

RWTH Aachen University  
Software Engineering



# Architecture and Behavior Modeling of Cyber-Physical Systems with MontiArcAutomaton

Jan Oliver Ringert  
Bernhard Rumpe  
Andreas Wortmann



December 18, 2014



# **Aachener Informatik-Berichte, Software Engineering**

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

Band 20

**Jan Oliver Ringert  
Bernhard Rumpe  
Andreas Wortmann**

## **MontiArcAutomaton**

Architecture and Behavior Modeling of  
Cyber-Physical Systems with MontiArcAutomaton

Shaker Verlag  
Aachen 2014

**Bibliografische Information der Deutschen Nationalbibliothek**

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

Copyright Shaker Verlag 2014

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-3299-4

ISSN 1869-9170

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

# Abstract

This book presents MontiArcAutomaton, a modeling language for architecture and behavior modeling of Cyber-Physical Systems as interactive Component & Connector models. MontiArcAutomaton extends the Architecture Description Language MontiArc with automata to describe component behavior.

The modeling language MontiArcAutomaton provides syntactical elements for defining automata with states, variables, and transitions inside MontiArc components. These syntactical elements and a basic set of well-formedness rules provide the syntax for a family of modeling languages for state-based behavior modeling in Component & Connector architectures. We present two concrete language profiles with additional well-formedness rules to model time-synchronous component behavior and untimed, event-driven behavior of components.

This book gives an overview of the MontiArcAutomaton language including examples, a language reference, and a context-free grammar for MontiArcAutomaton models. It also provides syntax definition, well-formedness rules, and semantics for two language profiles. We summarize projects and case studies applying MontiArcAutomaton.

MontiArcAutomaton is implemented using the DSL framework MontiCore. Available tools include a textual editor with syntax highlighting and code completion as well as a graphical editor and a powerful and extensible code generation framework for target languages including EMF, Java, Mona, and Python.



# Contents

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| <b>1. Introduction</b>                                             | <b>1</b>  |
| <b>2. The Architecture Description Language MontiArc</b>           | <b>3</b>  |
| 2.1. Applications and Extensions of MontiArc . . . . .             | 5         |
| <b>3. MontiArcAutomaton Example and Overview</b>                   | <b>7</b>  |
| 3.1. Related Modeling Languages . . . . .                          | 10        |
| <b>4. A MontiArcAutomaton Language Profiles and Semantics</b>      | <b>13</b> |
| 4.1. Language Profile for Time-Synchronous Communication . . . . . | 14        |
| 4.2. A Language Profile for Event-Driven Automata . . . . .        | 16        |
| <b>5. MontiArcAutomaton Language Reference</b>                     | <b>19</b> |
| 5.1. Automaton Declarations . . . . .                              | 19        |
| 5.2. Variables . . . . .                                           | 19        |
| 5.3. Values and Sequences . . . . .                                | 20        |
| 5.4. State Declarations . . . . .                                  | 21        |
| 5.4.1. Initial States and Initial Outputs . . . . .                | 22        |
| 5.5. Transitions . . . . .                                         | 23        |
| 5.5.1. Guard Expressions over Ports and Variables . . . . .        | 24        |
| 5.5.2. Input on Ports and Current Variable Values . . . . .        | 24        |
| 5.5.3. Output on Ports . . . . .                                   | 26        |
| 5.5.4. Variable Value Assignments . . . . .                        | 27        |
| 5.6. Generic Types . . . . .                                       | 27        |
| <b>6. MontiArcAutomaton Context Conditions</b>                     | <b>29</b> |
| 6.1. Uniqueness Conditions . . . . .                               | 29        |
| 6.2. Convention Conditions . . . . .                               | 31        |
| 6.3. Referential Integrity Conditions . . . . .                    | 32        |
| 6.4. Type Correctness Conditions . . . . .                         | 33        |
| 6.5. $MAA_{ts}$ Specific Context Conditions . . . . .              | 37        |
| 6.6. $MAA_{ed}$ Specific Context Conditions . . . . .              | 40        |
| <b>7. Case Studies</b>                                             | <b>43</b> |
| 7.1. Lego NXT Coffee Service . . . . .                             | 44        |
| 7.2. ROS Robotino Logistics . . . . .                              | 44        |
| 7.3. SmartSoft Robotino Logistics . . . . .                        | 45        |

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| 7.4. IServeU . . . . .                                               | 46        |
| <b>8. Discussion and Current Limitations</b>                         | <b>49</b> |
| 8.1. Expressiveness of Automata for Behavior Modeling . . . . .      | 49        |
| 8.2. Extension with Additional Behavior Modeling Languages . . . . . | 50        |
| 8.3. Code Generation, Deployment, and Platforms . . . . .            | 50        |
| <b>9. Summary</b>                                                    | <b>51</b> |
| <b>Bibliography</b>                                                  | <b>53</b> |
| <b>List of Figures</b>                                               | <b>61</b> |
| <b>List of Listings</b>                                              | <b>63</b> |
| <b>A. Human Readable Grammar</b>                                     | <b>65</b> |
| <b>B. Parser Grammar</b>                                             | <b>67</b> |