

Informatik-Fachberichte 280

**Herausgeber: W. Brauer
im Auftrag der Gesellschaft für Informatik (GI)**

Bernhard Westfechtel

Revisions- und Konsistenzkontrolle in einer integrierten Software- entwicklungsumgebung



Springer-Verlag

Berlin Heidelberg New York London Paris
Tokyo Hong Kong Barcelona Budapest

Autor

Bernhard Westfechtel
Lehrstuhl für Informatik III, RWTH Aachen
Ahornstr. 55, W-5100 Aachen

D 82 (Diss. TH Aachen)

CR Subject Classification (1991): D.2.6, D.2.7, D.2.9, F.3.1

ISBN-13: 978-3-540-54432-6

e-ISBN-13: 978-3-642-76870-5

DOI: 10.1007/978-3-642-76870-5

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, bei auch nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Springer-Verlag Berlin Heidelberg 1991

Satz: Reproduktionsfertige Vorlage vom Autor

33/3140-543210 – Gedruckt auf säurefreiem Papier

Danksagung

Bei diesem Buch handelt es sich um die überarbeitete Fassung meiner Dissertation, die im Rahmen des IPSEN-Projekts über strukturbezogene, integrierte Softwareentwicklungsumgebungen entstanden ist und unter dem Titel "Revisionskontrolle in einer integrierten Softwareentwicklungsumgebung" an der Rheinisch-Westfälischen Technischen Hochschule Aachen angenommen wurde.

Der Dank des Autors gilt vor allen anderen dem Initiator und Leiter des IPSEN-Projekts, Prof. Dr. Nagl, der die Arbeit betreut und jederzeit mit großem Interesse verfolgt hat. Seine große Diskussionsbereitschaft und seine vielen konstruktiven Anregungen haben maßgeblich zum Gelingen dieser Arbeit beigetragen.

Ferner danke ich dem Zweitgutachter, Prof. Dr. Kreowski, der die Mühe nicht gescheut hat, für diese doch recht umfangreiche Arbeit ein kompetentes Gutachten zu erstellen.

Mein Dank gilt weiterhin allen – jetzigen und auch ehemaligen – Kollegen, die mir in vielen Diskussionen wertvolle Anregungen gegeben haben. Besonders hervorzuheben ist hier Andy Schürr, der über lange Jahre hinweg als ständiger Diskussionspartner zur Verfügung stand und bei vielen Gesprächen zwischen Tür und Angel wertvolle und konstruktive Kritik übte. Insbesondere durch die Entwicklung der Spezifikationssprache PROGRESS, die von mir extensiv benutzt wurde, hat er dazu beigetragen, daß die Arbeit eine handfeste Gestalt annehmen konnte. Weiterhin danke ich auch Claus Lewerentz, dessen Dissertation den Ausgangspunkt für meine Überlegungen bildete und der in der Anfangsphase der Arbeit als kompetenter Gesprächspartner eine überaus wertvolle Hilfe war. Ferner danke ich Martin Lefering und Albert Zündorf, die Teile der Arbeit Korrektur gelesen haben.

Darüber hinaus bin ich auch den Diplomanden und Programmiererinnen, die sich bei der Implementierung von Werkzeugen außerordentlich engagiert haben, zu großem Dank verpflichtet. Hier sind zu nennen: Michael Broekmans, Ursula Cordts, Marita Lischewski, Gerd Metzen, Peter Heimann und Cornelia Weigmann.

Schließlich danke ich meinen Eltern und meiner Schwester für die moralische Unterstützung, die sie mir in den letzten Jahren – und natürlich auch vorher – gegeben haben. Meiner Freundin Monika danke ich für das Verständnis, das sie insbesondere in der Endphase der Arbeit aufgebracht hat.

Aachen, im Juni 1991

Zusammenfassung

Dieses Buch ist im Rahmen des IPSEN-Projekts entstanden, das sich mit integrierten, strukturbezogenen Softwareentwicklungsumgebungen befaßt. Es beschäftigt sich zum einen mit der Revisionskontrolle, d.h. der Kontrolle der zeitlichen Entwicklung von Softwaredokumenten (Anforderungsdefinitionen, Architekturen, Modulimplementationen etc.). Im Verlauf ihrer Lebensgeschichte durchlaufen Dokumente bezüglich ihres Inhalts nacheinander verschiedene Zustände, weil Fehler beseitigt werden, Optimierungen vorgenommen werden etc. Solche Zustände werden als Revisionen bezeichnet. Die Revisionskontrolle ist insbesondere bei der Wartung komplexer Softwaresysteme von herausragender Bedeutung.

Zum anderen befaßt sich dieses Buch mit der Konsistenzkontrolle. Die Dokumente eines Softwaresystems sind nicht voneinander unabhängig, sondern zwischen ihnen bestehen enge inhaltliche Beziehungen. Beispielsweise werden in der Architektur eines Softwaresystems Modulschnittstellen festgelegt, die bei der Ausprogrammierung der Modulimplementationen zu beachten sind, und die technische Dokumentation eines Softwaresystems muß Anforderungsdefinition, Architektur und Modulimplementationen korrekt beschreiben. Um diese externe Konsistenz abhängiger mit bestimmenden Dokumenten zu gewährleisten, muß der Softwareentwickler durch geeignete Werkzeuge unterstützt werden. Dabei ist insbesondere zu berücksichtigen, daß die Dokumente jeweils in verschiedenen Revisionen existieren und somit verwaltet werden muß, welche Revisionen sich konsistent kombinieren lassen.

In diesem Buch wird ein formaler Ansatz zur Revisions- und Konsistenzkontrolle dargestellt, der sich durch folgende Eigenschaften auszeichnet:

- Einheitliches formales Datenmodell. Die Revisions- und Konsistenzkontrolle beruht auf einem einheitlichen formalen Datenmodell, nämlich gerichteten, markierten und attribuierten Graphen. Zur formalen Spezifikation von Werkzeugen werden programmierte Graphersetzungssysteme verwendet.
- Strukturbezogenheit. Die Werkzeuge zur Revisions- und Konsistenzkontrolle selbst arbeiten strukturbezogen, und sie sind mit anderen, ebenfalls strukturbezogenen Werkzeugen integriert.
- Dokumenttypunabhängigkeit. Die Werkzeuge zur Revisions- und Konsistenzkontrolle werden so weit wie möglich von den Typen der Dokumente unabhängig gehalten, deren Verwaltung unterstützt wird. Dadurch wird erreicht, daß die Werkzeuge sich allgemein einsetzen lassen.
- Arbeitsbereichsübergreifende Integration von Aktivitäten. Die Integration von Aktivitäten, die verschiedenen, voneinander abhängigen Arbeitsbereichen (Programmieren im Großen, Programmieren im Kleinen etc.) zuzuordnen sind, beschränkt sich nicht auf Konsistenzanalysen, sondern es werden Vorgaben erzeugt und abhängige Revisionen angepaßt, nachdem bestimmte Revisionen geändert worden sind.

Inhaltsverzeichnis

1 Motivation und Zielsetzung	1
1.1 Unterstützung der Softwareerstellung durch Werkzeuge	1
1.2 Konfigurationsverwaltung: Bedeutung und Unterstützung	5
1.3 Werkzeuge zur Revisions- und Konsistenzkontrolle in IPSEN	9
1.4 Kapitelübersicht	16
2 Ein Ansatz zur Revisions- und Konsistenzkontrolle in IPSEN	18
2.1 Modellierung der Revisions- und Konsistenzkontrolle auf der grobkörnigen Ebene	18
2.2 Modellierung der Revisionskontrolle auf der feinkörnigen Ebene	26
2.3 Modellierung der Konsistenzkontrolle auf der feinkörnigen Ebene	34
2.4 Realisierung der Revisions- und Konsistenzkontrolle	42
2.5 Zusammenfassung	48
3 Andere Ansätze zur Konfigurationsverwaltung	50
3.1 Grundbegriffe	50
3.2 Werkzeuge	63
3.3 Zusammenfassung	86
4 Modellierung der Revisions- und Konsistenzkontrolle auf der grobkörnigen Ebene	88
4.1 Modellierung der statischen Struktur	89
4.1.1 Bestandteile von Revisionsgraphen	89
4.1.2 Konsistenzbedingungen	93
4.1.3 Szenariospezifische Aspekte	105
4.1.4 Erweiterungen	106
4.2 Modellierung von Veränderungen	111
4.2.1 Szenariounabhängige Basisoperationen	113
4.2.2 Szenariounabhängige komplexe Operationen	127
4.2.3 Szenariospezifische komplexe Operationen	144
4.3 Literaturvergleich	146
4.4 Zusammenfassung und Ausblick	157
5 Verfeinerung des grobkörnigen Modells der Revisions- und Konsistenzkontrolle	159
5.1 Verfeinerung der Struktur von Revisionsgraphen	159
5.2 Verfeinerung grobkörniger Operationen auf Revisionsgraphen	166
5.2.1 Szenariounabhängige Basisoperationen	166
5.2.2 Szenariounabhängige komplexe Operationen	172
5.3 Literaturvergleich	177
5.4 Zusammenfassung und Ausblick	180

6	Modellierung der Konsistenzkontrolle auf der feinkörnigen Ebene .	181
6.1	Normierte EBNF's und abstrakte Syntaxgraphen	181
6.2	EBNF-Korrespondenzen	187
6.3	Ein inkrementeller Transformator	194
6.3.1	Ein Beispiel	195
6.3.2	Prinzipien der inkrementellen Transformation	200
6.3.3	Formale Spezifikation	204
6.4	Literaturvergleich	214
6.5	Zusammenfassung und Ausblick	219
7	Modellierung der Revisionskontrolle auf der feinkörnigen Ebene ..	221
7.1	Kontextfreies Verschmelzen	221
7.1.1	Prämissen und Prinzipien	221
7.1.2	Formale Spezifikation	227
7.1.3	Probleme	241
7.2	Kontextsensitives Verschmelzen	245
7.2.1	Prämissen und Prinzipien	245
7.2.2	Formale Spezifikation	250
7.2.3	Probleme	254
7.3	Dokumentübergreifende Aspekte des Verschmelzens	254
7.4	Literaturvergleich	260
7.5	Zusammenfassung und Ausblick	264
8	Realisierung der Revisions- und Konsistenzkontrolle	266
8.1	Der Graphenspeicher GRAS	266
8.2	Realisierung eindeutiger Knotenbezeichner und graphübergreifender Kanten	270
8.2.1	Realisierung eindeutiger Knotenbezeichner	270
8.2.2	Realisierung graphübergreifender Kanten	272
8.2.3	Literaturvergleich	282
8.3	Realisierung von Basisoperationen zur Änderungsverwaltung	284
8.3.1	Funktionalität der Basisoperationen	284
8.3.2	Realisierung der Basisoperationen	289
8.3.3	Literaturvergleich	299
8.4	Stand der Implementierung	303
8.5	Zusammenfassung und Ausblick	308
	Literaturverzeichnis	309
	Stichwortverzeichnis	319