

Inhaltsverzeichnis

Abkürzungsverzeichnis	11
1. Einleitung	15
1.1. Motivation	15
1.2. Einordnung und Beitrag	17
1.3. Übersicht	18
1.4. Konventionen	20
2. Grundlagen	21
2.1. Traceability	21
2.1.1. Definitionen	21
2.1.2. Eine Ontologie für Traceability-Informationen	26
2.1.3. Die Wertschöpfungskette der Traceability	31
2.1.4. Stand der Wissenschaft	34
2.2. Modellierung und Metamodellierung	42
2.2.1. Der allgemeine Modellbegriff	42
2.2.2. Modellierung und Metamodellierung im Software Engineering	43
2.2.3. Festlegungen zur Modellierung in dieser Arbeit	50
3. TraceML: Eine Modellierungssprache für Traceability-Informationen	55
3.1. Motivation	56
3.2. Übersicht	58
3.3. Grundlagen der TraceML	63
3.3.1. Ein formales Modell von Traceability-Informationen	63
3.3.2. Die Beschreibung von Traceability Links	66
3.4. Die TraceML Infrastructure	77
3.4.1. Das Traceability-Metamodell	77
3.4.2. Das Traceability Modeling Profile	81
3.4.3. Das Core-Paket	82
3.4.4. Beispiel	103
3.5. Die TraceML Superstructure	105
3.5.1. Extraktion: Das Extraktionsmodell	109
3.5.2. Repräsentation: Die Link Library	131

3.5.3.	Analyse: QVT als Analysesprache	160
3.6.	Beispiel	168
3.7.	Einflüsse anderer wissenschaftlicher Arbeiten auf die TraceML	169
3.7.1.	Modellbasiert auf Basis des MOF-Standards	169
3.7.2.	Traceability-Informationen in einem eigenständigen Modell	170
3.7.3.	Baseline-Orientiertierung	171
3.7.4.	Anwendungsspezifische Traceability-Modelle	172
3.7.5.	Unterstützung der gesamten Traceability-Wertschöpfungskette	173
3.7.6.	Aufteilung in Infrastructure und Superstructure	174
3.8.	Validierung der TraceML	174
4.	ReflexML: Konsistenzanalyse zwischen Architektur und Code	177
4.1.	Einleitung	178
4.2.	Die Beispiel-Applikation	182
4.3.	Architektur-zu-Code Traceability	183
4.3.1.	UML Reflexion Profile	184
4.3.2.	Die Syntax von Reflexion-Ausdrücken	186
4.3.3.	Beispielhafte Anwendung des UML Reflexion Profile	188
4.4.	Architektur-zu-Code-Konsistenzprüfregeln	188
4.4.1.	Absenzen	189
4.4.2.	Divergenzen	189
4.5.	Umsetzung der ReflexML auf Basis der TraceML	196
4.5.1.	Extraktion	196
4.5.2.	Repräsentation	198
4.5.3.	Analyse	204
4.5.4.	Übersicht und Bewertung	207
4.6.	Stand von Wissenschaft und Technik	208
4.7.	Fallstudie	211
4.8.	Validierung der ReflexML gegen die Anforderungen	215
4.9.	Zusammenfassung zur ReflexML	218
5.	Der Softwareleitstand	219
5.1.	Konzept	219
5.1.1.	Das Informationsmodell des Softwareleitstands	220
5.1.2.	Das erweiterte Reflexion-Modell	222
5.2.	Architektur	224
5.2.1.	Integration Pipeline	225
5.2.2.	Holistic Model	228
5.2.3.	Analysis Core	228
5.3.	Einsatz der TraceML	228
5.3.1.	Extraktion	228

5.3.2. Repräsentation	229
5.3.3. Analyse	230
5.4. Zusammenfassung	232
6. Zusammenfassung und Ausblick	235
6.1. Zusammenfassung	235
6.2. Ausblick	235
6.3. Hinweis	237
A. Erläuterung der bekannten Link-Schema-Ansätze	239
A.1. Ramesh2001	239
A.2. Espinoza2006	240
A.3. Walderhaug2006	241
A.4. Goknil2008	243
A.5. Drivalos2009	245
A.6. Maeder2009	247
A.7. Schwarz2010	248
A.8. OMG UML2	250
A.9. W3C OWL2	251
B. TraceML Metamodellierung	255
B.1. Ontologisches Metamodell	255
B.2. Linguistische Metamodelle	255
C. Semantik von Link-Typen	259
C.1. Mapping des Link-Schemas auf TraceML-Modellelemente	259
C.2. Muster der Multiplizitäten von Link-Typen	260
C.3. Übersicht der formalen Link-Eigenschaften der Link Library	260
D. Listings	263
D.1. AspectJ Type Pattern Syntax	263
D.2. QVT Listings	264
Tabellenverzeichnis	269
Abbildungsverzeichnis	271
Literatur	275
Index	293