

Erik Tittel

Aus der Reihe: e-fellows.net stipendiaten-wissen

e-fellows.net (Hrsg.)

Band 199

Refactoring in der Ontologiegetriebenen Softwareentwicklung

Ein Konzept zur Entwicklung und Evolution ontologiegetriebener
Softwaresysteme

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 2011 GRIN Verlag
ISBN: 9783640994090

Dieses Buch bei GRIN:

<https://www.grin.com/document/177623>

Erik Tittel

Aus der Reihe: e-fellows.net stipendiaten-wissen

e-fellows.net (Hrsg.)

Band 199

Refactoring in der Ontologiegetriebenen Softwareentwicklung

Ein Konzept zur Entwicklung und Evolution ontologiegetriebener Softwaresysteme

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

Diplomarbeit

Refactoring in der Ontologiegetriebenen Softwareentwicklung

bearbeitet von

Erik Tittel

Technische Universität Dresden

Fakultät Informatik
Institut für Software- und Multimediatechnik
Lehrstuhl Softwaretechnologie

Eingereicht am 03.05.2011

Vorwort

Abstract

In this thesis an approach is elaborated for the development and evolution of ontology-driven software systems. Ontology-driven software systems are software systems for which ontologies serve as main design documents. Ontologies are furthermore central parts of the running system. They describe the structure of the system and hold data. Parts of the software system are automatically derived from the structure descriptions of the ontology. This work concentrates on the evolution of those systems, thereby defining a catalogue of ontology refactorings. A tool suite called OntoMore is implemented to show the feasibility of the elaborated approach. OntoMore can transform ontologies in metamodels and models of EMF to integrate them in software systems. It can furthermore execute refactorings synchronously on both structures, which is called Co-Refactoring. Hence the consistent evolution of ontologies and models is ensured. The implementation is evaluated with an example ontology about the freelancer domain.

Abstract (deutsch)

In der vorliegenden Arbeit wird ein Konzept zur Entwicklung und Evolution ontologiegetriebener Softwaresysteme erarbeitet. Ontologiegetriebene Softwaresysteme sind Softwaresysteme, bei denen Ontologien als zentrale Designdokumente zum Einsatz kommen. Ontologien sind gleichzeitig zentrale Bestandteile des ausführbaren Systems und dienen zur Strukturbeschreibung und Datenhaltung. Dabei werden Teile des Softwaresystems automatisch aus den Strukturbeschreibungen der Ontologie abgeleitet. Diese Arbeit konzentriert sich auf die Weiterentwicklung solcher Systeme und stellt dafür einen Katalog von Ontologie-Refactorings auf. Es werden mehrere Werkzeuge, gemeinsam als OntoMore bezeichnet, implementiert, um die Umsetzbarkeit des aufgestellten Konzepts zu zeigen. OntoMore kann Ontologien in Metamodelle und Modelle des EMF umwandeln und somit in Softwaresysteme integrieren. Außerdem ist es in der Lage, Refactorings auf beiden Strukturen synchron auszuführen. Dieser Prozess wird als Co-Refactoring bezeichnet. Damit wird die konsistente Evolution von Ontologien und Modellen sichergestellt. Die Implementierung wird anhand einer Beispiel-Ontologie zum Freelancer-Management evaluiert.

Danksagung

Diese Arbeit wurde in Kooperation mit der buschmais GbR unter der Betreuung von Herrn Dipl.-Inf. Tobias Israel geschrieben.

Meinem Betreuer Tobias Israel sei an dieser Stelle ein besonderer Dank ausgesprochen. Er hat mir in vielen anregenden Diskussionen geholfen, immer auf dem richtigen Weg zu bleiben. Auch den anderen Mitarbeitern von buschmais, insbesondere Torsten Busch, Christiane Köhler und Frank Schwarz, sei für die umfangreiche Unterstützung gedankt.

Ein besonderer Dank gilt ebenso meinem Betreuer vom Lehrstuhl Softwaretechnologie Jan Reimann, der mir mit konstruktiver Kritik und vielen Verbesserungsvorschlägen stets weitergeholfen hat. Ich danke ebenfalls Christian Wende, der geduldig alle meine Fragen zu den am Lehrstuhl entwickelten Werkzeugen beantwortete.

Außerdem sei Matthew Horrdige von der University of Manchester gedankt. Durch seine Hilfe war es mir möglich, auch die kleinsten Details der Manchester OWL Syntax zu verstehen. Abhishek Shivkumar von ClearForest verdanke ich im Übrigen viele Erkenntnisse bezüglich der Abbildung von Ontologien auf Domänenmodelle.

Ich danke meinen Eltern, dass sie Ablenkungen von mir fern hielten und es mir ermöglichten, mich auf meine Arbeit zu konzentrieren. Sie standen mir mit ihren Ratschlägen stets zur Seite.

Nicht zuletzt bedanke ich mich bei allen Helfern, die mit ihren Korrekturen, Hinweisen und Verbesserungsvorschlägen sehr zum Gelingen der Arbeit beigetragen haben. Dazu zählen, abgesehen von den bereits genannten Personen: Anne Berger, Ulrich Kramer und Lisa Rothe.

Erik Tittel, Mai 2011



Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	2
1.3	Gliederung	3
2	Grundlagen	5
2.1	Modellgetriebene Softwareentwicklung	5
2.2	Refactoring	7
3	Ontologien und semantische Techniken	11
3.1	Grundlagen semantischer Techniken	11
3.2	Ontologien	14
3.2.1	Definition und Begriffe	14
3.2.2	Die Web Ontology Language (OWL)	14
3.2.3	Ontologie-Syntaxen	16
3.2.4	Beschreibungslogiken, Teilsprachen und Profile	18
3.2.5	Abfragen mit SPARQL	19
3.2.6	Beispiele für Ontologien	20
3.3	Ontologie-Evolution und Versionierung	21
3.4	Unterschiede zwischen Ontologie-Evolution und Refactoring	23
3.5	Zukünftige Entwicklungen – Linked Data	24
3.6	Zusammenfassung	25
4	Eingesetzte Werkzeuge	27
4.1	Verbindung von Ontologien und Modellen mit OntoMoPP	27
4.2	Generisches Modell-Refactoring mit Refactory	28
5	Stand der Forschung	31
5.1	Abbildung von Ontologien auf Domänenmodelle	31
5.1.1	Einsatz von Ontologien in der Modellierung	31
5.1.2	Abbildungen zwischen Ontologien und Datenbanken	35
5.2	Ontologie-Evolution	37
5.2.1	Anforderungen an Ontologie-Evolution	37
5.2.2	Elementare und komplexe Änderungsoperationen	38
5.2.3	KAON – Das Karlsruhe Ontology and Semantic Web Framework	38
5.2.4	Das NeOn-Toolkit	41
5.2.5	Protégé	45
5.2.6	Bewertung vorhandener Systeme zur Ontologie-Evolution	47
5.3	Co-Evolution von Metamodellen und Modellen	48
5.4	Zusammenfassung	50
6	Konzept zur Entwicklung und Evolution ontologiegetriebener Softwaresysteme	51
6.1	Gesamtkonzept	51

6.2	OWL-Ecore-Transformation	53
6.3	Refactorings für OWL und Ecore	62
6.3.1	Definition und Katalog von Ontologie-Refactorings	62
6.3.2	Schema eines Ontologie-Refactorings	63
6.3.3	Beispiel eines Ontologie-Refactorings	66
6.4	Co-Refactoring	67
6.4.1	Definition und Ansätze	67
6.4.2	Herleitung und Architekturvergleich	70
6.4.3	Der CoRefactorer	70
6.5	Zusammenfassung	73
7	Praktische Umsetzung	75
7.1	Architektur und eingesetzte Techniken	75
7.2	Testgetriebene Entwicklung	76
7.3	OWL-Ecore-Transformator	79
7.4	Refactoring mit Refactory	80
7.5	CoRefactorer	85
7.6	Zusammenfassung	90
8	Evaluation	91
8.1	Die Beispiel-Ontologie: FrOnto	91
8.2	Bewertung der Umsetzung	94
8.3	Grenzen der Umsetzung	97
8.4	Grenzen der Konzeption	98
8.5	Zusammenfassung	99
9	Zusammenfassung	101
9.1	Ergebnisse	101
9.1.1	Das Gesamtkonzept	101
9.1.2	Beziehung zwischen Ontologien und Domänenmodellen	101
9.1.3	Konzeption von Refactorings und Co-Refactorings	102
9.1.4	Implementierung von OntoMore	103
9.1.5	Evaluation anhand einer Beispiel-Ontologie	103
9.2	Ausblick	104
Anhang		i
A	Weitere Ontologie-Refactorings	i
B	Auswirkungen von Ontologie-Refactorings bezüglich der Datenmigration	i
C	Die MiniFrOnto vor und nach dem Refactoring	iii
D	Refactoring mit der OWL-API	v
Abbildungsverzeichnis		vii
Tabellenverzeichnis		ix
Listings		xi
Abkürzungsverzeichnis		xiii
Literaturverzeichnis		xv

1 Einleitung

1.1 Motivation

Die Welt ändert sich. Auch Softwaresysteme müssen sich ändern, denn sie stellen einen Teil der wirklichen Welt dar, die so genannte Anwendungs- oder Fachdomäne. Als Softwaresystem wollen wir hier ein Computerprogramm bezeichnen, das für den industriellen Einsatz in großen Unternehmen geeignet ist. Ein solches Programm ist universal einsetzbar, wurde getestet, dokumentiert, besteht aus mehreren Modulen und verfügt über ein klar definiertes Interface [Bro95]. Nehmen wir als Beispiel ein Softwaresystem, das für ein Unternehmen Daten über Kunden und verkaufte Produkte speichern und zurückgeben soll. Die Anforderungen des Unternehmens und der Mitarbeiter an dieses Softwaresystem werden sich im Laufe der Zeit ständig ändern. Softwaresysteme müssen demnach beständig weiterentwickelt werden. Andernfalls nimmt der Nutzen des Softwaresystems ab [Leh96]. Es ist also wichtig, während der Entwicklung und des Betriebs eines Softwaresystems Techniken einzusetzen, die Änderungen auf einfache Weise ermöglichen.

Ein Softwaresystem soll Daten speichern und dem Benutzer Informationen zur Verfügung stellen. Daten sind Repräsentationen von Eigenschaften von Objekten der realen Welt [Row07]. In der Unternehmensanwendung sind Daten zum Beispiel Name und Adressen der Kunden sowie Eigenschaften von Produkten. Das *Domänenmodell* ist eine abstrakte Beschreibung aller für das Softwaresystem relevanter Daten und deren Beziehungen und stellt damit die Struktur eines Softwaresystems dar.

Daten werden oft in Datenbanken gespeichert. Die Speicherung erfolgt dort in tabellarischer Form. Beziehungen zu Daten anderer Tabellen werden durch Schlüsselwerte ausgedrückt. Die Entwicklung des Semantic Web [BLHL01] hat zu einer Reihe neuer Techniken zur Datenspeicherung geführt, weil im Web die Daten nicht zentral gespeichert und verwaltet werden, sondern in unterschiedlicher Struktur über das ganze Web verteilt sind. Die homogene, tabelleartige Struktur der Daten ist somit nicht gegeben. Die Beziehungen zwischen den Daten treten an dieser Stelle stärker in den Vordergrund. Zu diesen neuen Techniken gehören die *Ontologien*. Ontologien sind Begriffsnetze, die eine netzartige Struktur von Daten beschreiben. Sie ermöglichen gegenüber Datenbanken einige zusätzliche Funktionen, die sie als Datenspeicher interessant machen. Dazu gehören beispielsweise die semantische Suche, Ableitung impliziten Wissens und Konsistenzprüfungen [HKRS08].

Im Rahmen dieser Arbeit wird untersucht, welche Möglichkeiten bestehen, Softwaresystemen mit Ontologien zu modellieren. Die in der Ontologie definierten Konzepte müssen folglich auf ein Domänenmodell abgebildet werden. Somit können die in der Ontologie gespeicherten Daten und Strukturen direkt in das Softwaresystem integriert werden. Dies erfordert jedoch, dass das Domänenmodell in einer Struktur vorliegt, die eine definierte Beziehung zwischen Domänenmodell und Ontologie erlaubt. Dies ist bei der *modellgetriebenen Softwareentwicklung* (MDS) der Fall. In der MDS werden Software-Systeme auf einer höheren Abstraktionsebene mit Modellen beschrieben [SVEH07]. Das Domänenmodell liegt hier explizit in Form von Metamodellen vor, die mit den Ontologien in Beziehung gesetzt werden können.

1 Einleitung

Die Herausforderung besteht nun darin, Änderungen der Struktur der Daten oder der Daten selbst möglichst einfach auf die Ontologie und das Domänenmodell zu übertragen. Für die Durchführung von Änderungen mit klar definierten Anfangs- und Endzuständen hat sich das *Refactoring* in der Softwareentwicklung etabliert. Das Prinzip des Refactorings wurde ursprünglich dazu entworfen, das Design von Programmcode durch systematische Umstrukturierungen zu verbessern. Diese Verbesserungen äußern sich beispielsweise in einer besseren Lesbarkeit oder Wartbarkeit des Programmcodes. Martin Fowler hat für objektorientierte Computerprogramme einen allgemein anerkannten Katalog von Refactorings aufgestellt [FBB⁺99]. Dabei beschreibt ein Refactoring eine Folge von Änderungen, die in ihrer Gesamtheit zu einer wohldefinierten Umstrukturierung führen. Außerdem existiert mit *Refactory* bereits ein Framework, welches Refactorings auf beliebigen Modellen durchführen kann [RSA10]. Nun soll das Prinzip des Refactorings auch auf Ontologien übertragen werden, um Änderungen systematisch durchzuführen. An dieser Stelle stehen wir vor dem Problem, dass Refactorings auf zwei zusammengehörigen Strukturen – Ontologie und Domänenmodell – synchron ausgeführt werden müssen. Um dieses Problem zu lösen, wird das Prinzip des *Co-Refactorings* eingeführt. Dies ermöglicht die Synchronisation von Refactorings auf zwei unterschiedlichen Strukturen. Eine Änderung der Ontologie oder des Domänenmodells führt somit zu einer Änderung der jeweils anderen Struktur, sodass die Beziehung zwischen beiden Strukturen konsistent bleibt.

1.2 Zielsetzung

Ziel der Arbeit ist es, zu untersuchen, inwieweit Ontologien zum Design von Softwaresystemen eingesetzt werden können. Dazu werden vier Teilbereiche bearbeitet. Zuerst ist die Beziehung zwischen Ontologien und Domänenmodellen zu untersuchen. Es soll festgestellt werden, in welchem Umfang Ontologien auf Domänenmodelle abgebildet werden können und wie eine entsprechende Transformation formalisiert werden kann. Im Rahmen des MOST-Projekts¹ wurde bereits eine Abbildung von Domänenmodell auf Ontologie erarbeitet. Diese kann als Ausgangspunkt für die Erarbeitung einer Abbildung in die andere Richtung dienen.

Des Weiteren sind Refactorings für Ontologien und Domänenmodelle zu entwerfen. Diese Refactorings stellen formale Änderungsoperationen dar, die es ermöglichen, Ontologien und Domänenmodelle von einem definierten Zustand in einen anderen definierten Zustand zu versetzen. Die Definition der Ontologie-Refactorings kann sich dabei an vorhandenen Arbeiten zur Ontologie-Evolution orientieren [NK04, GPS10]. Darauf aufbauend soll ein Konzept für Co-Refactorings aufgestellt werden, mit dem Refactorings auf mehreren Modellen synchron ausgeführt werden können. Refactory als Werkzeug zum generischen Modell-Refactoring soll dabei zur Umsetzung des Co-Refactorings eingesetzt werden [RSA10].

Die Implementierung eines Prototyps soll die Umsetzbarkeit der erarbeiteten Konzepte demonstrieren. Der Prototyp umfasst zwei Programmteile. Der erste Teil ermöglicht eine Transformation von Ontologien in Domänenmodelle, während der zweite Teil Co-Refactorings auf beiden Strukturen durchführt.

Die erarbeiteten Konzepte werden schließlich anhand einer Beispiel-Ontologie evaluiert, welche die Grundlage für ein Softwaresystem zum Freelancer-Management darstellt. Die Arbeit beschränkt sich dabei auf die Erstellung der Ontologie. Die Erstellung des Softwaresystems und die Anbindung der Ontologie daran liegen außerhalb des Rahmens der Arbeit.

¹<http://www.most-project.eu>

1.3 Gliederung

Die vorliegende Arbeit gliedert sich in neun Kapitel. Im Kapitel 2 werden mit der MDSD und dem Refactoring die Grundlagen dieser Arbeit behandelt. Das Kapitel 3 stellt die Grundlagen zu Ontologien und semantischen Techniken vor. Es behandelt darüber hinaus die Ontologie-Evolution und die Unterschiede zwischen Ontologie-Evolution und Refactoring. Im Anschluss an diese theoretischen Betrachtungen werden im Kapitel 4 mit OntoMoPP und Refactory die in dieser Arbeit verwendeten Werkzeuge vorgestellt. Das Kapitel 5 stellt ausführlich den aktuellen Stand der Forschung bezüglich der Ontologie-Domänenmodell-Abbildung, der Ontologie-Evolution und der Co-Evolution dar. Der Schwerpunkt dieser Arbeit, das Konzept zur Entwicklung und Evolution ontologiegetriebener Softwaresysteme, wird in Kapitel 6 präsentiert. Das Kapitel 7 zeigt die praktische Umsetzung dieses Konzepts. Anschließend erfolgt im Kapitel 8 die Evaluation der implementierten Refactorings und Werkzeuge anhand einer Beispiel-Ontologie. Zum Schluss fasst das Kapitel 9 die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf mögliche zukünftige Entwicklungen.