

Inhaltsverzeichnis

Kurzfassung	i
Abstract	ii
Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
1. Einführung	1
1.1. Problemstellung	2
1.1.1. Beispiel zur Umsetzung von Verfolgbarkeit	2
1.1.2. Ursachen für aufwändige Verfolgbarkeitsanalysen	3
1.2. Einordnung und Beitrag	4
1.3. Kapitelübersicht	6
2. Grundlagen	7
2.1. Verfolgbarkeit	7
2.1.1. Begriffe	7
2.1.2. Verfolgbarkeitsprozess	9
2.2. Modell	11
2.2.1. Modellbegriff	11
2.2.2. Metamodell	13
2.2.3. Graph	13
2.3. Domänen spezifische Sprache	15
3. Stand von Wissenschaft und Technik	17
3.1. Analysearten	17
3.2. Bewertungskriterien	20
3.2.1. Anwendbarkeit	20
3.2.2. Repräsentationsunabhängigkeit	22
3.2.3. Konfigurierbarkeit	23
3.3. Manuelle Ansätze	24
3.3.1. Visualisierungstechniken	25
3.3.2. Bewertungskriterien	26
3.4. Werkzeugbasierte Ansätze	28
3.4.1. Allgemeine Werkzeuge	28
3.4.2. Domänen spezifische Werkzeuge	32
3.4.3. Domänen spezifische Arbeitsumgebungen	34
3.5. Sprachbasierte Ansätze	36
3.5.1. Allgemeine Programmiersprachen	36
3.5.2. Allgemeine domänenspezifische Sprachen	37
3.5.3. Domänen spezifische Sprachen	38
3.6. Zusammenfassung	41

4. TracAL: Die Traceability Analysis Language	43
4.1. Anforderungen	43
4.1.1. Zielgruppe	43
4.1.2. Einsatzszenarien	44
4.1.3. Realisierungsaufwand	45
4.2. Festlegung der Funktionalität	45
4.2.1. Datenanbindung	45
4.2.2. Anfrage	45
4.2.3. Auswertung und Verarbeitung	46
4.3. Auswahl der Ansatzkategorie	47
4.3.1. Spracheigenschaften	47
4.3.2. Modelleigenschaften	51
4.4. Realisierung	53
4.4.1. Funktionsbibliothek	53
4.4.2. Syntaxerweiterungen	56
4.4.3. Verteilung	57
4.5. Arbeiten mit <i>TracAL</i>	57
4.5.1. Datenanbindung	58
4.5.2. Anfrage	60
4.5.3. Auswertung und Verarbeitung	61
4.5.4. Erweiterungen	62
4.6. Verwandte Arbeiten	64
4.6.1. Alternative Ansätze zur Umsetzung von Verfolgbarkeitsanalysen	65
4.6.2. Abgrenzung zu den Arbeiten des Forschungspartners	68
4.7. Zusammenfassung	69
5. Repräsentationsunabhängigkeit	71
5.1. Statisch typisierte Graphen	71
5.1.1. Problembeschreibung und Lösungsansatz	71
5.1.2. Erweiterung des Eigenschaften-Graphmodells	74
5.1.3. Filterkonzept	76
5.1.4. Generische und konkrete Typfilter	78
5.1.5. Allgemeine Filter	80
5.1.6. Statisch typisierte Graphklassen	83
5.1.7. Polymorphie statisch typisierter Graphklassen	87
5.1.8. Allgemeine Graphoperationen	89
5.1.9. Graphschema	91
5.1.10. Verwandte Arbeiten	95
5.2. Adapterkonzept	98
5.2.1. Allgemeiner Prozess zur Datenanbindung	98
5.2.2. Problembeschreibung und Lösungsansatz	100
5.2.3. Mehrstufiges Adapterkonzept	101
5.2.4. Modulare Bausteine	106
5.2.5. Verbesserte Repräsentationsunabhängigkeit	110
5.2.6. Verwandte Arbeiten	111
5.3. Zusammenfassung	115
6. Konfigurierbarkeit	117
6.1. Motivation	117

6.2.	Lösungsansatz	119
6.2.1.	Monaden	120
6.2.2.	Anwendungsbeispiel	121
6.2.3.	Einschränkungen	123
6.2.4.	Wissenschaftlicher Beitrag	124
6.3.	Klassen mit monadischen Effekten	125
6.3.1.	Datentyp	125
6.3.2.	Typklasse	127
6.4.	Effektstapel	131
6.4.1.	Aufbau	132
6.4.2.	Syntaxerweiterungen	134
6.4.3.	Umsetzung von Stapeloperationen	139
6.4.4.	Bearbeitungsreihenfolge	142
6.5.	Stapelbasierte Traversierschritte	144
6.5.1.	Allgemeine Traversierschritte	144
6.5.2.	Elementbasierte Traversierschritte	147
6.5.3.	Die Festlegung von Abarbeitungsregeln	149
6.5.4.	Das Einbringen von Effektberechnungen	150
6.5.5.	Wiederholungen	153
6.6.	Konfigurierbare Verfolgbarkeitsanalysen	156
6.6.1.	Modularität	156
6.6.2.	Erweiterbarkeit	157
6.6.3.	Anpassbarkeit	157
6.7.	Verwandte Arbeiten	158
6.7.1.	Monaden und Monadentransformatoren	158
6.7.2.	Funktionale Graphalgorithmen	160
6.7.3.	Alternative Umsetzung des Effektstapels	160
6.7.4.	Effektberechnungen in anderen graphbasierten Sprachen	162
6.8.	Zusammenfassung	163
7.	Evaluation	165
7.1.	Testkonfiguration	166
7.1.1.	Hardware	166
7.1.2.	Software	166
7.2.	Eigenschaften des Effektstapels	167
7.2.1.	Untersuchung des Laufzeitmehraufwands	167
7.2.2.	Untersuchung des Vorteils statischer Typisierung	171
7.2.3.	Zusammenfassung	172
7.3.	Fallstudie Zyklendetektion	172
7.3.1.	Aufbau und Umsetzung der Analyse in <i>TracAL</i>	173
7.3.2.	Vergleich mit anderen Ansätzen	177
7.3.3.	Laufzeitmessungen	178
7.3.4.	Erweiterung der Analyse	183
7.3.5.	Zusammenfassung	184
7.4.	Fallstudie Divergenzdetektion	185
7.4.1.	Aufbau und Umsetzung der Analyse in <i>TracAL</i>	185
7.4.2.	Vergleich mit anderen Ansätzen	191
7.4.3.	Vergleich der Ausdrucksstärke	192
7.4.4.	Laufzeitmessungen	201
7.4.5.	Erweiterung der Analyse	203

Inhaltsverzeichnis

7.4.6. Zusammenfassung	204
7.5. Fallstudie iTrust	204
7.5.1. Datenanbindung	206
7.5.2. Anfrage, Auswertung und Verarbeitung	217
7.5.3. Zusammenfassung	222
7.6. Zusammenfassung	223
8. Zusammenfassung und Ausblick	225
8.1. TracAL: Die Traceability Analysis Language	225
8.1.1. Anwendbarkeit	225
8.1.2. Repräsentationsunabhängigkeit	226
8.1.3. Konfigurierbarkeit	227
8.1.4. Evaluation	228
8.2. Ausblick	229
A. Anhang	231
A.1. UML-Klassendiagramme für Scala	231
A.1.1. Typparameter	232
A.1.2. Stereotypen	233
A.1.3. Methoden und Funktionen	234
A.2. Referenzimplementierung ausgewählter Effektberechnungen	235
A.2.1. Identitätsklasse	235
A.2.2. Pfadeffekt	238
A.2.3. Besuchseffekt	239
Liste der Veröffentlichungen	243
Hinweis und Danksagung	244
Quellenverzeichnis	245