

Modellierung und Realisierung von Konsistenzsicherungswerkzeugen für simultane Dokumentenentwicklung

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH
Aachen University zur Erlangung des akademischen Grades einer Doktorin der
Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatikerin
Anne-Thérèse Körtgen

aus Bonn-Beuel

Berichter: Universitätsprofessor Dr.-Ing. Manfred Nagl
Universitätsprofessor Dr. rer. nat. Wilhelm Schäfer

Tag der mündlichen Prüfung: 25. Juni 2009

Zusammenfassung

Die Ergebnisse von Entwicklungsprozessen werden in Dokumenten festgehalten. Diese enthalten Spezifikationen/Realisierungen des zu entwickelnden Systems aus unterschiedlichen Blickwinkeln und auf verschiedenen Abstraktionsstufen. Dokumente können sich inhaltlich überlappen, wodurch es bei paralleler Bearbeitung der Dokumente zu inkonsistenten, d. h. widersprüchlichen Beschreibungen kommen kann.

Thema dieser Arbeit sind allgemeine Konzepte und Werkzeuge zur Konsistenzwiederherstellung, auch *Integration* genannt, wobei die syntaktische Konsistenz zwischen Dokumenten verschiedener Abstraktionsstufen im Vordergrund steht. Die Arbeit baut auf den im SFB IMPROVE erzielten Ergebnissen zu Integrationswerkzeugen auf. Zur Demonstration der Anwendbarkeit der eingeführten Konzepte werden insbesondere Konsistenzprobleme in verfahrenstechnischen Entwicklungsprozessen und zwischen den dort entwickelten Fließbildern untersucht.

Bestehende Ansätze lösen Inkonsistenzen durch Transformation auf, die meisten Ansätze, wie auch der Ansatz dieser Arbeit, verwenden Modell- und Graphtransformationen, wobei sie feingranulare Beziehungen überlappender Strukturen zwischen Dokumenten speichern. Die Vorgehensweisen zur Inkonsistenzbehebung sind jedoch nicht ausreichend.

Des Weiteren sind die verwendeten Transformationssprachen für umfangreiche, komplexe Dokumente nicht ausdrucksstark genug. Es bedarf einiger Modularisierungs- und Operationalisierungskonzepte sowie weiterer Ausdrucksmittel zur Spezifikation komplexer Graphstrukturen. Zudem berücksichtigen sie für die Erzeugung von Dokumentelementen nicht das Layout der Dokumente und das in den Graphmustern der Transformationen angegebene Layout, welches insbesondere in verfahrenstechnischen Fließbildern sehr wichtig ist.

Die vorliegende Arbeit liefert allgemeine Konzepte zur Konsistenzwiederherstellung und demonstriert die Umsetzbarkeit mit einer prototypischen Realisierung an Beispielen aus der Verfahrenstechnik und der Softwareentwicklung. Verschiedenartige, konsistenzwiederherstellende Transformationen, *Reparaturaktionen* genannt, werden dynamisch zur Laufzeit erzeugt unter Berücksichtigung der inkonsistenten Beziehungen zweier Dokumente und der dort auftretenden Fehler. Vom Benutzer durchgeführte komplizierte Restrukturierungen an einem Dokument werden durch die dynamische Behandlung erkannt. Die erstellten Transformationen definieren verschiedene Alternativen, wie sich Änderungen des Benutzers auf beide Dokumente auswirken können (u. U. an beiden Dokumenten gleichzeitig), um einen konsistenten Zustand zu erreichen. Die Alternativen werden dem Benutzer präsentiert, der durch die Auswahl einer Transformation bestimmt, wie die Inkonsistenz behoben werden soll.

Die zur Wiederherstellung der Konsistenz verwendete Graphtransformationssprache wurde modularisiert, d. h. Graphmuster- und Transformationsspezifikationen können in andere Spezifikationen zur Wiederverwendung importiert werden. Ent-

weder werden sie ganzheitlich importiert, um diese zu erweitern, oder anteilig, um die Anteile in der linken Regelseite einer Transformation als sog. *Kontext* zu verwenden. Es entstehen zwischen Transformationen je nach Verwendungsart *Erweiterungs-* oder *Kontextbeziehungen*.

Die Erweiterungsbeziehungen werden insbesondere bei der Ermittlung geeigneter Reparaturaktionen berücksichtigt, um ggf. noch konsistente Korrespondenzbeziehungen bei *kleineren* zu verwendenden Transformationen zu erhalten. Kontextbeziehungen zwischen Transformationen optimieren die Mustersuche und ermöglichen Abhängigkeitsanalysen zwischen ausführbaren Transformationen. Diese Abhängigkeiten sind für den Benutzer bei einer Transformationsentscheidung nützlich.

Des Weiteren erlaubt die Transformationssprache operationale Aufrufe anderer Transformationen mit imperativen Sprachelementen. Es wurden außerdem Ausdrucksmittel eingeführt, um mengenwertige, wiederholte und alternative Graphmuster zu spezifizieren, und Ausdrucksmittel, um ähnliche Elemente mit Stringmatching-Techniken und Synonymvergleichen aufzuspüren.

Die Erstellung der Transformationen ist ein aufwändiger Prozess. Zur Unterstützung dieses Prozesses sind in dieser Arbeit zwei Ansätze entwickelt worden, in denen aus bestehenden Korrespondenzbeziehungen (i) zwischen Elementen der Dokumentenebene bzw. (ii) zwischen Typen und Attributen der Dokumentenmodellebene Transformationen induziert werden. Die Korrespondenzbeziehungen werden zuvor mit einer interaktiven Korrespondenzanalyse ermittelt, die auch von dem Konsistenzsicherungswerkzeug durchgeführt wird.

Die Analysen finden - jeweils passend zu einem generischen Regelsatz, der nur *grobe* Korrespondenzbeziehungen definiert - existierende *konkrete* Beziehungen zwischen Dokumenten bzw. Dokumentenmodellen. Ein Regelsatz für die Dokumentenebene wird *jeweils* für ein *bestimmtes* Paar von Dokumententypen modelliert und bedient sich insbesondere der Ausdrucksmittel für komplexe Graphmuster. Im Gegensatz dazu gibt es nur *einen* Regelsatz für die Dokumentenmodellebene, in dem insbesondere Ausdrucksmittel zur Ermittlung ähnlicher Elemente verwendet werden. Attributkorrespondenzen zwischen komponierten Typen sind in den Regeln ebenso definiert.

Die Ergebnisse dieser Arbeit verbessern das Vorgehen zur Erstellung von Integrationswerkzeugen. So kann ein Integrator jetzt zur Laufzeit konfiguriert werden. Des Weiteren sind einfach bedienbare Werkzeuge zur Regelmodellierung und Integrationsdurchführung entstanden, die leicht in bestehende Entwicklungsumgebungen integriert werden können. Durch diese Werkzeuge wird die Integration zusammen mit den neu eingeführten Reparaturaktionen realitätsnah in die Praxis umgesetzt.

Danksagung

Obwohl die Arbeit von mir allein verfasst ist, hatte ich vielfältige Unterstützung von Menschen aus meinem Umfeld, denen ich an dieser Stelle herzlich danken möchte.

Zunächst möchte ich Herrn Prof. Manfred Nagl für die interessante Aufgabenstellung, die Betreuung dieser Arbeit und die sonstige Unterstützung danken. Die Zeit war sehr schön und ausgesprochen lehrreich nicht zuletzt wegen der vielen Gespräche und seinem guten Vorbild. Außerdem gilt mein Dank Herrn Prof. Wilhelm Schäfer für die Übernahme des Zweitgutachtens zu dieser Arbeit sowie den Herren Prof. Stefan Kowalewski, Prof. Thomas Seidl und Prof. Peter Rossmanith für die Teilnahme als Vorsitzender bzw. Prüfer an meiner Doktorprüfung.

Mein besonderer Dank gilt Simon Becker, meinem Projektvorgänger, für die gute Einarbeitung und das inhaltliche Korrekturlesen dieser Arbeit.

Auch René Wörzberger möchte ich danken, dass er mir in der Endphase der Arbeit mit seinen Projekterfahrungen sehr helfen konnte. Seine Anregungen zur Darstellung der komplizierten Konzepte haben die Arbeit qualitativ verbessert.

Unterstützung hatte ich auch von meiner lieben Freundin Margit, die in kürzester Zeit die gesamte Arbeit gewissenhaft Korrektur gelesen und viele Formulierungsempfehlungen gegeben hat, ohne die der Ausdruck der Arbeit nicht so gelungen wäre. Sie war auch sonst stets für mich da, wofür ich ihr sehr dankbar bin.

Für die Vermittlung der Abläufe in verfahrenstechnischen Entwicklungsprozessen, die Anforderungsgenerierung und die Priorisierung von Funktionalitäten für Integratoren von Anwenderseite danke ich sehr Herrn Bernd Kokkelink, Produktmanager von Comos PT, sowie den Herren Walter Kattenbusch und Bernd Schneider der Firma Comos Industry Solutions GmbH für die technische Unterstützung.

Für die technische Umsetzung aber auch fachlichen Besprechungen möchte ich insbesondere den studentischen Hilfskräften und Diplomanden ganz herzlich danken. Am längsten - sogar länger als ich - war Tobias Campmann im Projekt dabei, der u. a. den Reparaturaktionengenerator realisiert hat. Es war mir immer ein besonderes Vergnügen, mit ihm zu arbeiten. Dann stießen Stefan Heukamp und Achim Drews dazu. Stefan realisierte die Dokumentenwiederverwendung, Achim baute den Dokumentenwrapper für Fließbilder in Comos. Am Ende waren Michael Brysch, Robert Schmidt und Konstantin Steinhauer für die Realisierung der Werkzeuge zuständig. Ohne ihre fleißige Unterstützung wäre die Arbeit nicht so gut verlaufen. Ich danke auch allen Beteiligten des Vorgängerprojekts, stellvertretend Marco Schmidt, der meine Konferenzbeiträge noch lange nach seiner produktiven Zeit Korrektur las.

Allen meinen Kollegen möchte ich für die schöne Zeit danken, die ich mit ihnen am Lehrstuhl hatte. Tapferer Mitstreiter aus dem Transferbereich war Thomas Heer. Ihm und Daniel Retkowitz zolle ich großen Respekt, weil sie es stets geschafft haben, auch in schwierigen Situationen die Ruhe zu bewahren. René Wörzberger und belustigte durch seinen Witz, wodurch jeder Tag am Lehrstuhl zu einem Original wurde. Mit Erhard Weinell, dem Linux-Guru, teilte ich das Büro. In gemeinsamen Lehrveranstaltungen bildeten wir das perfekte Team. Unsere Gespräche werde ich sehr ver-

missen. Mit Christof Mosler, meinem Lieblings-Immer-etwas-Neues-ausprobieren-Kollegen, habe ich während gemeinsamer Laufrunden und Besuchen im Polonia unzählige Businesspläne geschmiedet, die wir danach - vielleicht ist es auch besser so - nicht umgesetzt haben. Stephanie Mosler möchte ich an dieser Stelle dafür danken, dass sie diese Ideen immer richtig zu nehmen wusste. Thomas Haase hat mir viele Programmieretechniken beigebracht, Ulrike Ranger und Christian Fuss zeigten mir, wie man Lehrveranstaltungen organisiert.

Ibrahim Armac und Cem Mengi haben den Info-Cup ins Leben gerufen und durch den Aufbau einer sehr erfolgreichen Lehrstuhlmansschaft für ein besonders gutes Klima gesorgt, nicht zuletzt auch ihre Frauen Ebru und Azime. Zum Glück konnte ich die beiden Fußballgenies für die „andere Seite des Sports“ aufmuntern, dem Laufen. Zusammen mit René und den beiden neuen Kollegen aus Braunschweig, Jan Ringert und Martin Schindler, vertraten wir den Lehrstuhl beim Lousberglauf, für den wir gemeinsam trainiert hatten. Lautstarke Unterstützung an der Laufstrecke erhielten wir neben Thomas Heer auch von unseren jüngsten Kollegen Thomas Kurpick, Ingo Weisemöller und Arne Haber. Sie verkörpern ab nun den Lehrstuhl und vertreten ihn hoffentlich auch in folgenden Jahren bei sportlichen Wettkämpfen. Vielleicht sind dann auch die anderen „Neuen“ Claas Pinkernell, Christoph Herrmann und Christian Berger dabei. Zusammen mit Jan hatten sie die ehrenvolle Aufgabe, den Lehrstuhl von Dingen mit „historischem Wert“ zu befreien und ihn dafür mit frischem „niedersächsischem“ Wind zu bereichern. Prof. Bernhard Rumppe, dem neuen Lehrstuhlinhaber, verdanken wir fachlich gesehen neue Inspirationen und persönlich gesehen viel gute Laune, nicht zu vergessen einen neuen Kicker.

Konstanz in den Lehrstuhl brachten Marita Breuer, Galina Volkova, Angelika Fleck und Silke Cormann. Marita und Galina verkörperten das technische Wissen des Lehrstuhls der vergangenen Jahre, Angelika und Silke das organisatorische. Ihnen möchte ich sehr dafür danken, dass sie immer da waren, stets auf alle Fragen Antworten hatten und selbst bei kurzfristigen Angelegenheiten sehr hilfsbereit waren.

Danken möchte ich auch allen Mitarbeitern des Lehrstuhls vor meiner Zeit für die Entwicklung von Integratoren, PROGRES und TGG, deren Konzepte ich nutzen und erweitern durfte, stellvertretend den Profes. Andy Schürr und Albert Zündorf.

Neben den Personen aus meinem fachlichen Umfeld möchte ich all meinen Freunden danken, die mich beim Sport oder am Wochenende so schön auf andere Gedanken gebracht haben. Ohne diese Ablenkung wäre ich bestimmt nicht so motiviert wieder an die Arbeit gegangen. Schließlich danke ich meinem Freund Simon für seine humorvolle und gelassene Gesellschaft, die mir sehr geholfen hat. Von ihm habe ich die Lebensweisheiten „Et kütt wie et kütt“ und „Et hätt noch immer jot jejeange“ gelernt, die mir eine völlig neue Lebensqualität gezeigt haben.

Aachen, 21. Juli 2009

Anne-Thérèse Körtgen

Inhaltsverzeichnis

Abbildungsverzeichnis	ix
1 Einleitung	1
1.1 Konsistenzprobleme in Entwicklungsprozessen	3
1.2 Verwandte Arbeiten	6
1.2.1 Einordnung	7
1.2.2 Tripelgraphgrammatiken	10
1.2.3 Konsistenzsicherung von Modellen	12
1.2.4 Arbeiten am Lehrstuhl für Informatik 3	18
1.2.5 Integratorrahmenwerk aus dem SFB IMPROVE	23
1.3 Nicht ausreichend unterstützte Eigenschaften	36
1.3.1 Reparaturaktionen in der Modellsynchronisation	36
1.3.2 Fehlende Eigenschaften der Transformationen	38
1.3.3 Herausforderungen im industriellen Einsatz	39
1.4 Eigene Beiträge	41
1.4.1 Integrationsalgorithmus	41
1.4.2 Regelsprache	43
1.4.3 Regelerstellungsprozess	45
1.4.4 Benutzerinteraktion	46
1.4.5 Zusammenfassung	47
1.5 Aufbau der Arbeit	47
2 Szenario	49
2.1 Verfahrenstechnische Entwicklungsprozesse	50
2.2 Konsistenzsicherung von Fließbildern	53
2.2.1 Inkrementelle Transformation	54
2.2.2 Reparaturen nach Weiterbearbeitung	56
2.3 Regelinduktion und Korrespondenzanalyse	59
2.3.1 Ermittlung von Korrespondenzen bestehender Dokumente	59
2.3.2 Ermittlung von Typkorrespondenzen	62
2.4 Weitere Szenarien	65
2.5 Datenmodell der Fließbilder	66
3 Modellierung der Konsistenzregeln	69
3.1 Regelvereinfachung	69

3.2	Regelwiederverwendung	73
3.3	Operationalisierung von Regeln	78
3.4	Iterative und alternative Graphmuster	82
3.4.1	Wiederholte Teilgraphen	82
3.4.2	Alternative Teilgraphen	87
3.5	Attributbehandlungen	88
3.5.1	Inkrementvergleich	89
3.5.2	Attributeinschränkung und -propagation	92
3.6	Automatische Erstellung von Regeln	95
3.6.1	Regelerstellung aus bestehenden Dokumenten	96
3.6.2	Regelerstellung aus Dokumentenmodellen	100
3.7	Verwandte Arbeiten	111
3.7.1	Transformationssprachen	112
3.7.2	Ansätze zum Schemamatching	117
3.7.3	Generative Ansätze	120
4	Transformationen zur Konsistenzwiederherstellung	125
4.1	Konsistente Regelanwendungsstelle	127
4.2	Beschädigungs- und Reparaturarten	129
4.3	Unbeschädigter Restteilgraph	133
4.4	Reparaturaktionen	134
4.4.1	Ersetzung durch alternative Kanten	137
4.4.2	Ersetzung durch alternative Knoten	138
4.4.3	Ersetzung durch alternativen Kontext	141
4.4.4	Anwendung einer alternativen Regel	143
4.4.5	Neue Integration	144
4.4.6	Löschpropagation	144
4.4.7	Attributpropagation	146
4.4.8	Wiederherstellung der Attributbedingung	146
4.4.9	Erzeugung fehlender Knoten und Kanten	146
4.4.10	Wiederherstellung der NAC-Bedingung	147
4.4.11	Wiederherstellung der Loop-Bedingung	147
4.4.12	Verkleinerung der Regelanwendungsstelle	148
4.5	Verwandte Arbeiten	151
5	Algorithmen zur Regelausführung	153
5.1	Wiederherstellung der Korrespondenz	153
5.1.1	Einbettung in den Integrationsprozess	154
5.1.2	Reparaturaktionen generieren	156
5.1.3	Reparaturoptionen finden	156
5.1.4	Reparatur ausführen	157
5.1.5	Prioritäten & Phasen	160

5.2	Kontrollierte Regeltrigger	161
5.3	Ausführung neuer Ausdrucksmittel	168
5.3.1	Wiederholte und alternative Graphmuster	168
5.3.2	Attributgleichungen	185
5.4	Optimierte Korrespondenzanalyse	188
5.5	Verwandte Arbeiten	199
6	Realisierte Integrationsumgebung	201
6.1	Systemarchitektur	201
6.2	Erweiterungen am Integratorkern	204
6.2.1	Reparaturaktionen	206
6.2.2	Attributbehandlung	218
6.2.3	Triggerausführung	220
6.2.4	Kontextabhängigkeiten	223
6.2.5	Korrespondenzanalyse	225
6.2.6	Layoutpropagation	228
6.3	Management von Integrationsmodellen	230
6.3.1	Dokumente und Abhängigkeiten	230
6.3.2	Dokumenttypen und Abhängigkeiten	233
6.3.3	Regelmanager	234
6.3.4	Integration in Entwicklungsumgebungen	237
6.4	Unterstützung bei der Integratorenerstellung	239
6.4.1	Konfigurationswerkzeug	241
6.4.2	Regeleditor	245
6.4.3	Regelgenerator	260
6.5	Benutzerinteraktion während der Integration	263
7	Schlussbemerkungen	267
7.1	Zusammenfassung	267
7.2	Ausblick	269
	Literaturverzeichnis	273