Modeling and Developing Digital Twins

Malte Heithoff, Judith Michael, Bernhard Rumpe
Software Engineering
RWTH Aachen

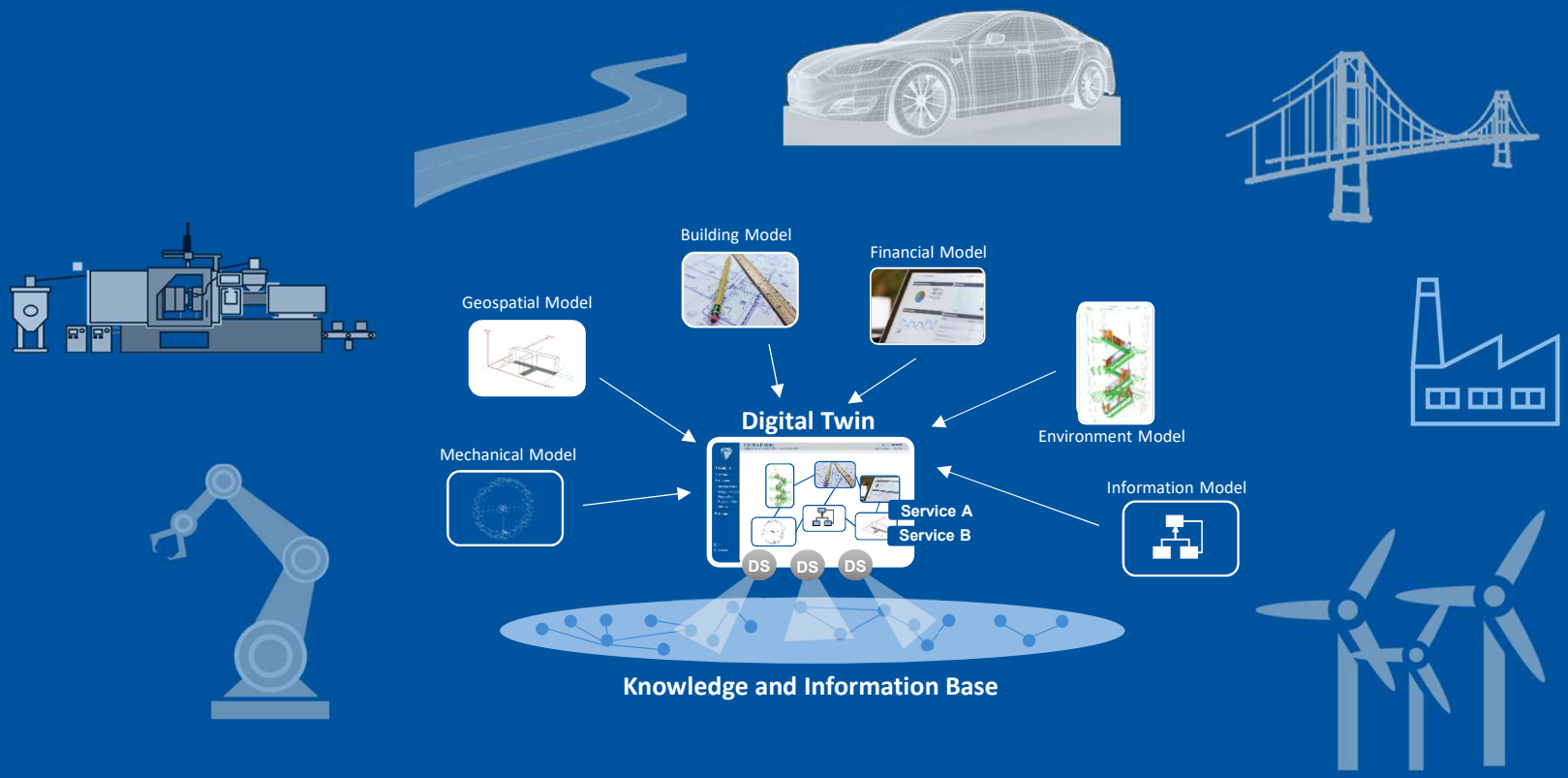
http://www.se-rwth.de
@SE_RWTH

28.9.2022, Aachen

SE Software Engineering | RWTH AACHEN UNIVERSITY

Creating digital twins for …

… elements in the physical world
that can be monitored, sensed, actuated and controlled

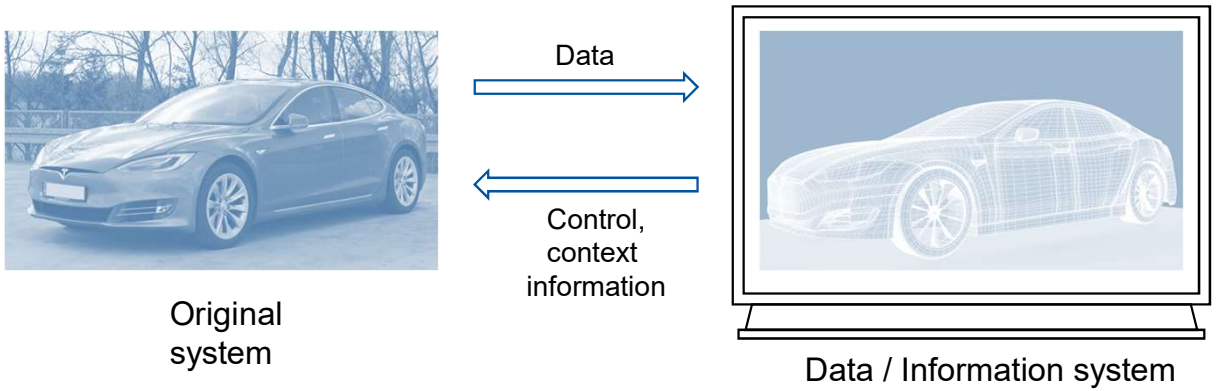


Digital twin consists of models, data and services

Our digital twin definition [DMR+20]

contextual data and their aggregation and abstraction

A digital twin of a system consists of a **set of models of the system**, a **set of digital shadows**, and a **set of services** that allow using the data and models **purposefully with respect to the original system**.



Active software systems connected to another (cyber-physical) system and using data and models during its runtime to interact with the system, e.g., to optimize its behavior

Models for the Engineering of Digital Twins

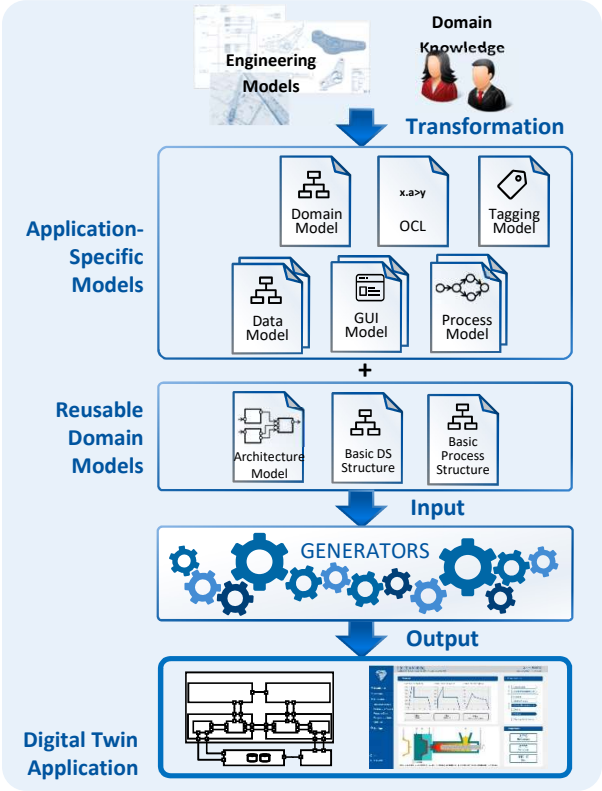
Derive information from engineering models to create system models

Models at design time

- Application-specific models, e.g.,
  - domain model, GUI models [GMN+20]
  - process models (planned processes) [BHK+21]
  - OCL for validation, Tagging
- Reusable domain models , e.g.,
  - system architecture
  - basic digital shadow structure [BBD+21]
  - basic process structure [BHK+21]

Models at run-time

- e.g., digital shadows, processes (in reality), goals, actions, events, …



Developing Digital Twins: Challenges and Approach

- **Challenges**
  - digital twins configured and operated by shop-floor experts: rarely professional software engineers
  - rising complexity of digital twins
  - heterogeneous data sources and software applications to connect

2-step generation process [DHM+22]
- we generate the low-code platform
- shop-floor experts configure a digital twin via the low-code platform and
- generate one or more digital twins

- **Enablers of the approach**
  - model-driven toolchain for generating information systems
  - model-driven digital twin architecture and toolchain
  - reusable language components, services and models

PDF of this
Publication
se-rwth.de/publications

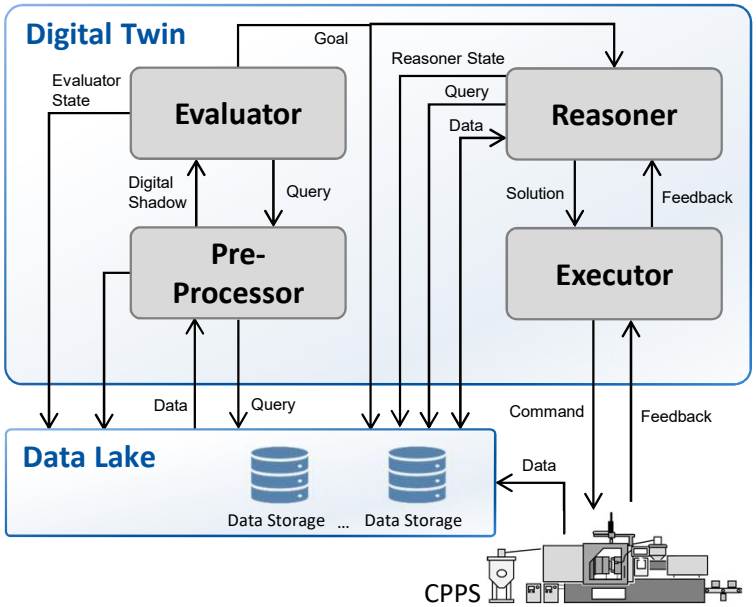
[DHM+22] M. Dalibor, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, S. Varga, A. Wortmann: *Generating Customized Low-Code Development Platforms for Digital Twins*. Journal of Computer Languages (COLA)

Digital Twin Architectures

- Digital twins can actively
  - represent, control, and/or optimize the behavior of an observed (cyber-physical) system
  - reflected in the DT system architecture:
    - digital twin services

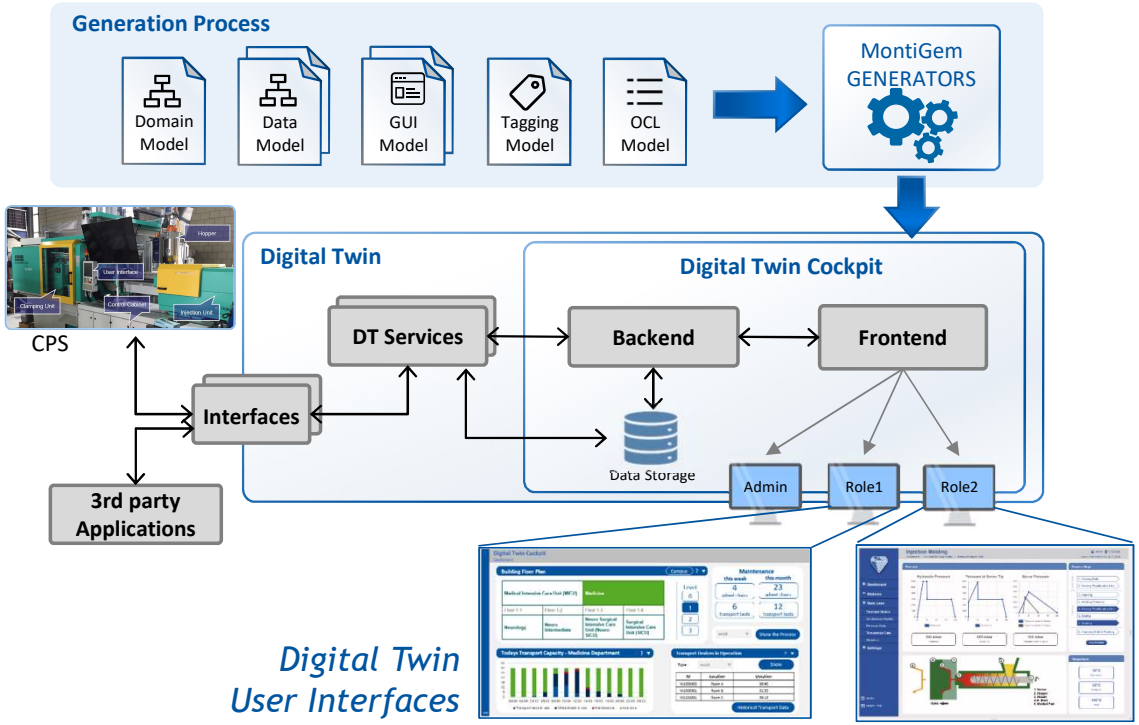
- Architecture model as reference model
  - e.g., MontiArc [BDH+20]
  - used for Model-Driven Software Engineering

- Digital twin services
  - e.g., monitoring, analyzing, optimization, execution, visualization, process conformance, simulation,…
  - possible implementation for self-adaptive digital twins: MAPE-K [BDH+20]



Creating Digital Twin Cockpits with MontiGem

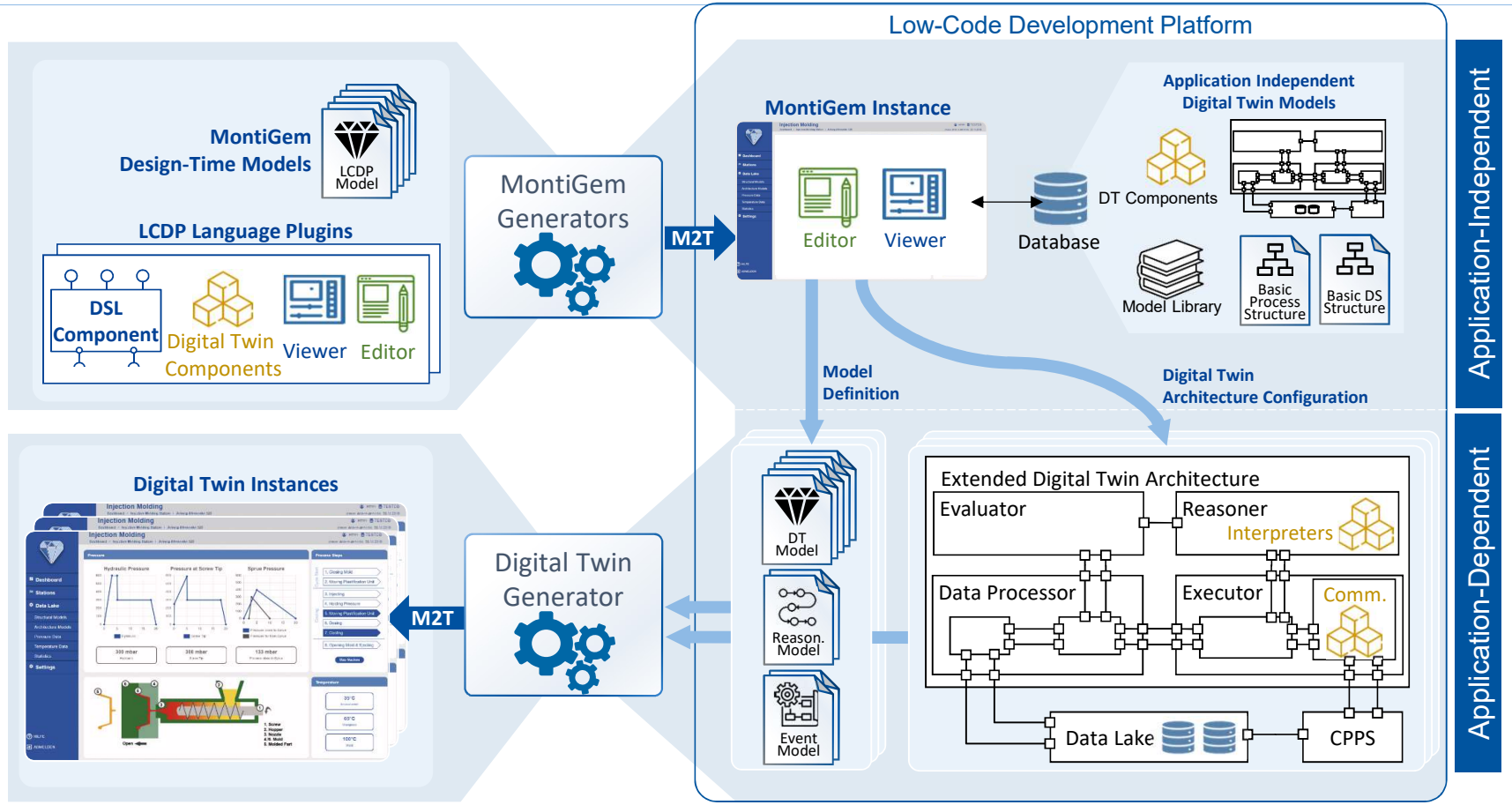
- Digital twin cockpit [DMR+20]
  - visualization of monitoring data and models of CPS
- Generating digital twin cockpits
  - from models
  - with the generator framework MontiGem [GMN+20]
  - loose coupling with DT services
- DT services
  - e.g., self-adaptivity (MAPE-K), AI, visualization, conformance checking, optimization
  - interfaces to CPS | 3rd party applications
- Applied in several use cases
  - injection molding [DMR+20]
  - engineering of wind turbines [MNN+22]
  - automated hospital transportation system [BMR+22]



Low-Code Platforms for Model-Driven Digital Twins | Overview

2-step generation process [DHM+22]

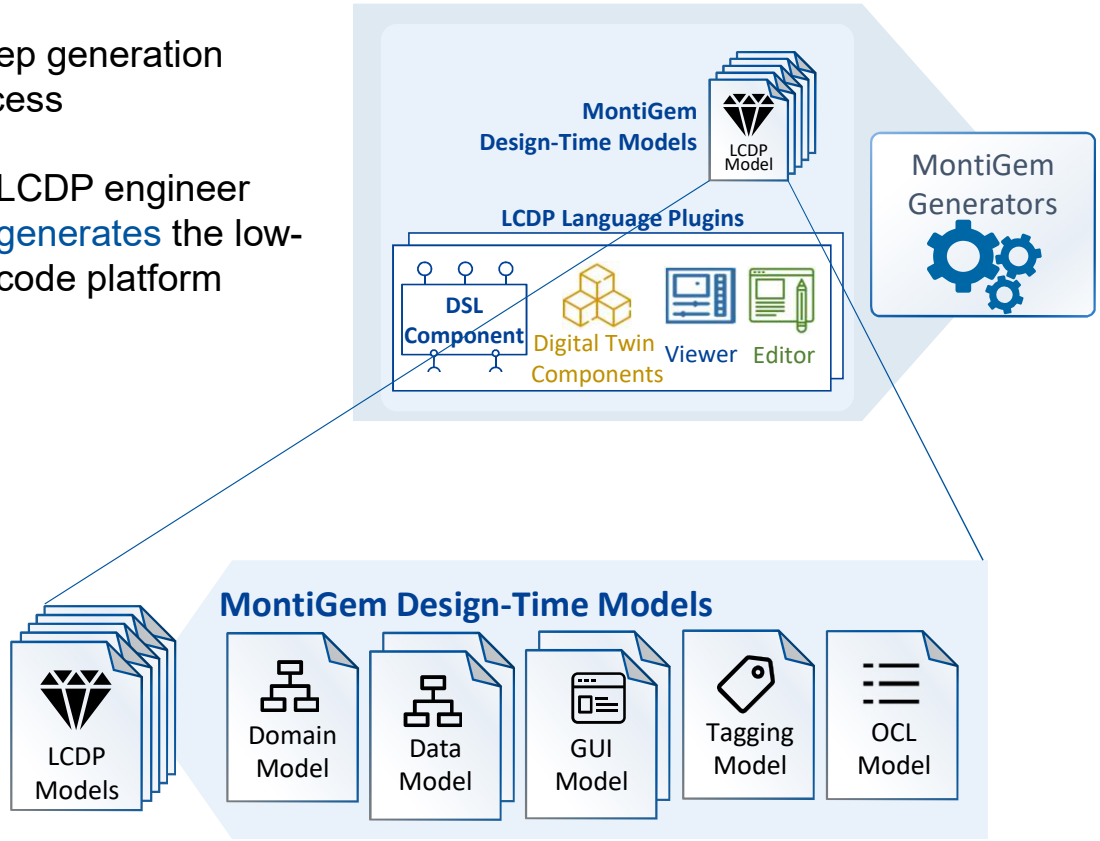
1) LCDP engineer generates the low-code platform
2) Digital twin designer configures a digital twin via the LCDP and generate one or more digital twins
3) Domain experts operate on digital twins



Low-Code Platforms for Model-Driven Digital Twins | Design-Time Models

2-step generation process

1) LCDP engineer generates the low-code platform



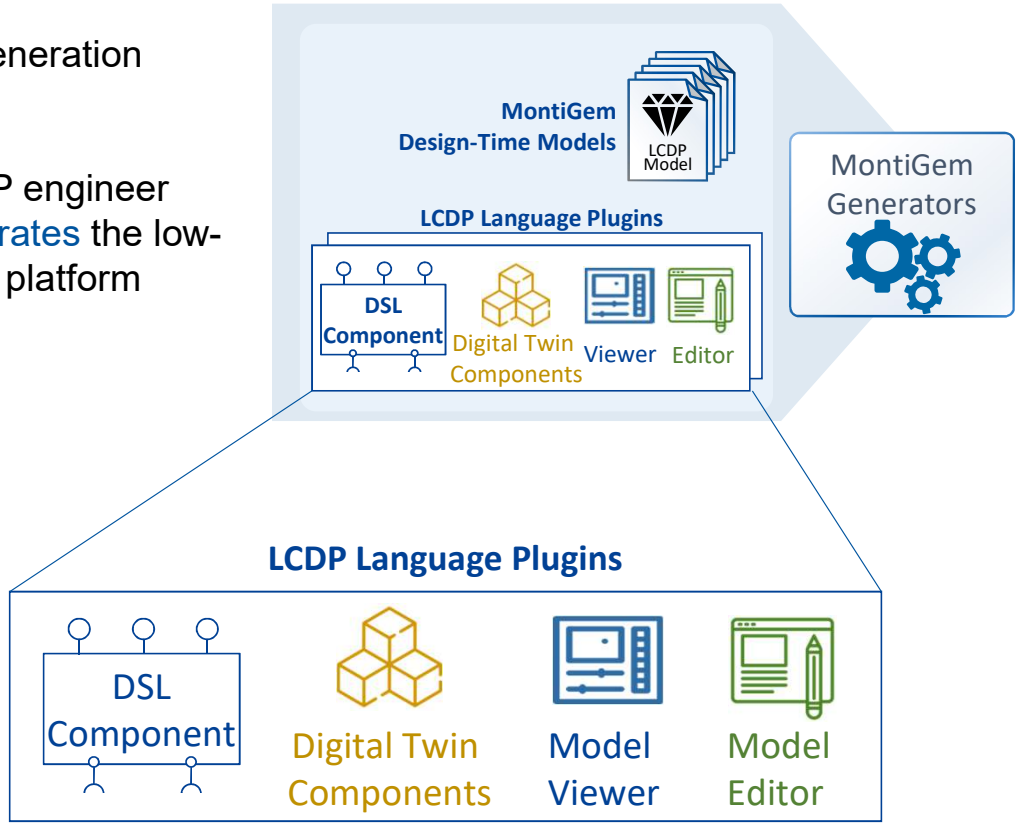
- MontiGem Design-Time Models are defining
  - Data structure of the LCDP
  - Graphical user interfaces and data views
  - Validators
  - Technology specific information

- Considerations
  - *What data structures are needed in the low-code development platform?*
  - *What modeling languages do we need for generating the LCDP?*
  - *What should be generated?*

Low-Code Platforms for Model-Driven Digital Twins | Language Plugins

2-step generation process

1) LCDP engineer generates the low-code platform



- LCDP Language Plugins are defining
  - Domain-specific language to be used
  - Software component of the digital twin able to interpret models in that DSL
  - Viewer and Editor for models usable in the LCDP

- Considerations
  - *What components should be selectable by the digital twin designer in the LCDP?*
  - *What DSL should be used during runtime?*

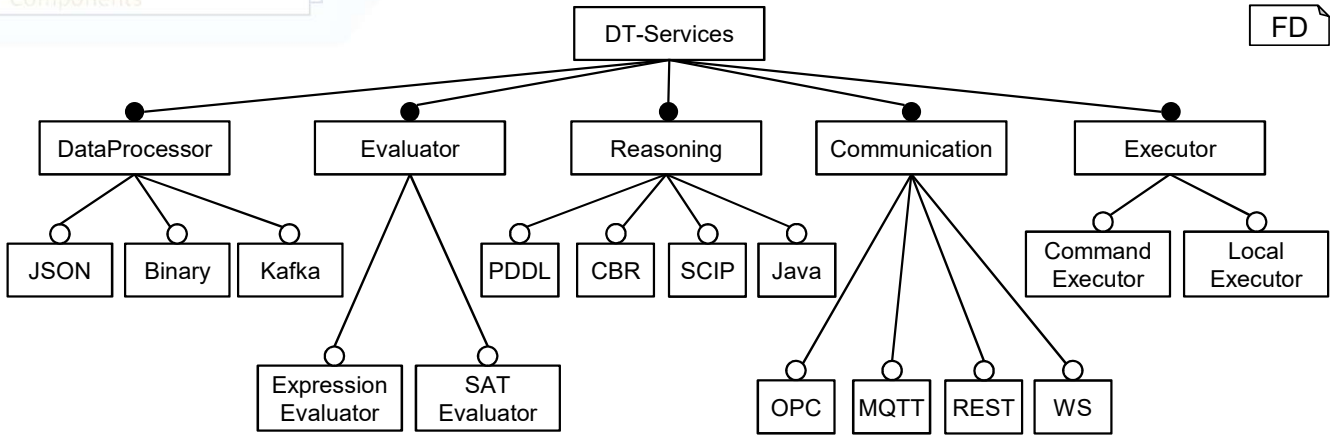
Language Components for Digital Twins

2-step generation process

1) LCDP engineer generates the low-code platform

Reasoning Language Plugins

- New (e.g., AI) planning capabilities for digital twin
- New reasoner subcomponents
- Models can change often during runtime
  - e.g., case-based reasoning, PDDL

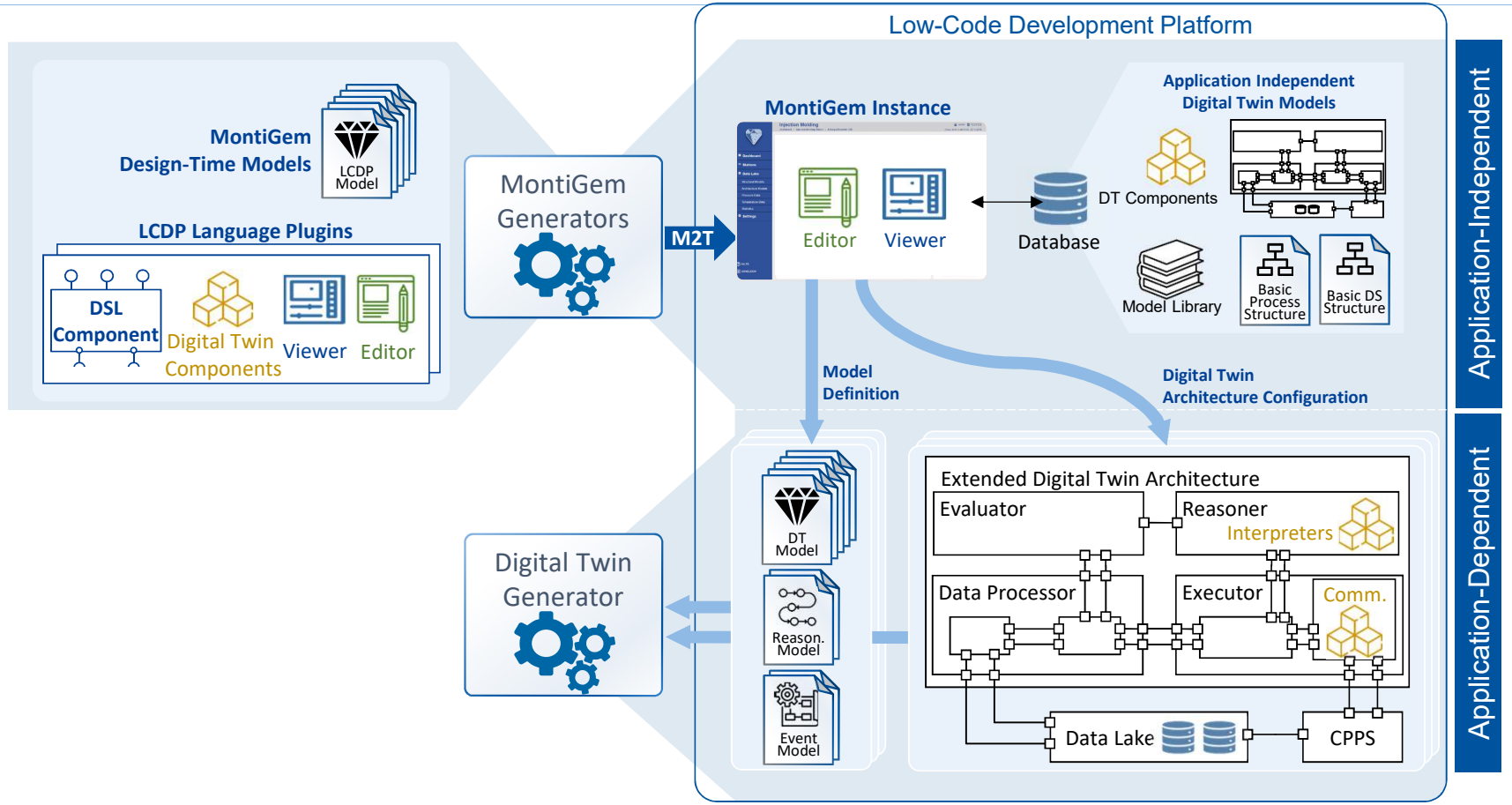


Low-Code Platforms for Model-Driven Digital Twins | Configuration

2-step generation process

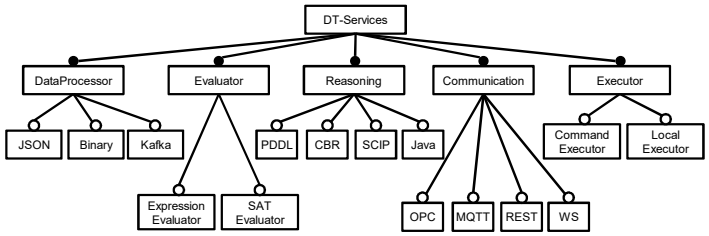
1) LCDP engineer generates the low-code platform

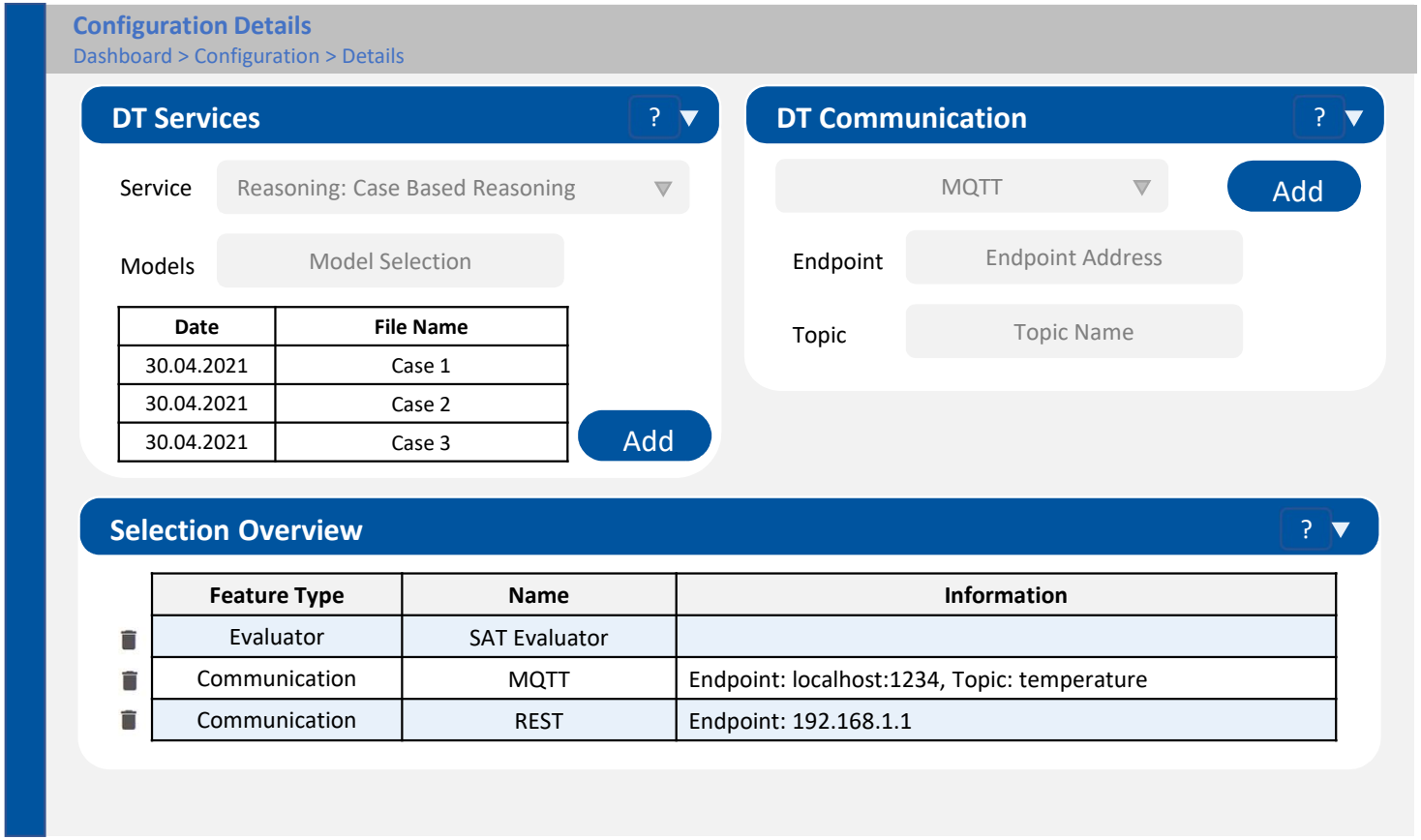
2) Digital twin designer configures a digital twin via the LCDP and generate one or more digital twins



Using the Low-Code Development Platform to Configure DTs

- Configuration of services
  - Selection
  - Definition or upload of models
- Configuration of DT communication
  - parametrization

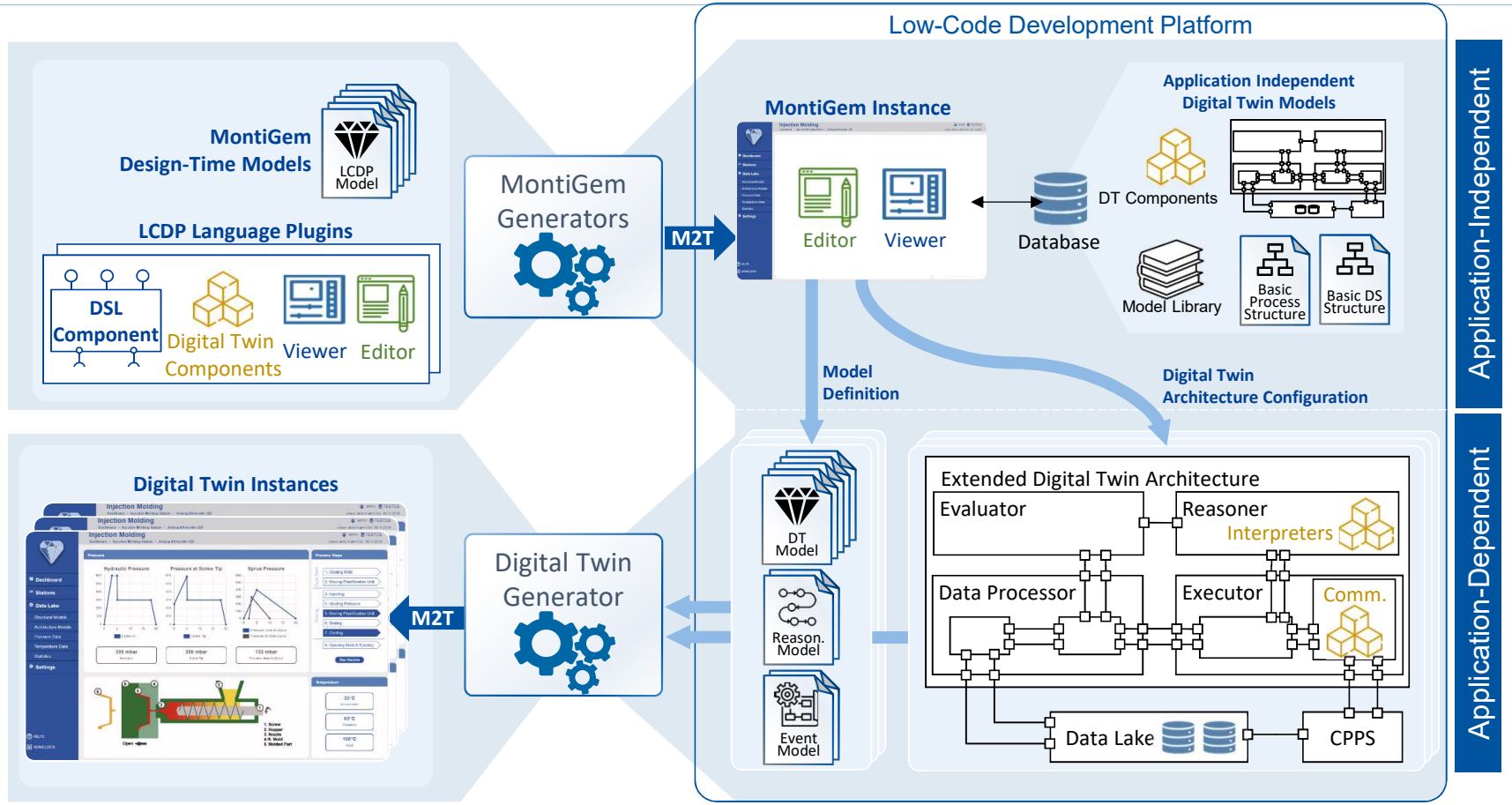




Low-Code Platforms for Model-Driven Digital Twins | Generation & Operation

2-step generation process

1) LCDP engineer generates the low-code platform
2) Digital twin designer configures a digital twin via the LCDP and generate one or more digital twins
3) Domain experts operate on digital twins



Summary

- *Low-Code Development*
  - Success story of model-driven engineering
  - Reduces complexity
  - Empowers citizen developers

- *Model-Driven Engineering of Digital Twins…*
  - Facilitates their creation and configuration
  - Fosters re-use of DSLs and SW components
  - Helps us to handle data and models
    - cross-domain
    - from different data sources
    - and on a sufficient aggregation level

LangDev 2022
organized by SE | RWTH AACHEN UNIVERSITY

Well-designed DSLs provide a good base for creating complex software systems and systems-of-systems!

Selected References

Modeling in Industry 4.0

- [BDJ+22] P. Brauner, M. Dalibor, M. Jarke, I. Kunze, I. Koren, G. Lakemeyer, M. Liebenberg, J. Michael, J. Pennekamp, C. Quix, B. Rumpe, W. van der Aalst, K. Wehrle, A. Wortmann, M. Ziefle: *A Computer Science Perspective on Digital Transformation in Production*. ACM TIOT 3, 2022
- [Mic22] J. Michael: *A Vision Towards Generated Assistive Systems for Supporting Human Interactions in Production*. Modellierung'22
- [WBCW20] A. Wortmann, O. Barais, B. Combemale, M. Wimmer: *Modeling Languages in Industry 4.0: an Extended Systematic Mapping Study*. SoSyM 19(1), 2020

Digital Twins

- [DJR+22] M. Dalibor, N. Jansen, B. Rumpe, D. Schmalzing, L. Wachtmeister, M. Wimmer, A. Wortmann: *A cross-domain systematic mapping study on software engineering for Digital Twins*. Journal of Systems and Software, 2022
- [MNN+22] J. Michael, I. Nachmann, L. Netz, B. Rumpe, S. Stüber: *Generating Digital Twin Cockpits for Parameter Management in the Engineering of Wind Turbines*. Modellierung'22
- [BMR+22] D. Bano, J. Michael, B. Rumpe, S. Varga, M. Weske: *Process-Aware Digital Twin Cockpit Synthesis from Event Logs*. Journal of Computer Languages (COLA) 70, 2022.
- [DHM+22] M. Dalibor, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, S. Varga, A. Wortmann: *Generating Customized Low-Code Development Platforms for Digital Twins*. Journal of Computer Languages (COLA) 70, 2022.
- [MPRW22] J. Michael, J. Pfeiffer, B. Rumpe, A. Wortmann: *Integration Challenges for Digital Twin Systems-of-Systems*. SESoS'22
- [MW21] J. Michael, A. Wortmann: *Towards Development Platforms for Digital Twins: A Model-Driven Low-Code Approach*. APMS'21
- [BBD+21a] T. Bolender, G. Bürvenich, M. Dalibor, B. Rumpe, A. Wortmann: *Self-Adaptive Manufacturing with Digital Twins*. SEAMS'21.
- [DMR+20] M. Dalibor, J. Michael, B. Rumpe, S. Varga, A. Wortmann: *Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits*. ER'20.
- [BDH+20] P. Bibow, M. Dalibor, C. Hopmann, B. Mainz, B. Rumpe, D. Schmalzing, M. Schmitz, A. Wortmann: *Model-Driven Development of a Digital Twin for Injection Molding*. CAiSE'20.
- [BHK21] T. Brockhoff, M. Heithoff, I. Koren, J. Michael, J. Pfeiffer, B. Rumpe, M.S. Uysal, W. M. P. van der Aalst, A. Wortmann: *Process Prediction with Digital Twins*. Models@runtime'21

Digital Shadows

- [BBD+21] F. Becker, P. Bibow, M. Dalibor, A. Gannouni, V. Hahn, C. Hopmann, M. Jarke, I. Koren, M. Kröger, J. Lipp, J. Maibaum, J. Michael, B. Rumpe, P. Sapel, N. Schäfer, G. J. Schmitz, G. Schuh, and A. Wortmann: *A conceptual model for digital shadows in industry and its application*. ER'21

MontiGem

- [DMM+22] I. Drave, J. Michael, E. Müller, B. Rumpe, S. Varga: *Model-Driven Engineering of Process-Aware Information Systems*. Springer Nature Computer Science, 2022.
- [DGM+21] I. Drave, A. Gerasimov, J. Michael, L. Netz, B. Rumpe, S. Varga: *A Methodology for Retrofitting Generative Aspects in Existing Applications*. JOT 20, 2021
- [AMN+20a] K. Adam, J. Michael, L. Netz, B. Rumpe, S. Varga: *Enterprise Information Systems in Academia and Practice: Lessons learned from a MBSE Project*. EMISA'19, 2020.
- [GMN+20] A. Gerasimov, J. Michael, L. Netz, B. Rumpe, S. Varga: *Continuous Transition from Model-Driven Prototype to Full-Size Real-World Enterprise Information Systems*. AMCIS'20

Current
Publications
se-rwth.de/publications