung der Analyse	205
10.2.	Eingabeartefakte des BumperBots	209
10.3.	Make-basierte Buildprozesse	212
10.3.1.	Methodik zur Instrumentierung von Makefiles	214

10.3.2. Werkzeug zur automatisierten Instrumentierung von Makefiles	215
10.4. Inkrementelle Generierung mit MontiCore	216
10.4.1. Lösungsansätze	221
10.4.2. Evaluation	225
10.5. Artefakte bei der Testfallgenerierung im Automotivkontext	227
11. Zusammenfassung und Ausblick	231
11.1. Zusammenfassung	231
11.2. Ausblick	233
Literaturverzeichnis	235
A. Markierungen in Abbildungen und Listings	257
B. Tracing der Anforderungen	259
C. Artefaktmodell für MontiCore-basierte MDD-Projekte	261
D. Tagschema für artefaktbasierte Analysen	275
E. MontiArT-Werkzeuge	277
F. DEx-Artefaktanalyse	279
F.1. Definition der DEx-MontiArt-Instanz als Skript	279
F.2. Moduldefinition und Soll-Architektur	285
F.3. Ist-Architektur	297
F.4. Differenzarchitektur	300
G. BumperBot-Artefaktanalyse	303
G.1. Extrahierte Artefaktdateien	303
G.2. Ungenutzte Importe	308
H. Lebenslauf	309
Abkürzungsverzeichnis	311
Abbildungsverzeichnis	313
Listings	317
Tabellenverzeichnis	321