

## Komposition Domänenspezifischer Sprachen unter Nutzung der MontiCore Language Workbench, am Beispiel SysML 2

Katrin Hölldobler<sup>1</sup>, Nico Jansen<sup>1</sup>, Bernhard Rumpe<sup>1</sup>, Andreas Wortmann<sup>1</sup>

**Abstract:** MontiCore ist eine Language Workbench zum Design und zur Realisierung von textuellen domänenspezifischen Sprachen (DSLs). MontiCore ermöglicht die Erforschung von modellbasierten Softwareentwicklungsmethoden durch eine Vielfalt von DSLs und Sprachkomponenten. Darüber hinaus wird MontiCore erfolgreich in akademischen wie auch industriellen Projekten in verschiedenen Domänen wie der Automobilbranche, im Bereich Energiemanagement und der Robotik eingesetzt. In diesem Tutorial erklären wir die kompositionale Entwicklung von DSLs mit MontiCore am Beispiel SysML 2.

**Keywords:** Modellgetriebene Softwareentwicklung, Language Workbench, domänenspezifische Sprachen, Komposition, Modularität

### 1 Gegenstand des Tutorials

Die modellbasierte Softwareentwicklung (MBSE) verwendet Modelle, um die Komplexität der zu entwickelnden Systeme zu reduzieren. Zur Entwicklung dieser Modellen werden General-Purpose Modellierungssprachen wie beispielsweise die UML oder domänenspezifische Sprachen (DSLs) verwendet. DSLs bieten hierbei den Vorteil, dass sie sich besser für die jeweilige Domäne eignen, allerdings auch spezifisch für diese entwickelt werden müssen. MontiCore [HR17] ist eine Language Workbench, die die Entwicklung entsprechender, textueller DSLs und Modellierungssprachen erleichtert. Hierzu bietet MontiCore ein EBNF-artiges Grammatikformat sowie Möglichkeiten zur modularen Entwicklung und Komposition von DSLs und dazu passenden Tools und Infrastruktur [KRV10, HRW18]. Das Ziel der Entwicklung von MontiCore war und ist es eine leistungsstarke und effiziente Workbench zu erschaffen, die eine agile Entwicklung von DSLs zusammen mit dazu passender Infrastruktur für Analysen, Transformationen und Codegenerierung, ermöglicht. Wichtige Features von MontiCore sind zudem:

- Kombinierte Spezifikation von konkreter und abstrakter Syntax
- Anpassbare Generierung von Parser und abstraktem Syntaxbaum
- Generierung von Analyseinfrastruktur inklusive Visitoren
- Variable Grammatikverarbeitung via interpretierten Groovy Skripten
- Konfigurierbares Logging and Prozessreporting
- Freemarker Template Engine für leichtgewichtige Codegenerierung

<sup>1</sup> Software Engineering, RWTH Aachen University, Aachen, Germany



## 2 Ziele des Tutorials

In diesem Tutorial präsentieren wir die Language Workbench MontiCore. Das ca. zweistündige Tutorial teilt sich in einen Präsentations- und einen Hands On-Teil. Es wird erklärt wie DSLs und Modellierungssprachen mit MontiCore entwickelt werden können. Hierzu dienen Modellierungssprachen der SysML 2 [Ob] als Beispiel. Der Fokus des Tutorials liegt auf der Entwicklung neuer Sprachen durch Komposition und Erweiterung vorhandener Sprachen und Sprachkomponenten. Dieses Tutorial richtet sich an modellierungserfahrene Personen, die verstehen möchten, wie eine Language Workbench funktioniert, um modular neue Sprache aus vorhandenen Sprachkomponenten zu konstruieren. Idealerweise kennen die Teilnehmer bereits Modellierungssprachen wie die UML oder SysML 2 und die Programmiersprache Java. Das Tutorial umfasst:

- Definition von DSLs mit der Language Workbench MontiCore
- Die MontiCore Sprachbibliothek und deren Kombinationsmöglichkeiten
- Entwicklung von SysML 2 Modellierungssprachen durch Komposition und Erweiterung vorhandener Sprachen und Sprachkomponenten

**Bernhard Rumpe** leitet den Lehrstuhl für Software Engineering der RWTH Aachen University. Sein Hauptinteresse sind rigorose und praktische Software- und Systementwicklungsmethoden, die auf adäquaten Modellierungstechniken basieren. Dies umfasst sowohl agile Entwicklungsmethoden als auch Modellengineering basierend auf UML-artigen Notationen und domänenspezifischen Sprachen. Er hat einen Beitrag zu einer Vielzahl an Modellierungstechniken geleistet wie beispielsweise der UML Standardisierung. Er wendet Modellierungstechniken auf autonomes Fahren, Gehirnsimulation, BIM Energiemanagement, Produktionsautomatisierung und vieles mehr an. In seinen Projekten arbeitet er intensiv mit allen großen, deutschen Automobilherstellern, Energieunternehmen, Versicherungen und Banken, einem großen Flugzeughersteller, einem Raumfahrtunternehmen sowie innovativen Start-Ups der IT-Branche zusammen.

## Literaturverzeichnis

- [HR17] Hölldobler, Katrin; Rumpe, Bernhard: MontiCore 5 Language Workbench Edition 2017. Aachener Informatik-Berichte, Software Engineering, Band 32. Shaker Verlag, December 2017.
- [HRW18] Hölldobler, Katrin; Rumpe, Bernhard; Wortmann, Andreas: Software Language Engineering in the Large: Towards Composing and Deriving Languages. Computer Languages, Systems & Structures, 54:386–405, 2018.
- [KRV10] Krahn, Holger; Rumpe, Bernhard; Völkel, Stefan: MontiCore: a Framework for Compositional Development of Domain Specific Languages. International Journal on Software Tools for Technology Transfer (STTT), 12(5):353–372, September 2010.
- [Ob] Object Management Group: , SYSML V2. <http://www.omg.sysml.org/SysML-2.htm>. [Online; accessed 18. November 2019].