



Nico Jansen, Bernhard Rumpe

Kompositionelle Sprachentwicklung mit der Language Workbench MontiCore

Wiederverwendbarkeit im Software Engineering

MontiCore is a state-of-the-art language workbench for designing and implementing domain-specific languages. It is an open-source research project, continuously developed since 2004, fostering open collaboration and innovation.

MontiCore's research is focused on generation technology and the provision of generative practices. The language workbench is a pioneer in language composition and comes with a large library of reusable language components. MontiCore's development and application is a successful research software engineering endeavor with numerous offsprings in both academia and industry. MontiCore has a lasting relevance in software language engineering, documented in multiple publications and dissertations, and plays a fundamental role in innovating advanced technologies and development practices.

Software hat einen großen Einfluss in Industrie, Gesellschaft und Forschung. Viele Produkte, Produktionsanlagen oder auch Forschungsunterfangen werden stark durch Software getrieben oder unterstützt. Dies erfordert effiziente und qualitativ hochwertige Softwarelösungen. Explizite Modellierungssprachen, wie die Unified Modeling Language, kurz UML, erlauben Architektur und Verhalten von Software zu beschreiben. Domänenspezifische Sprachen, DSLs für Englisch Domain-Specific Languages, ermöglichen es Expertinnen und Experten aus den jeweiligen Disziplinen, Lösungen innerhalb ihrer Expertise zu modellieren, indem Modellierungssprachen präzise auf die Terminologie der Domänen zugeschnitten werden.

Die reduzierten, aber problemangepassten Modellierungstechniken erlauben die automatisierte Verarbeitung der Modelle in Form von hochwertigen Analysen sowie die Synthese vollständiger Softwaresysteme weit jenseits des puren Ausführens von Simulationen. Modellierungssprachen schließen damit die Lücke zwischen Problem- und Lösungsdomäne. Dies bedeutet, dass Menschen ohne weitreichende Programmierkenntnisse die Kontrolle über die Softwareentwicklung übernehmen können. Die Erstellung und

Bereitstellung von DSLs ist daher von großer Bedeutung in Forschung und Industrie. Modellierungssprachen sind aber selbst Forschungsgegenstand, weshalb Forschergruppen einschließlich des Lehrstuhls für Software Engineering (Informatik 3) an der Language Workbench MontiCore^[1] gleichzeitig als Forschungssoftware und als Werkzeug zur Erstellung anderer Forschungssoftware arbeiten. MontiCore ist damit primär ein Meta-Werkzeug, das es ermöglicht, schnell Software-Werkzeuge zu erstellen, die ihrerseits die eigentliche Softwareentwicklung unterstützen.

Die Workbench erlaubt es komplexe, textuelle DSLs aus Bausteinen sowie darauf aufbauende Analyse- und Synthese-Werkzeuge zu erzeugen. Damit dient MontiCore als Basis für laufende Forschung in der modellgetriebenen Softwareentwicklung und unterstützt eine kompositionelle Entwicklung von DSLs für agile Projekte. Das Prinzip der Modellbibliothek wurde systematisiert; angeboten wird auch eine Bibliothek an Kernsprachen, die für die Zusammenstellungen eigener, domänenspezifischer Modellierungssprachen geeignet sind. Beispiele hierfür sind etwa NESTML^[2] für biologische neuronale Netze, SpesML^[3] für das Systems Engineering, Statecharts für die Möglichkeit explizit Unterspezifikation und

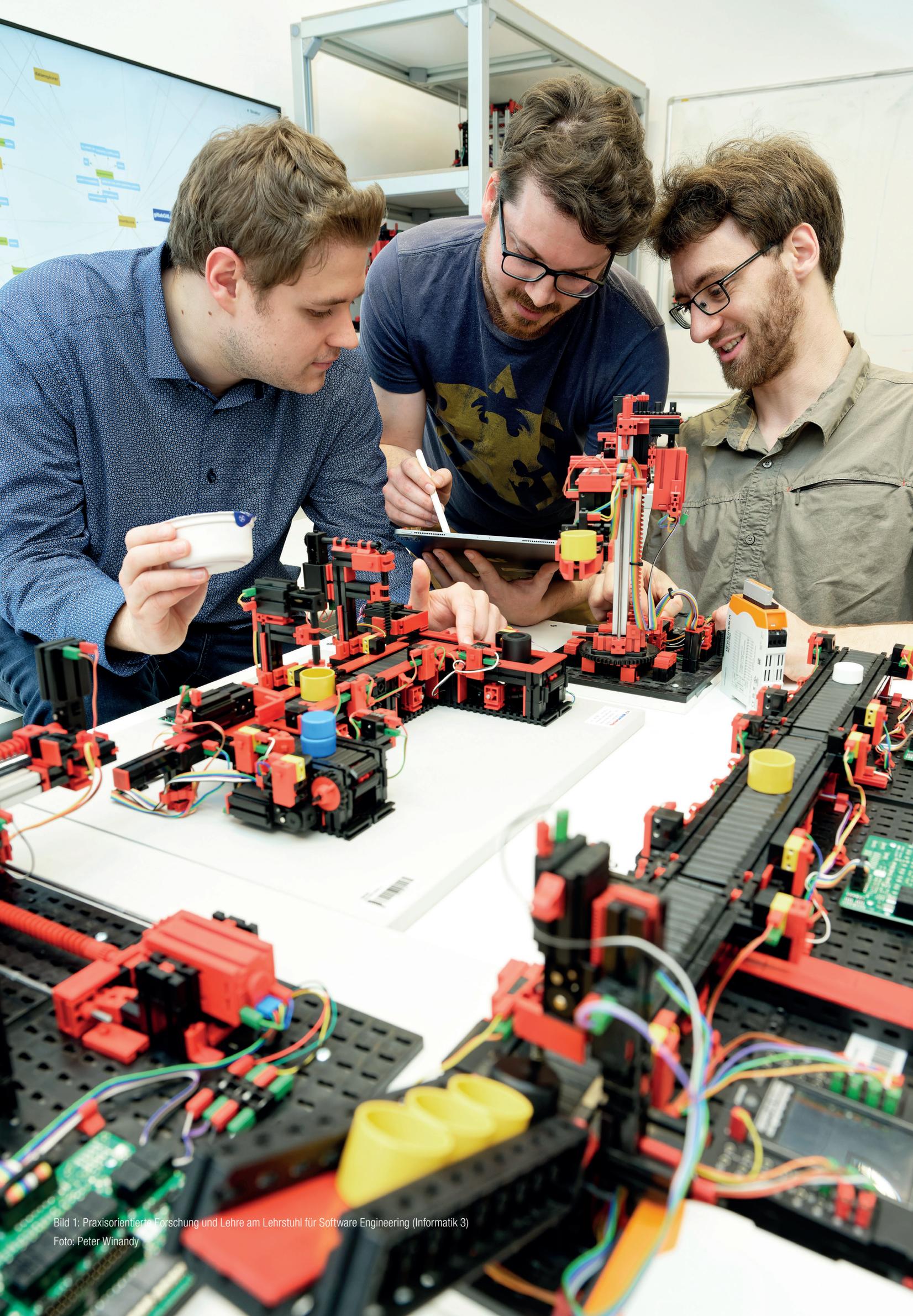


Bild 1: Praxisorientierte Forschung und Lehre am Lehrstuhl für Software Engineering (Informatik 3)

Foto: Peter Winandy

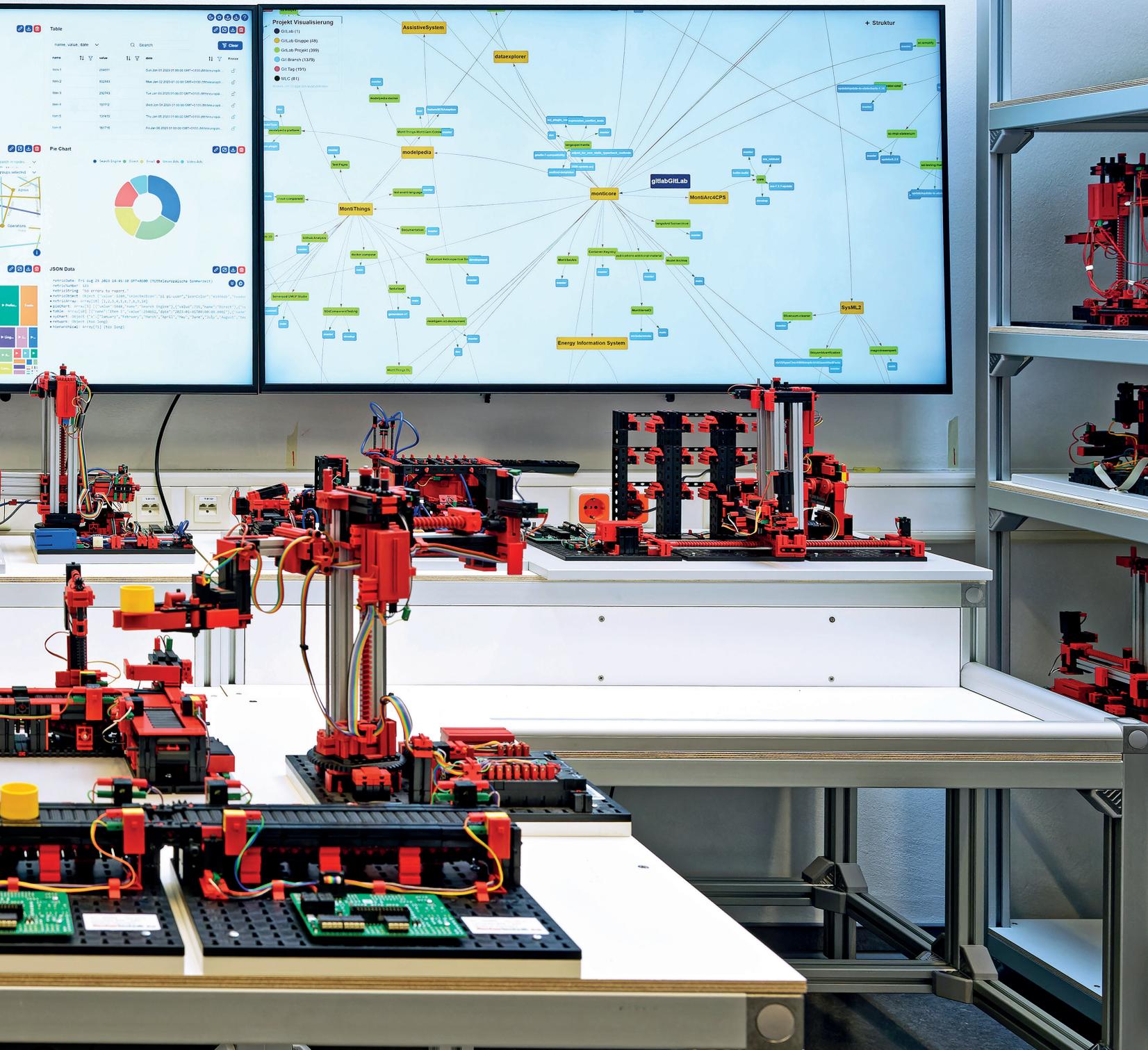


Bild 2: Modellgetriebene Analyse und Simulation einer Demofabrik
Foto: Peter Winandy

Nichtdeterminismus in die Modellierungstechniken einzubringen, Ontologie-Sprachen in zahlreichen Varianten oder Modellierungstechniken für digitale Zwillinge^[4]. Die entwickelten Konzepte und Sprachen verbessern den Software-Engineering-Prozess nachhaltig. Sie erleichtern die Erstellung hochwertiger, individueller DSLs. Die Modelle dieser Sprache können wiederverwendbaren, domänenspezifischen Analysen unterzogen, aber auch als Zwischenschritt zwischen dem wissenschaftlich erklärenden Papier und dem finalen Code genutzt werden und unterstützen so eine effiziente, evolutionäre Weiterentwicklung. So lassen sich zum Beispiel Statecharts darauf vergleichen, ob eine Ver-

feinerungsbeziehung in Bezug auf das Verhalten herrscht und damit die Substituierbarkeit möglich ist. Auch können mit semantikbasierten Differenzverfahren Varianten einer Ontologie hinsichtlich ihrer unterschiedlichen Objektstrukturen geprüft werden. Für die technische Umsetzung basiert MontiCore auf Grammatiken zur Sprachbeschreibung und synthetisiert daraus Infrastruktur für die Modellverarbeitung. Dazu gehören ein Modelloader, Infrastruktur für Wohlgeformtheitsregeln (Kontextbedingungen) und das Management von Symboltabellen^[5]. Basierend auf der Modellierungssprache werden außerdem eine zugehörige Augmentationsprache (Tagging) und eine Transformations-

sprache synthetisiert, die die Modellnutzung und Übersetzung deutlich vereinfachen. Weitere Technologien bieten die Einbettung in Entwicklungsumgebungen, eine flexibel anpassbare Template-basierte Code-Generierung, und eine Übersetzung in Model Checking- und Verifikationsumgebungen. MontiCore zeichnet sich durch die adäquate Anwendung bewährter Entwurfsmuster der Softwaretechnik aus, erweitert diese und hat darüber hinaus eigene Muster für die Entwicklung von DSLs realisiert^[6]. So wird beispielsweise die effiziente Traversierung von Objektstrukturen mithilfe des generierten Visitor Patterns und mittels Template-Hook und RealThis Pattern der kompositionelle



Aufbau von Datenstrukturen und Analyse-/Synthese-Algorithmen ermöglicht. Dies ist eine Kernkonstruktion zur nachhaltigen und flexiblen Wiederverwendung von DSL-Komponenten in verschiedenen Sprachen und damit Werkzeugen. Maßgeschneiderte Werkzeuge können so schnell und effizient entstehen.

Wiederverwendbarkeit auf Sprachebene

Wiederverwendbarkeit ist im Software Engineering von entscheidender Bedeutung, da sie die Effizienz von Entwicklungsprozessen, sowie die Qualität und Wartbarkeit der Software erheblich verbessert. MontiCore ist ein Pionier auf dem Gebiet der Sprach- und

der Werkzeugkomposition und stellt vielfältige Mechanismen zur Erweiterung, Einbettung und Aggregation von DSLs zur Verfügung, um Wiederverwendbarkeit auf Sprachebene zu ermöglichen. Mehrere Mechanismen umfassen sowohl eine technische Umsetzung als auch eine methodische Beschreibung, welche dabei unterstützen, neue DSLs aus existierenden Komponenten zu entwerfen. Eine aggregierende Kopplung von Modellen (auch verschiedener Sprachen) wird über die Technologie der Symboltabellen realisiert, welche Elemente zwischen Modellen nutzbar machen und so eine Gesamtbeschreibung von Systemen ermöglichen. Eine Variante ist die konservative Erweiterung einer Sprache,

bei der die Modelle der ursprünglichen DSL gültig bleiben sowie modular definierte Analyse- und Synthese-Algorithmen ebenfalls weiter eingesetzt werden können. Das unterstützt die Agilität der Spracherweiterung passend zu den ergänzten Anforderungen. Die Einbettung von Sprachkomponenten ermöglicht deren Wiederverwendung und Integration in einer Gesamtsprache. Ein Beispiel ist die Einbettung verschiedener Sprachkomponenten für mehrere Arten von Expressions, Statements, Typdefinitionen oder Literalen in eine Hostsprache. Dadurch wird der Aufbau von Sprachbibliotheken für die modulare Komposition erst möglich wodurch DSLs gemeinsame Sprachkonzepte teilen und so den

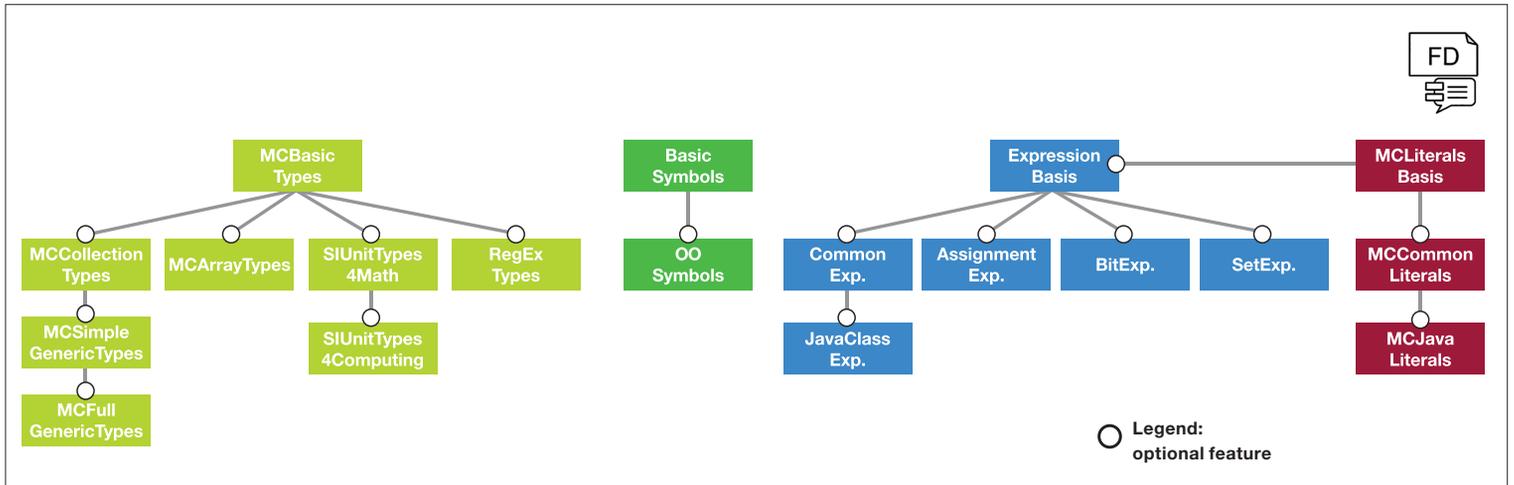


Bild 3: Bibliothek von wiederverwendbaren Sprachkomponenten in MontiCore

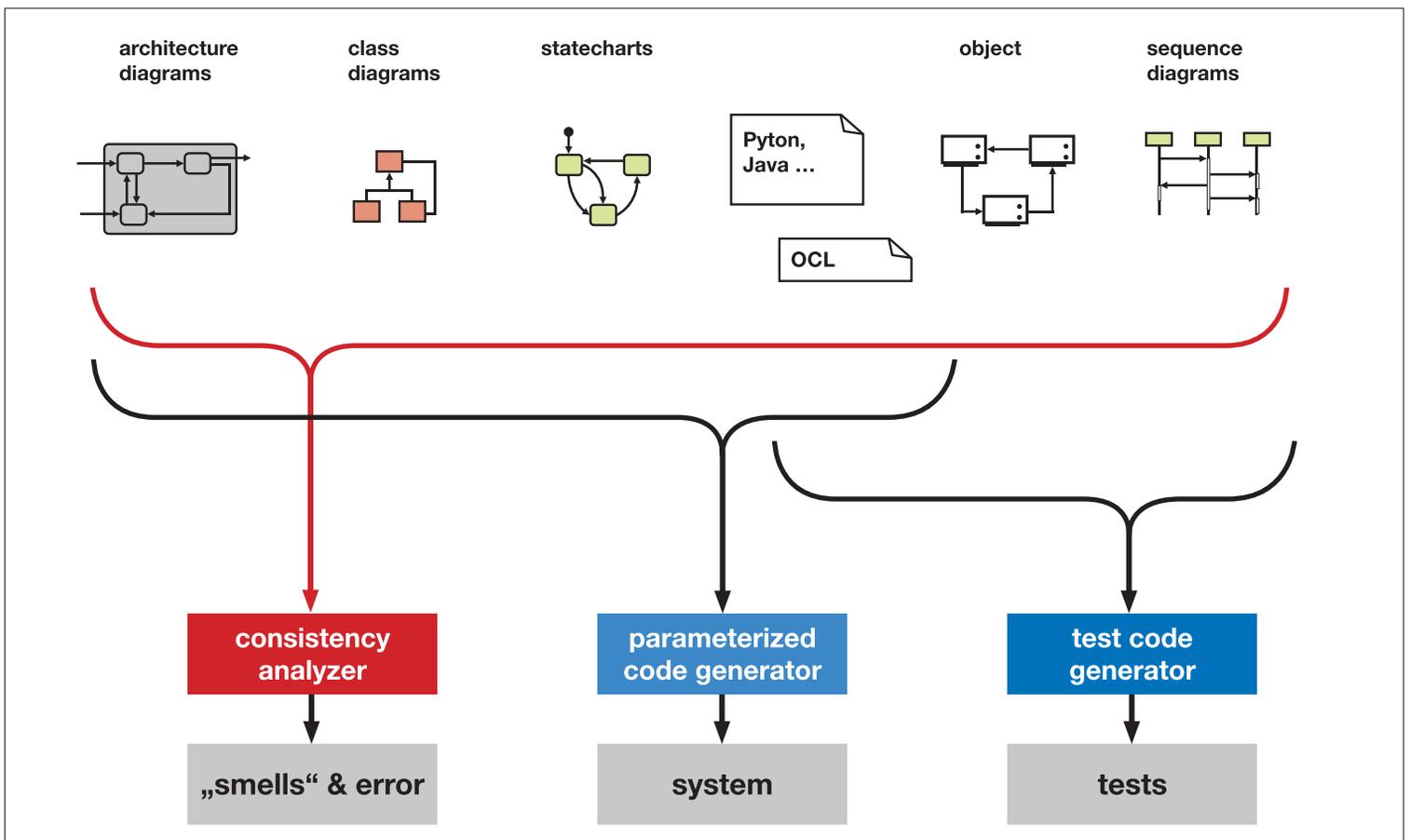


Bild 4: Bibliothek von Software-Modellierungssprachen und ihre Verwendung

Nutzern schnellere Einarbeitung und besseres Wiedererkennen ermöglichen. MontiCore bietet dafür eine weitreichende Bibliothek mit gebräuchlichen Sprachkomponenten^[7]. Bild 3 zeigt den modularen Aufbau dieser Sprachbibliothek. Die verschiedenen Komponenten repräsentieren Ausbaustufen dedizierter Sprachkonzepte. Je nach Anwendungsfall können beim Design einer DSL beispielsweise nur einfache Datentypen (z.B. int, String) oder auch physikalische Typen, also SI-Units (z.B. 3m/s, 220VA) typischer in eine Sprache

eingbracht werden, sodass sich die integrierte Werkzeuginfrastruktur automatisch um die Korrektheit der Konversionen kümmert. Darauf aufbauend hat MontiCore eine weitere Bibliothek an Sprachkomponenten von Modellierungstechniken für Automaten, Sequenzdiagramme, Klassendiagramme, Ontologien oder Prozessmodelle, die den agilen, modellbasierten Entwicklungsprozess unterstützen. Die Language Workbench wird als Forschungssoftware seit 2004 kontinuierlich

entwickelt und hat im Bereich Software Language Engineering viele neue Konzepte hervorgebracht, welche industrielle Frameworks nachhaltig inspirierten. MontiCore hat sich als stabiles und erweiterbares Projekt erwiesen und gleichzeitig den Grundstein für viele weitere Forschungsvorhaben gelegt. Die Workbench ist öffentlich auf GitHub verfügbar, was die Zusammenarbeit mit Industrie und Forschung vorantreibt und Innovation fördert. Durch kontinuierliche Integration und automatisierte Tests, die die Konsistenz und

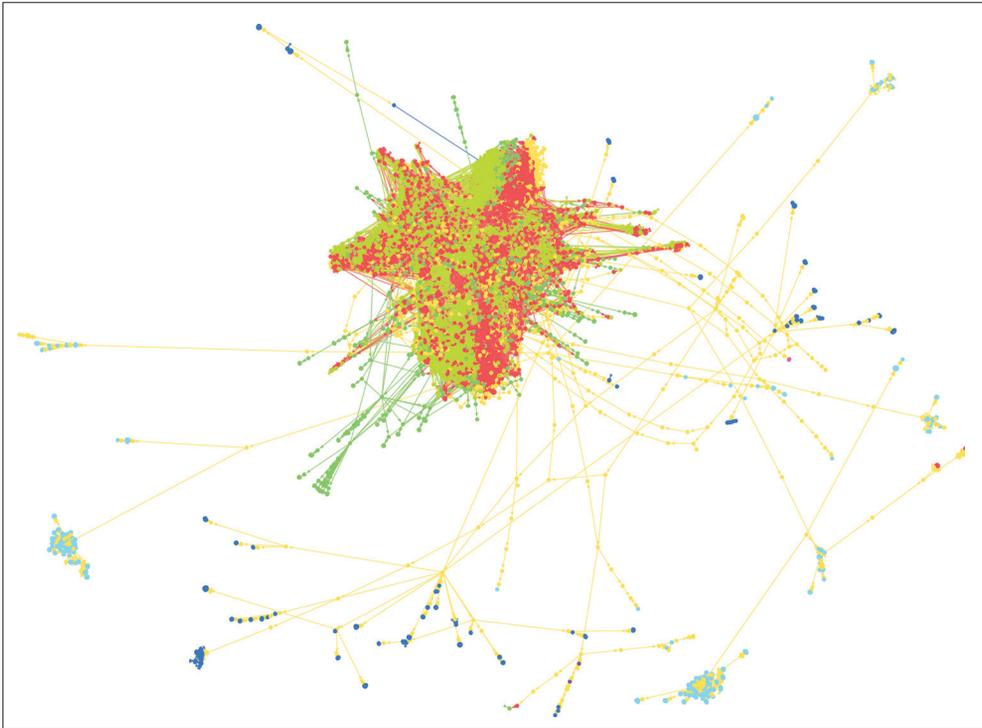


Bild 5: Abhängigkeitsgraph zusammenhängender Artefakte im Software-Engineering-Prozess

Zuverlässigkeit der Software prüfen, wird fortlaufend eine hohe Qualität sichergestellt. Insgesamt waren an der Weiterentwicklung über 40 Personen in mittlerweile 100 Personenn Jahren beteiligt.

MontiCore als Meta-Werkzeug

Als Meta-Werkzeug hat MontiCore zahlreiche Ableger, also konkrete Werkzeuge für Aufgaben in unterschiedlichen technologischen Bereichen. Eine Architekturbeschreibungssprache^[9] erlaubt die Kontrolle von Softwarearchitekturen speziell für die automatisierte Synthese komplexer cyberphysischer Systeme, Infrastrukturen im Bereich Internet of Things und auch Digitaler Zwillinge. Assistenzsysteme im privaten Umfeld oder im Produktionsbereich^[9], Robotersteuerungen, autonome Fahrinfrastrukturen, Verbesserung der Mensch-Maschine-Interaktion durch abstrakte Nutzerführungsmodelle, eine erste Modellierungssprache für Modellbasiertes Machine Learning und vieles mehr wurden so realisiert. Modellbasierte Artefaktanalysen helfen in Software- und Systems-Engineering-Projekten die Projektlandschaft zu verstehen, Abhängigkeiten zu erkennen, Konflikte aufzudecken und die Prozessperformance zu evaluieren^[10]. Bild 5 zeigt einen Abhängigkeitsgraphen von Entwicklungsartefakten und verdeutlicht, wo sogenannte Spaghetti-Klumpen die Modularität zerstören. MontiCore ist in die internationale Forschungskollaboration GEMOC Initiative integriert, um

Software Language Engineering nachhaltig voranzubringen und so die modellbasierte Software- und Systementwicklung weiter in die Praxis zu bringen. Zahlreiche Publikationen belegen die Relevanz von MontiCore in Software Language Engineering, die aber auch durch industrielle Verwendungen nachgewiesen ist. Neuere Forschungsaktivitäten dienen dem Ausbau der Language Workbench als Infrastruktur für Forschungssoftware anderer Domänen, um zum Beispiel Forschungssoftware präziser mit den abstrakten, oft mathematischen Modellen in Physik, Chemie, oder anderen Fachbereichen zu verzahnen, Datenmengen und ihre Meta-Daten effektiv zu managen und gleichzeitig stringente Nachvollziehbarkeit der Forschungsergebnisse und Weiterentwicklungen der Software sicherzustellen.



www.se-rwth.de/research/

Autoren

Nico Jansen, M.Sc., ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Software Engineering (Informatik 3).
 Univ.-Prof. Dr.rer.nat. Bernhard Rumpe ist Inhaber des Lehrstuhls für Software Engineering (Informatik 3).

Literatur

- [1] Hölldobler, K., Kautz, O., Rumpe, B., MontiCore Language Workbench and Library Handbook: Edition 2021, Aachener Informatik-Berichte, Software Engineering, Band 48, ISBN 978-3-8440-8010-0, Shaker Verlag, May 2021
- [2] Plotnikov, D., Blundell, I., Ippen, T., Eppler, J. M., Morrison, A., Rumpe, B., NESTML: a modeling language for spiking neurons, in: Modellierung 2016 Conference, Volume 254, pp. 93-108, LNI, Bonner Köllen Verlag, Mar 2016
- [3] Gupta, R., Jansen, N., Regnat, N., Rumpe, B., Implementation of the SpesML Workbench in MagicDraw, in: Modellierung 2022 Satellite Events, pp. 61-76, Gesellschaft für Informatik, Jun 2022
- [4] Dalibor, M., Heithoff, M., Michael, J., Netz, L., Pfeiffer, J., Rumpe, B., Varga, S., Wortmann, A., Generating Customized Low-Code Development Platforms for Digital Twins, in: Journal of Computer Languages (COLA), Volume 70, Art. 101117, Elsevier, Jun 2022
- [5] Butting, A., Michael, J., Rumpe, B., Language Composition via Kind-Typed Symbol Tables, Journal of Object Technology (JOT), October 2022
- [6] Drux, F., Jansen, N., Rumpe, B., A Catalog of Design Patterns for Compositional Language Engineering, Journal of Object Technology (JOT), October 2022
- [7] Butting, A., Eikermann, R., Hölldobler, K., Jansen, N., Rumpe, B., Wortmann, A., A Library of Literals, Expressions, Types, and Statements for Compositional Language Design, in: Journal of Object Technology (JOT), October 2020
- [8] Haber, A., Ringert, J. O., Rumpe, B., MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems, RWTH Aachen, AIB-2012-03, Technical Report, Feb 2012
- [9] Michael, J., A Vision Towards Generated Assistive Systems for Supporting Human Interactions in Production, in: Modellierung 2022 Satellite Events, pp. 150-153, Gesellschaft für Informatik e.V., Jul 2022
- [10] Greifenberg, T., Hillemacher, S., Hölldobler, K., Applied Artifact-Based Analysis for Architecture Consistency Checking, in: Ernst Denert Award for Software Engineering 2019, pp. 61-85, Springer, Dec 2020