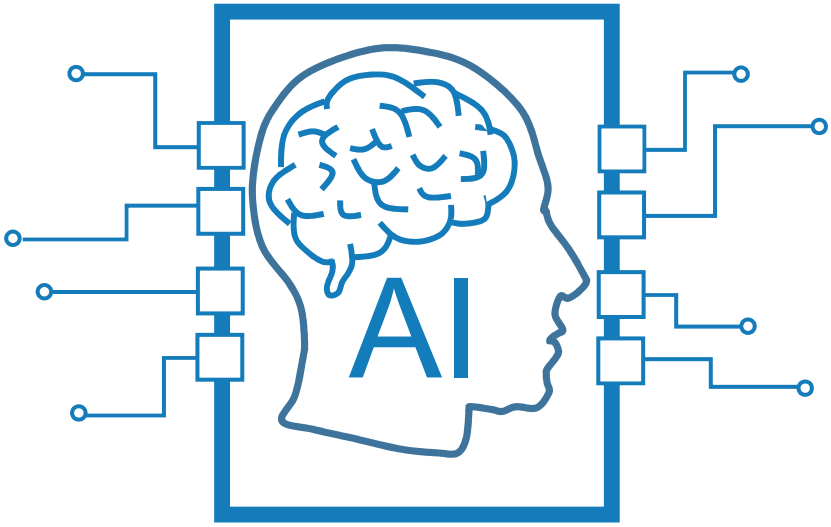
SE | RWTH AACHEN UNIVERSITY

Evgeny Kusmenko

Model-Driven Development Methodology and Domain-Specific Languages for the Design of Artificial Intelligence in Cyber-Physical Systems



EMADL

Aachener Informatik-Berichte,
Software Engineering

Band 49

Hrsg: Prof. Dr. rer. nat. Bernhard Rumpe

Model-Driven Development Methodology and Domain-Specific Languages for the Design of Artificial Intelligence in Cyber-Physical Systems

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

Dipl.-Ing.
Evgeny Kusmenko
aus Odessa, Ukraine

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

Tag der mündlichen Prüfung: 19. Juli 2021

D 82 (Diss. RWTH Aachen University, 2021)

Aachener Informatik-Berichte, Software Engineering

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

Band 49

Evgeny Kusmenko
RWTH Aachen University

Model-Driven Development Methodology and Domain-Specific Languages for the Design of Artificial Intelligence in Cyber-Physical Systems

Shaker Verlag
Düren 2021

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, 2021)

Copyright Shaker Verlag 2021
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-8286-9
ISSN 1869-9170

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9
Internet: www.shaker.de • e-mail: info@shaker.de

Eidesstattliche Erklärung

Ich, Evgeny Kusmenko, erkläre hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden. Hiermit erkläre ich an Eides statt

1. Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;

2. Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;

3. Wenn immer andere eigene oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;

4. Wenn aus anderen eigenen oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;

5. Alle wesentlichen Quellen von Unterstützung wurden benannt;

6. Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;

7. Ein Teil oder Teile dieser Arbeit wurden zuvor veröffentlicht und zwar in:

   - [KPRS19] E. Kusmenko, S. Pavlitskaya, B. Rumpe, S. Stüber: On the Engineering of AI-Powered Systems. In: ASE'19. Software Engineering Intelligence Workshop (SEI'19), San Diego, pp. 126-133, IEEE, Nov. 2019.
   - [KNP+19] E. Kusmenko, S. Nickels, S. Pavlitskaya, B. Rumpe, T. Timmermanns: Modeling and Training of Neural Processing Systems. In: Conference on Model Driven Engineering Languages and Systems (MODELS'19), Munich, pp. 283-293, IEEE, Sep. 2019.
   - [GKR19] N. Gatto, E. Kusmenko, B. Rumpe: Modeling Deep Reinforcement Learning Based Architectures for Cyber-Physical Systems. In: Proceedings of MODELS 2019. Workshop MDE Intelligence, Munich, pp. 196-202, IEEE, Sep. 2019.
   - [KKRZ19] J. C. Kirchhof, E. Kusmenko, B. Rumpe, H. Zhang: Simulation as a Service for Cooperative Vehicles. In: Proceedings of MODELS 2019. Workshop MASE, Munich, pp. 28-37, IEEE, Sep. 2019.

- [KKMR19] J. C. Kirchhof, E. Kusmenko, J. Meurice, B. Rumpe: Simulation of Model Execution for Embedded Systems. In: Proceedings of MODELS 2019. Workshop MLE, Munich, pp. 331-338, IEEE, Sep. 2019.
- [KKR19] N. Kaminski, E. Kusmenko, B. Rumpe: Modeling Dynamic Architectures of Self-Adaptive Cooperative Systems. In: The Journal of Object Technology, 18(2), pp. 1-20, AITO, July 2019.
- [HKKR19] A. Hellwig, S. Kriebel, E. Kusmenko, B. Rumpe: Component-based Integration of Interconnected Vehicle Architectures. In: 30th Intelligent Vehicles Symposium (IV'19). Workshop on Cooperative Interactive Vehicles, Paris, pp. 146-151, IEEE, June 2019.
- [KKRS19] S. Kriebel, E. Kusmenko, B. Rumpe, I. Shumeiko: Learning Error Patterns from Diagnosis Trouble Codes. In: 30th Intelligent Vehicles Symposium (IV'19). Workshop on Unsupervised Learning for Automated Driving, Paris, pp. 246-251, IEEE, June 2019.
- [DGH+19] I. Drave, T. Greifenberg, S. Hillemacher, S. Kriebel, E. Kusmenko, M. Markthaler, P. Orth, K. S. Salman, J. Richenhagen, B. Rumpe, C. Schulze, M. von Wenckstern, A. Wortmann: SMArDT Modeling for Automotive Software Testing. In: Software: Practice and Experience, 49(2):301-328, Wiley Online Library, Feb. 2019.
- [FIK+18] C. Frohn, P. Ilov, S. Kriebel, E. Kusmenko, B. Rumpe, A. Ryndin: Distributed Simulation of Cooperatively Interacting Vehicles. In: International Conference on Intelligent Transportation Systems (ITSC'18), pp. 596-601. IEEE, Hawaii, 2018.
- [KRSvW18a] E. Kusmenko, B. Rumpe, S. Schneiders, M. von Wenckstern: Highly-Optimizing and Multi-Target Compiler for Embedded System Models: C++ Compiler Toolchain for the Component and Connector Language EmbeddedMontiArc. In: Conference on Model Driven Engineering Languages and Systems (MODELS'18), pp. 447-457, Copenhagen, ACM, Oct. 2018.
- [KRSvW18b] E. Kusmenko, B. Rumpe, I. Strepkov, M. von Wenckstern: Teaching Playground for C&C Language EmbeddedMontiArc. In: Proceedings of MODELS 2018. Workshop ModComp, Copenhagen, Oct. 2018.
- [KRRvW18] E. Kusmenko, J. Ronck, B. Rumpe, M. von Wenckstern: EmbeddedMontiArc: Textual Modeling Alternative to Simulink. In: Proceedings of MODELS 2018. Workshop EXE, Copenhagen, Oct. 2018.
- [BKL+18] C. Brecher, E. Kusmenko, A. Lindt, B. Rumpe, S. Storms, S. Wein, M. von Wenckstern, A. Wortmann: Multi-Level Modeling Framework for Machine as a Service Applications Based on Product Process Resource Models.

International Symposium on Computer Science and Intelligent Control (ISCSIC'18). Stockholm, ACM, Sep. 2018.

- [KKRvW18] S. Kriebel, E. Kusmenko, B. Rumpe, M. von Wenckstern: Finding Inconsistencies in Design Models and Requirements by Applying the SMARDT Process. In: Tagungsband des Dagstuhl-Workshop MBEES: Modellbasierte Entwicklung eingebetteter Systeme XIV (MBEES'18). Univ. Hamburg, Apr. 2018.
- [KSRvW18] E. Kusmenko, I. Shumeiko, B. Rumpe, M. von Wenckstern: Fast Simulation Preorder Algorithm. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD'18), pp. 256-267. Funchal, Portugal, SciTePress, Jan. 2018.
- [HKK+18] S. Hillemacher, S. Kriebel, E. Kusmenko, M. Lorang, B. Rumpe, A. Sema, G. Strobl, M. von Wenckstern: Model-Based Development of Self-Adaptive Autonomous Vehicles using the SMARDT Methodology. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD'18), pp. 163-178. Funchal, Portugal, SciTePress, Jan. 2018.
- [GKR+17] F. Grazioli, E. Kusmenko, A. Roth, B. Rumpe, M. von Wenckstern: Simulation Framework for Executing Component and Connector Models of Self-Driving Vehicles. In: Proceedings of MODELS 2017. Workshop EXE, Austin, CEUR 2019, Sept. 2017.
- [KRRvW17] E. Kusmenko, A. Roth, B. Rumpe, M. von Wenckstern: Modeling Architectures of Cyber Physical Systems. In: European Conference on Modelling Foundations and Applications (ECMFA'17), Marburg, pp. 34-50. LNCS 10376, Springer, July 2017.
- [DDE+17] J. Dankert, C. Dernehl, L. Eckstein, S. Kowalewski, E. Kusmenko, B. Rumpe: RapidCoop - Robuste Architektur durch geeignete Paradigmen für Kooperativ Interagierende Automobile. In: Automatisiertes und Vernetztes Fahren (AAET'17), Feb. 2017.

Aachen, den 27.04.2021

Evgeny Kusmenko

Abstract

The development of cyber-physical systems poses a multitude of challenges requiring experts from different fields. Such systems cannot be developed successfully without the support of appropriate processes, languages, and tools. Model-driven software engineering is an important approach which helps development teams to cope with the increasing complexity of today's cyber-physical systems. The aim of this thesis is to develop a model-driven engineering methodology with a particular focus on interconnected intelligent cyber-physical systems such as cooperative vehicles.

The basis of the proposed methodology is a component-and-connector architecture description language focusing on the decomposition and integration of cyber-physical system software. It features a strong, math-oriented type system abstracting away from the technical realization and incorporating physical units. To facilitate the development of highly-interconnected self-adaptive systems, the language enables its users to model component and connector arrays and supports architectural runtime-reconfiguration. Architectural elements can be altered, added, and removed dynamically upon the occurrence of trigger events.

In order to fully cover the development process, the proposed methodology, in addition to structural modeling, provides means for behavior specification and its seamless integration into the components of the architecture. A matrix-oriented scripting language enables the developer to specify algorithms using a syntax close to the mathematical domain. What is more, a dedicated deep learning modeling language is provided for the development and training of neural networks as directed acyclic graphs of neuron layers. The framework supports different learning methods including supervised, reinforcement, and generative adversarial learning, covering a broad range of applications from image and natural language processing to decision making and test data generation.

The presented toolchain enables an automated generation of fully functional C++ code together with the corresponding build and training scripts based on the architectural models and behavior specifications. Finally, to facilitate the integration and deployment of the modeled software in distributed environments, we use a tagging approach to model the middleware and to control a middleware generation toolchain.

Kurzfassung

Die Entwicklung cyber-physischer Systeme stellt eine Vielzahl von Herausforderungen und erfordert Experten aus unterschiedlichen Bereichen. Solche Systeme können nicht ohne die Unterstützung durch geeignete Prozesse, Sprachen und Tools erfolgreich entwickelt werden. Modellgetriebenes Software Engineering stellt einen wichtigen Ansatz dar, der Entwicklungsteams hilft, die zunehmende Komplexität heutiger cyber-physischer Systeme zu bewältigen. Das Ziel dieser Arbeit besteht darin, eine modellgetriebene Engineering-Methodik mit besonderem Fokus auf vernetzte intelligente cyber-physische Systeme wie kooperative Fahrzeuge zu entwickeln.

Die Grundlage der vorgestellten Methodik bildet eine Komponenten- und Konnektoren-basierte Architekturbeschreibungssprache zur Dekomposition und Integration von Software für cyber-physische Systeme. Diese verfügt über ein starkes statisches, mathematisch orientiertes Typsystem, welches physikalische Einheiten unterstützt und von der technischen Realisierung abstrahiert. Um die Entwicklung hochvernetzter selbstadaptiver Systeme zu erleichtern, ermöglicht die Sprache die Modellierung von Komponenten- und Konnektorarrays und unterstützt Laufzeit-Rekonfigurationen der Architektur. Architekturelemente können dabei ereignisbasiert dynamisch geändert, hinzugefügt und entfernt werden.

Um den Entwicklungsprozess vollständig abzudecken, bietet die vorgestellte Methodik neben der strukturellen Modellierung Mittel zur Verhaltensspezifikation und deren nahtlose Integration in die Komponenten der Architektur. Eine matrixorientierte Skriptsprache ermöglicht es dem Entwickler, Algorithmen in einer Syntax zu spezifizieren, die der mathematischen Domäne sehr nahe kommt. Darüber hinaus wird eine dedizierte Deep-Learning-Modellierungssprache für die Entwicklung und das Training von neuronalen Netzen in Form von azyklischen, aus Neuronenschichten bestehenden Graphen bereitgestellt. Das Framework unterstützt verschiedene Lernmethoden wie überwachtes, verstärkendes sowie GAN-basiertes Lernen und deckt damit ein breites Anwendungsspektrum von der Bild- und natürlichen Sprachverarbeitung bis hin zur Entscheidungsfindung und Testdatengenerierung ab.

Auf Basis der Architekturmodelle und Verhaltensspezifikationen erlaubt die vorgestellte Toolchain eine automatisierte Generierung von voll funktionsfähigem C++-Code zusammen mit den entsprechenden Build- und Trainingsskripten. Um die Integration und Bereitstellung der modellierten Software für verteilte Umgebungen zu erleichtern, verwenden wir schließlich einen Tagging-Ansatz zur Modellierung und Generierung von Middleware.

Acknowledgements

Ich möchte mich ganz herzlich bei meinem Doktorvater, Prof. Dr. Bernhard Rumpe, bedanken, der mir die Möglichkeit gegeben hat, am Lehrstuhl für Software Engineering der RWTH Aachen University zu promovieren und mich wissenschaftlich, aber auch persönlich zu entfalten. Für die zahlreichen Diskussionen, Ideen und Sichtweisen. Und für die interessanten Aufgaben, spannenden Projekte und Herausforderungen, an denen ich arbeiten durfte und die immer Lust auf mehr gemacht haben.

Ich danke Prof. Dr. Aßmann für die Zweitbegutachtung meiner Dissertation, aber auch dafür, während meiner Studienzeit an der TU Dresden als Dozent mein besonderes Interesse für das Gebiet Software Engineering geweckt zu haben. Des Weiteren bedanke ich mich bei Prof. Dr. Leibe für die Leitung meines Prüfungskomittees sowie bei Prof. Dr. Erika Ábrahám für die Abnahme der mündlichen Prüfung im Bereich der theoretischen Informatik.

Ganz besonders möchte ich mich auch bei meinen Kolleginnen und Kollegen vom Lehrstuhl für Software Engineering für eine produktive und harmonische Zusammenarbeit, interessante Diskussionen und eine schöne Zeit am Lehrstuhl bedanken. Mein Dank gilt Dr. Michael von Wenckstern für die enge Zusammenarbeit an EmbeddedMontiArc und vielen gemeinsamen Papieren, vor allem aber auch für die zahlreichen tollen Erlebnisse und Erinnerungen aus unserer gemeinsamen Promotionszeit; Jun.-Prof. Dr. Andreas Wortmann, der mich während meiner Promotionszeit von seiner Erfahrung profitieren ließ, die gemeinsame spannende Zeit in Schweden zu Beginn meiner Promotion sowie eine gute Zeit bei Konferenzen, die wir gemeinsam besucht haben; Dr. Judith Michael für die vielen Diskussionen, Anregungen und Ideen und ihr immer offenes Ohr; Dr. Katrin Hölldobler für ihre Unterstützung rund um MontiCore und die Gradle-Integration; Imke Drave für die produktive Zusammenarbeit im Bereich Automotive, insbesondere an unseren KI-lastigen Projekten und die KI-Diskussionen, aber auch dafür, die SE Runners wiederbelebt zu haben; Christian Kirchhof für seine Beiträge zu MontiSim, sein Engagement im Bereich Deep Learning für IoT und die gemeinsamen Papiere; Steffen Hillemacher für seine Unterstützung im Bereich Lehre und die Diskussionen rund um das Artefaktenmodell; Malte Heithoff für die spannende Zusammenarbeit an EmbeddedMontiArc und die gemeinsame Veröffentlichung in den letzten Zügen meiner Promotion, aber auch für seine tatkräftige Unterstützung bei der Umsetzung der SI Units; Matthias Markthaler für die interessanten Einblicke in die BMW-Welt und seine Unterstützung während meiner Projektaufenthalte in München. Deni Raco für die kurzweiligen Pokerturniere und Kickerpartien sowie die produktive Zusammenarbeit, vor allem an Avionics-Themen. Marita Breuer und Galina Volkova für die technische Unterstützung rund um MontiCore, Nexus, und eine Vielzahl anderer Themen. Sylvia Gunder und Sonja Müßigbrodt, die bei jeder Frage und Problemstellung immer wussten, was zu tun ist. Mein Dank gilt insbesondere auch Kai Adam, Vincent Bertram, Arvid

Butting, Joel Charles, Manuela Dalibor, Arkadii Gerasimov, Dr. Timo Greifenberg, Nico Jansen, Dr. Oliver Kautz, Achim Lindt, Dr. Markus Look, Dr. Klaus Müller, Pedram Nazari, Lukas Netz, Alexander Roth, David Schmalzing, Steffi Schrader, Dr. Christoph Schulze, Igor Shumeiko, Sebastian Stüber, Simon Varga, Louis Wachtmeister, Vassily Aliseyko, Lennart Bucher, Niklas Dienstknecht, Annika Donath, Christoph Engels, Joshua Mingers, Jerome Pfeiffer, Nina Pichler, Manuel Pützer, Brian Sinkovec und Max Voß.

Ferner bedanke ich mich bei den laufaffinen Kollegen Michael von Wenckstern, Alexander Roth, Pedram Nazari, Imke Drave, Malte Heithoff, Steffen Hillemacher, Matthias Markthaler und Lukas Netz für die zahlreichen zusammen zurückgelegten Kilometer, wenn es mal wieder Zeit war abzuschalten.

Für das Korrekturlesen und sehr hilfreiches Feedback zu meiner Dissertation bedanke ich mich bei Arvid Butting, Arkadii Gerasimov, Manuela Dalibor, Imke Drave, Malte Heithoff, Steffen Hillemacher, Nico Jansen, Oliver Kautz, Christian Kirchhof, Lukas Netz, David Schmalzing, Sebastian Stüber, Marcos Sullivan und Simon Varga.

Ferner bedanke ich mich bei meinen ehemaligen Bacheloranden und Masteranden, die mich tatkräftig bei der Erforschung spannender Themen unterstützt haben und ohne die an eine so komplexe Toolchain wie EmbeddedMontiArc mitsamt Simulator und Deep Learning Framework nicht zu denken wäre.

Mein ganz besonderer Dank gilt meiner Mutter Janna. Danke, dass du mir diesen Weg ermöglicht, mich ausnahmslos bei allen meinen Vorhaben unterstützt und mich stets motiviert hast nach den Sternen zu greifen.

Von ganzem Herzen bedanke ich mich auch bei meiner Freundin Daria für ihre Unterstützung und Motivation während meiner Promotionszeit. Danke für die Geduld, die du während der letzten Jahre aufbringen musstest, wenn die Dissertation alles um mich herum zu verdrängen schien. Danke auch für das Korrekturlesen zahlreicher meiner Papiere und meiner Dissertation.

Aachen, Oktober 2021
Evgeny Kusmenko

Contents