MontiCore: Efficient Development of Composed Modeling Language Essentials

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

vorgelegt von

Diplom-Informatiker
Pedram Mir Seyed Nazari
aus Teheran, Iran

Berichter: Universitätsprofessor Dr. Bernhard Rumpe
Universitätsprofessor Dr. Uwe Aßmann

Tag der mündlichen Prüfung: 6. Februar 2017

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 29

Pedram Mir Seyed Nazari
RWTH Aachen University

MontiCore: Efficient Development of Composed Modeling Language Essentials

Shaker Verlag
Aachen 2017

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: D 82 (Diss. RWTH Aachen University, 2017)

Printed in Germany.

ISBN 978-3-8440-5320-3
ISSN 1869-9170

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

Model-driven engineering exploits models as first-class artifacts to tackle the complexity and heterogeneity of large software systems. Such models can be built with domain-specific languages (DSLs) that enable higher-levels of abstraction and also facilitate separation of concerns and reuse. Unlike general-purpose languages, a DSL allows specification of system aspects by using the terminology of the domain. This, in turn, enables domain experts—who rarely have software engineering skills—to become directly involved in the development process.

In textual, grammar-based languages, an abstract syntax tree (AST) technically represents the model within a tool and thus serves as the central artifact for model processing, e.g., for static analysis or code generation. Deriving the AST directly from the grammar keeps it consistent with the concrete syntax [KRV07b] and, hence, reduces maintenance efforts. Thus, the AST greatly depends on the grammar in terms of both content and structure, leading to two drawbacks: First, since the AST does not necessarily provide a model's essential information (e.g., the element a name refers to) in a straightforward manner, this complexity can hamper the development of tools processing that model. Second, even small grammar changes can affect the AST that, in turn, may require dependent tools to be updated.

Moreover, since heterogeneous languages are typically required to specify different aspects of a software system, another problem is that models of those languages first have to be integrated before they can be analyzed and synthesized together [DBC+15]. Therefore, each model has to have an interface to enable composition with other models [HR13]. The composition can include models defined within the same language as well as models of independent, heterogeneous languages. Again, the AST does not explicitly exhibit a model's interface but instead mixes it with other, unessential information.

To address the above issues, this dissertation aims to promote the development of an additional structure (called ST) which captures (i) information that is essential for processing models of a language as well as (ii) a language's interface to enable composition of models of both the same language and heterogeneous languages. Unlike the (generated) AST, the ST can also contain information that is not directly defined in the model but related to it (e.g., all states that are reachable from a specific state of an automaton). Tools can employ the ST together with the AST to access the relevant information as needed, which facilitates model processing.

An additional structure, however, requires development effort itself. To further an efficient and effective development of STs, this dissertation presents a generic infrastructure with reasonable defaults. Moreover, the infrastructure provides generic concepts and mechanisms for defining model interfaces (as part of the respective STs) to allow an efficient composition of heterogeneous models via their interfaces in a non-invasive way. Thus, the models can be reused in various contexts. Furthermore, based on the

generic infrastructure, methods and patterns for developing language-specific STs are elaborated. To reduce the amount of handcrafted boilerplate code as well as the number of programming errors, a generative approach is employed, which produces parts of the language-specific ST infrastructure and provides ways for efficiently extending and customizing it. The generated ST targets essential elements of a language which are unlikely to change.

Ultimately, the proposed ST approach (including its concepts and methods) simplifies the development of tools for model processing by explicitly providing the essential information of a model. In addition, the maintenance of such tools will be improved, since, compared to the AST, the ST is more robust against changes in the grammar. Finally, the proposed ST approach provides model interfaces more explicitly, thereby facilitating the composition of models even from heterogeneous languages.

Kurzfassung

In der Modell-getriebenen Softwareentwicklung werden Modelle als Kernartefakte verwendet, um die Komplexität und Heterogenität von großen Softwaresystemen zu beherrschen. Dabei können die Modelle mittels Domänen-spezifischer Sprachen (DSLs) erstellt werden, welche eine höhere Abstraktion ermöglichen und außerdem die Trennung unterschiedlicher Aspekte und deren Wiederverwendung erleichtern. Anders als Allzweck-Programmiersprachen, können die Systemaspekte mithilfe einer DSL in der Terminologie der jeweiligen Domänen spezifiziert werden. Auf diese Weise können Domänenexperten, welche selten Erfahrungen in Softwareentwicklung haben, direkt in den Entwicklungsprozess involviert werden.

In textuellen, Grammatik-basierten Sprachen wird ein Modell technisch durch einen abstrakten Syntaxbaum (AST) innerhalb eines Werkzeugs dargestellt. Der AST dient dabei als zentrales Artefakt für die Modellverarbeitung, zum Beispiel für statische Analysen oder für die Code-Generierung. Wird der AST direkt aus der Grammatik generiert, bleibt er konsistent mit der konkreten Syntax [KRV07b] und reduziert so den Wartungsaufwand. Als Konsequenz ist der AST sowohl inhaltlich als auch strukturell stark von der Grammatik abhängig, was zwei Nachteile mit sich bringt: Zum einen liefert der AST die essentiellen Informationen eines Modells (z.B. das Element, welches durch einen Namen referenziert wird) nicht notwendigerweise in einer gut aufbereiteten Form, was die Entwicklung von Werkzeugen zur Verarbeitung des Modells erschweren kann. Zum anderen können sogar kleinere Änderungen an der Grammatik den AST betreffen, was wiederum eine Anpassung der davon abhängigen Werkzeuge erfordern kann.

Um die verschiedenen Aspekte eines Softwaresystems zu spezifizieren, werden typischerweise heterogene Sprachen benötigt. Dabei müssen die Modelle dieser Sprachen erst integriert werden bevor sie gemeinsam analysiert und synthetisiert werden können [DBC+15]. Dazu muss jedes Modell eine Schnittstelle zur Verfügung stellen, um eine Komposition mit anderen Modellen zu ermöglichen [HR13]. Eine solche Komposition kann sowohl Modelle derselben Sprache als auch Modelle aus unabhängigen, heterogenen Sprachen beinhalten. Der AST ist dafür nur teilweise geeignet, da er die Schnittstelle eines Modells nicht explizit zur Verfügung stellt, sondern diese mit anderen, unwesentlichen Informationen vermischt.

Das Ziel dieser Dissertation ist es die Entwicklung einer zusätzlichen Struktur (ST) voranzutreiben, welche (i) Informationen erfasst, die essentiell für die Verarbeitung von Modellen einer Sprache sind und (ii) die Schnittstelle einer Sprache zur Verfügung stellt, um eine Komposition von Modellen aus derselben als auch aus heterogenen Sprachen zu ermöglichen. Im Gegensatz zum (generierten) AST kann die ST auch Informationen enthalten, die nicht direkt im Modell definiert diesem aber zugehörig sind (z.B. alle erreichbaren Zustände, ausgehend von einem bestimmten Zustand eines Automaten). Um die Verarbeitung von Modellen und somit den Zugriff auf die benötigten relevanten

Informationen zu vereinfachen, kann die ST zusammen mit dem AST von Werkzeugen verwendet werden.

Des Weiteren präsentiert diese Dissertation eine generische Infrastruktur mit umfangreichen Standardimplementierungen, um die effiziente und effektive Entwicklung von STs zu unterstützen. Die Infrastruktur stellt generische Konzepte und Mechanismen für die Erstellung von Modellschnittstellen (als Teil der zugehörigen STs) zur Verfügung, um eine effiziente, nicht-invasive Komposition von heterogenen Modellen über diese Schnittstellen zu ermöglichen. Somit können die Modelle in verschiedenen Kontexten wiederverwendet werden. Darüber hinaus werden basierend auf der generischen Infrastruktur Methodiken und Entwurfsmuster für die Entwicklung von sprachspezifischen STs ausgearbeitet. Um die Menge von handgeschriebenem Boilerplate Code und auch die Anzahl von Programmierfehlern zu reduzieren, wird ein generativer Ansatz angewendet, welcher Teile einer sprachspezifischen ST generiert und Möglichkeiten bietet diese Teile effizient zu erweitern und anzupassen. Die generierte ST fokussiert die essentiellen Elemente einer Sprache, welche selten verändert werden.

Zusammengefasst vereinfacht der in dieser Arbeit vorgestellte Ansatz (mit den Konzepten und Methodiken) die Entwicklung von Werkzeugen zur Modellverarbeitung, indem die ST die essentiellen Informationen eines Modells explizit zur Verfügung stellt. Des Weiteren wird die Wartung solcher Werkzeuge verbessert, da die ST (im Vergleich zum AST) robuster im Hinblick auf Grammatikänderungen ist. Zusätzlich wird die Komposition von Modellen aus heterogenen Sprachen vereinfacht, da die ST Modellschnittstellen expliziter zur Verfügung stellt.

Danksagung

An dieser Stelle möchte ich mich bei den lieben Menschen bedanken, die mich während meiner Promotion unterstützt und so zum Erfolg dieser Dissertation beigetragen haben.

Mein ganz besonderer Dank gilt meinem Doktorvater Prof. Dr. Bernhard Rumpe für die spannende und lehrreiche Zeit am Lehrstuhl, für die ideenreichen und konstruktiven Diskussionen, für das in mich gesetzte Vertrauen zur Leitung der Modellierungsgruppe und für die Möglichkeit an zahlreichen herausfordernden Industrie- und Forschungsprojekten mitarbeiten zu können. Mit seinen Ratschlägen und weitsichtigen Anregungen hat Bernhard maßgeblich zum Erfolg dieser Arbeit beigetragen.

Prof. Dr. Uwe Aßmann danke ich herzlich für sein Interesse an meiner Arbeit und die Übernahme des Zweitgutachtens. Ich möchte mich ebenfalls bei Prof. Dr. Ulrik Schroeder für die Leitung meines Promotionskomitees und bei Prof. Dr. Erika Ábrahám für die Durchführung der Theorieprüfung bedanken. Darüber hinaus möchte ich mich herzlich bei Prof. Dr. Manfred Nagl für die zahlreichen interessanten Gespräche und hilfreichen Denkanstöße bedanken. Sein breites und detailreiches Wissen weit über die Informatik hinaus wird mich immer faszinieren.

Dr. Martin Schindler danke ich sehr für das anfängliche Mentoring sowie für seine fachliche und freundschaftliche Unterstützung während meiner Promotion. Ebenfalls danke ich Antonio "Toni" Navarro Pérez, der mit seiner herausragenden fachlichen Kompetenz zur Findung meines Forschungsthemas beigetragen hat. Sehr gerne denke ich an unsere philosophischen Gedankenexperimente zurück, die oft endeten wie sie anfingen: mit einem lachenden "Why?". Bei Michael von Wenckstern bedanke ich mich für unsere einzigartige Zeit als "digital Transformers" sowie für die zahlreichen und sehr spannenden Diskussionen über die unterschiedlichsten Themen. Alexander "Alex" Roth danke ich für die vielen Gespräche frühmorgens noch bevor der Lehrstuhl-Alltag begonnen hatte und für die gute Zusammenarbeit in Lehre und Forschung. Die Paper-Marathons gehören zu den Highlights meiner Zeit am Lehrstuhl. Dimitri Plotnikov bin ich dankbar für sein kontinuierliches Feedback zu MontiCore und der Symboltabelle. Danke auch an Klaus Müller für die spannenden Tischtennis Matches und die gute Zeit auf der Konferenz in Frankreich zusammen mit Alex. Vielen Dank an Deni Raco, der mithilfe einer Vielzahl von Kicker-, Poker- und Schachturnieren stets für einen guten Ausgleich gesorgt hat.

Darüber hinaus möchte ich mich ganz herzlich beim MontiCore Team bedanken, angefangen bei Robert Heim, mit dem ich zusammen die Planung und Analyse für das MontiCore Reengineering durchgeführt habe. Robert gehört zu den strukturiertesten und talentiertesten Menschen, die ich kennengelernt habe. Es war mir stets eine große Freude mit ihm zusammenzuarbeiten. Auch danke ich Katrin Hölldobler für die großartige Zusammenarbeit und ihr gutes Auge für Details. Andreas Horst hat leidenschaftlich dafür gesorgt, dass wir stets eine stabile und aktuelle Build-Infrastruktur hatten. Vielen Dank dafür. Des Weiteren danke ich Sebastian Oberhoff für die gute Unterstützung im

MontiCore Team. Ein sehr großer Dank gebührt Marita Breuer und Galina Volkova, die es mit ihrer Hilfsbereitschaft und der Übernahme diversester Aufgaben ermöglicht haben, MontiCore zum geplanten Zeitpunkt und in guter Qualität zu releasen. Ich danke ebenfalls den SE-Runners Michael, Alex und unserem Halbmarathonläufer Evgeny Kusmenko für die sportliche Abwechslung nach langen Diss-Tagen und das anschließende Entspannen bei einem guten Essen. Außerdem danke ich Kai Adam, Vincent Bertram, Lennart Bucher, Arvid Butting, Robert Eikermann, Timo Greifenberg, Dr. Arne Haber, Lars Hermerschmidt, Dr. Christoph Herrmann, Steffi Kaiser, Oliver Kautz, Carsten Kolassa, Thomas Kurpick, Achim Lindt, Dr. Markus Look, Matthias Markthaler, Dr. Cem Mengi, Dr. Claas Pinkernell, Manuel Pützer, Dr. Dirk Reiß, Dr. Jan Oliver Ringert, Christoph Schulze, Philipp Schütze, Brian Sinkovec, Max Voß und Dr. Andreas Wortmann für das exzellente Arbeitsklima und die spannende Zeit am Lehrstuhl. Ein großer Dank gebührt ebenfalls den zahlreichen Studenten, die durch ihre Abschlussarbeiten oder ihre Tätigkeiten als studentische Hilfskräfte zur Weiterentwicklung und Evaluation meiner Forschungsthemen beigetragen haben. Dazu gehören unter anderem: Odgerel Boldbaatar, Abdullah Celik, Peter Damm, Martin Hackenberg, Christoph Hommelsheim, Michael Krein, Markus Lüger, Mirko Mades, David Mularski, Stefan Nessel, Bao-Loc "Fe" Nguyen Ngo, Patrick Schlesiona, Simon Siewert und Alex Tritthart. Auch möchte ich Sylvia Gunder und Gabriele Heuschen danken, dass sie für einen angenehmen und reibungslosen Lehrstuhlbetrieb gesorgt haben. Vielen Dank an Robert, Katrin, Alex, Timo, Deni und Michael für das Korrekturlesen früher Fassungen der Dissertation.

Schließlich gilt mein unendlicher Dank meiner Familie. Meinen Eltern, Soudabeh und Mahmoud Nazari, danke ich von ganzem Herzen, dass Sie mich in jeder Situation meines Lebens unterstützt und immer alles daran gesetzt haben, dass ich meine Träume verwirklichen kann. Ohne Euch wäre diese Arbeit nicht möglich gewesen. Ich hoffe, dass ich Euch hiermit etwas zurückgeben konnte. Meinen Geschwistern Misam, Pegah und Navid Nazari danke ich für ihre endlose Unterstützung auf meinem Lebensweg und für die vielen schönen gemeinsamen Erlebnisse. Meiner süßen Nichte Nika danke ich, dass sie uns ohne Worte und mit nur einem Lächeln so viel Glück und Freude bereitet und unser Leben so sehr bereichert. Darüber hinaus danke ich meinen beiden Onkeln Siawash und Siamak Khajehi-Mahabadi, dass sie mir immer mit ihrer Erfahrung mit Rat und Tat zur Seite standen. Ich danke meinen Schwiegereltern Fereba und Nazier Chaled für ihre liebevolle Fürsorge und dass sie mich immer wie einen Sohn behandelt und an mich geglaubt haben. Meinen Schwägerinnen Shohreh Talebi und Natascha Chaled danke ich für ihre Unterstützung und die vielen tollen gemeinsamen Gespräche. Schließlich gebührt mein ganz besonderer Dank meiner wundervollen Frau Nilofar Nazari. Mit ihrer uneingeschränkten Unterstützung und Liebe und ihrer unglaublichen Geduld hat sie mich stets gestärkt und dadurch maßgeblich zum Gelingen dieser Arbeit beigetragen.

Aachen, April 2017
Pedram M. S. Nazari

Contents