Modeling

The Swiss Army Knife of Engineering Methods

Dr. Judith Michael
Software Engineering
RWTH Aachen

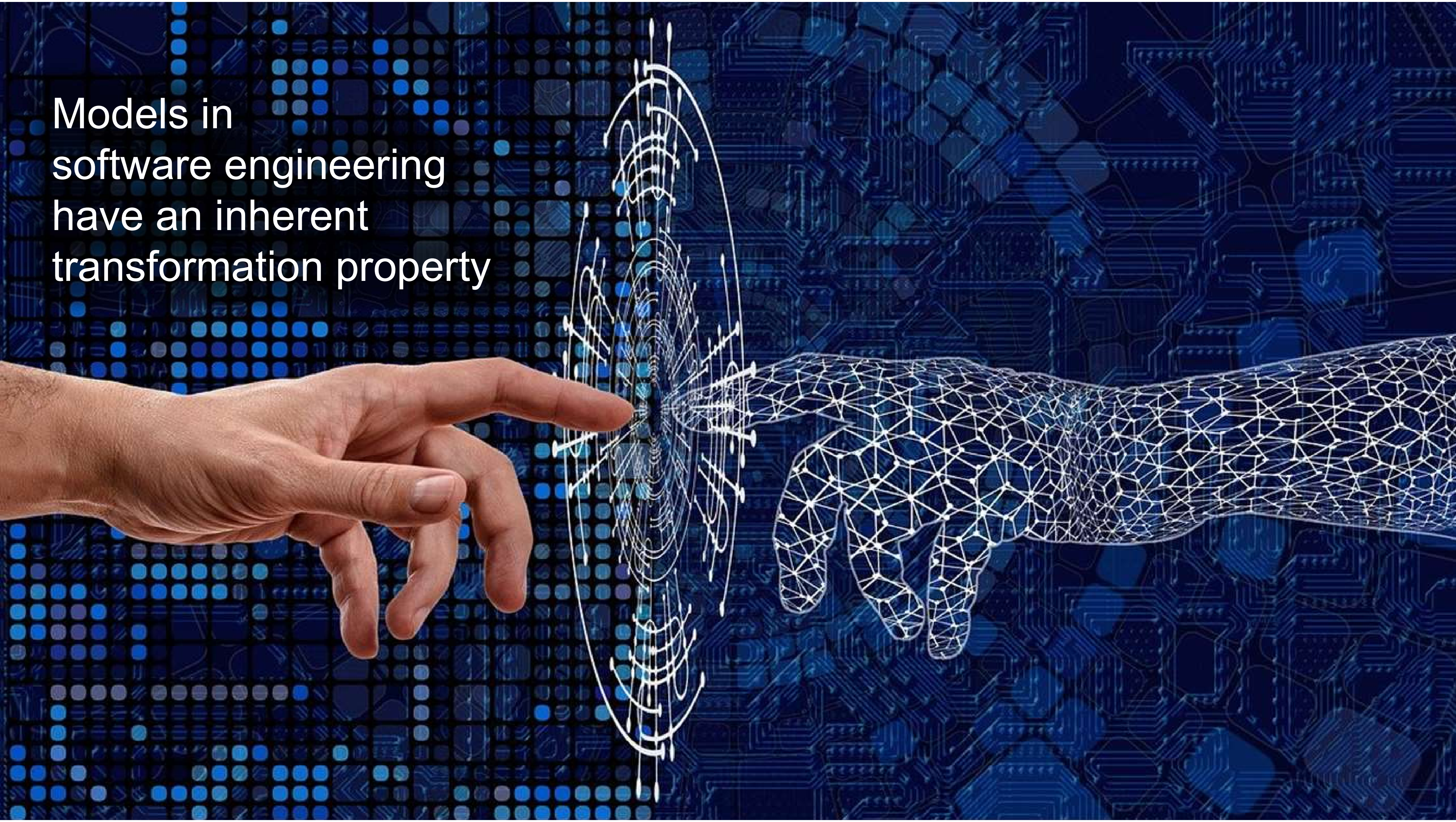
http://www.se-rwth.de
*@JudithMichael_*

…about research with Bernhard Rumpe and my colleagues from the Chair of Software Engineering

06.06.2023, Traunkirchen


Modelling - The Swiss Army Knife


Models serve different purposes!

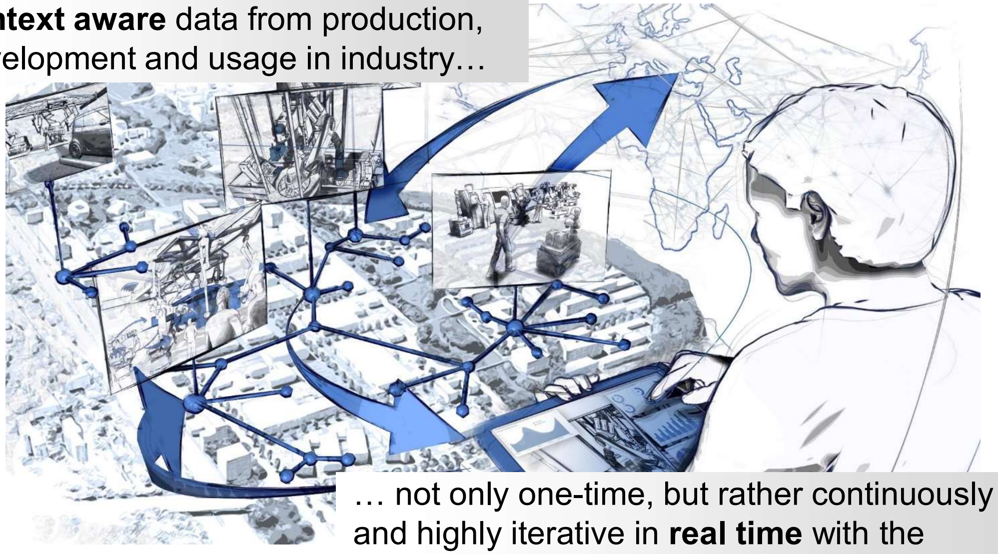

Models in
software engineering
have an inherent
transformation property

WHAT TO EXPECT FROM THIS TALK?

The Internet of Production (IoP) develops techniques for digital shadows and digital twins

Vision of the **IoP**

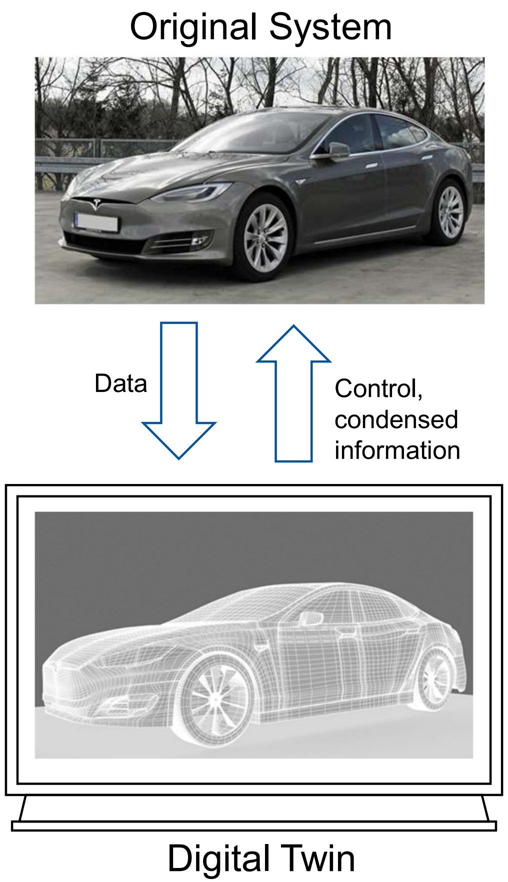
Providing **semantically adequate** and **context aware** data from production, development and usage in industry…



… not only one-time, but rather continuously and highly iterative in **real time** with the **adequate level of granularity**

- Central scientific approach of the IoP:
  digital shadows as mediators between the vast amounts of heterogeneous data and detailed production engineering models
  - Sufficiently aggregated, multi-perspective and persistent datasets
  - Generated by deliberate selection, cleaning, semantic integration and pre-analysis
  - Used for reporting, diagnosis, prediction and recommendation in domain-specific real-time
- The Internet of Production is huge:
  - 87,5 researchers (up to 2x7 years)
  - 13 research managers
  - 4 support positions
  - Overall app. 200 employees

Digital Twins as *complex, long-lasting, software-intensive systems*



contextual data and their aggregation and abstraction

A Digital Twin of a system consists of
- a set of models of the system and
- a set of digital shadows,
  - both of which are purposefully updated on a regular basis, and
- provides a set of services to use both purposefully with respect to the original system.

- The digital twin interacts with the original system by
  - providing useful information about the system's context and
  - sending it control commands.

Kinds of Engineering Models usable for a Digital Twin

- **Structural Models:** Representing relevant parts of the system-of-interest
  - The developed system
  - The environment of the developed system
  - Interactions between the developed system and the environment
- **Behavioral Models:** Describe a system's actions

UML/SysML/Ontology

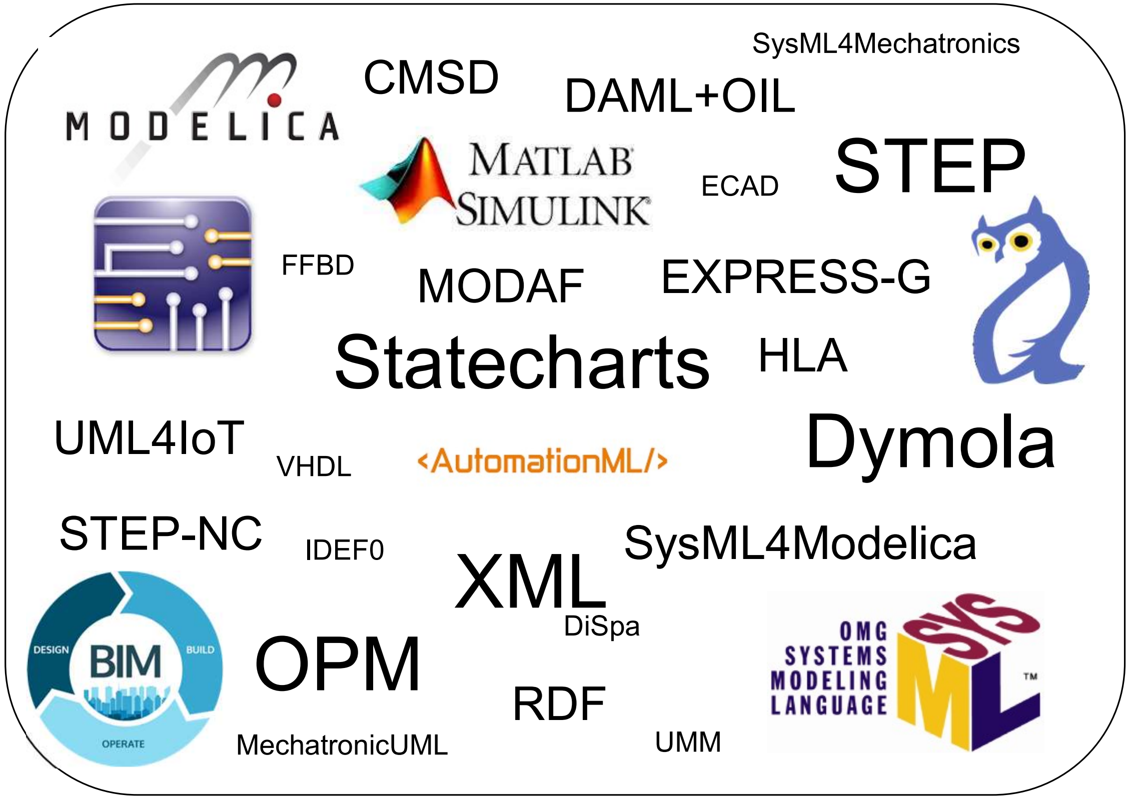
- **Physical Models:** Objects that are identical in the relevant attributes of the real system or similar, e.g., test bench
- **Geometrical Models:** Mathematical description of shapes
  - Procedural: Define shapes implicitly by an algorithm that generates the form
  - Digital Image: Represent shapes as a subset of a fine regular partition of space

STEP (ISO 10303)

- **Mathematical Models:** Expressions or numerical methods to convert input data into outputs with the same functional dependence as the actual system
  - Explain or prescribe system behavior

Modelling Languages in & for Systems Engineering

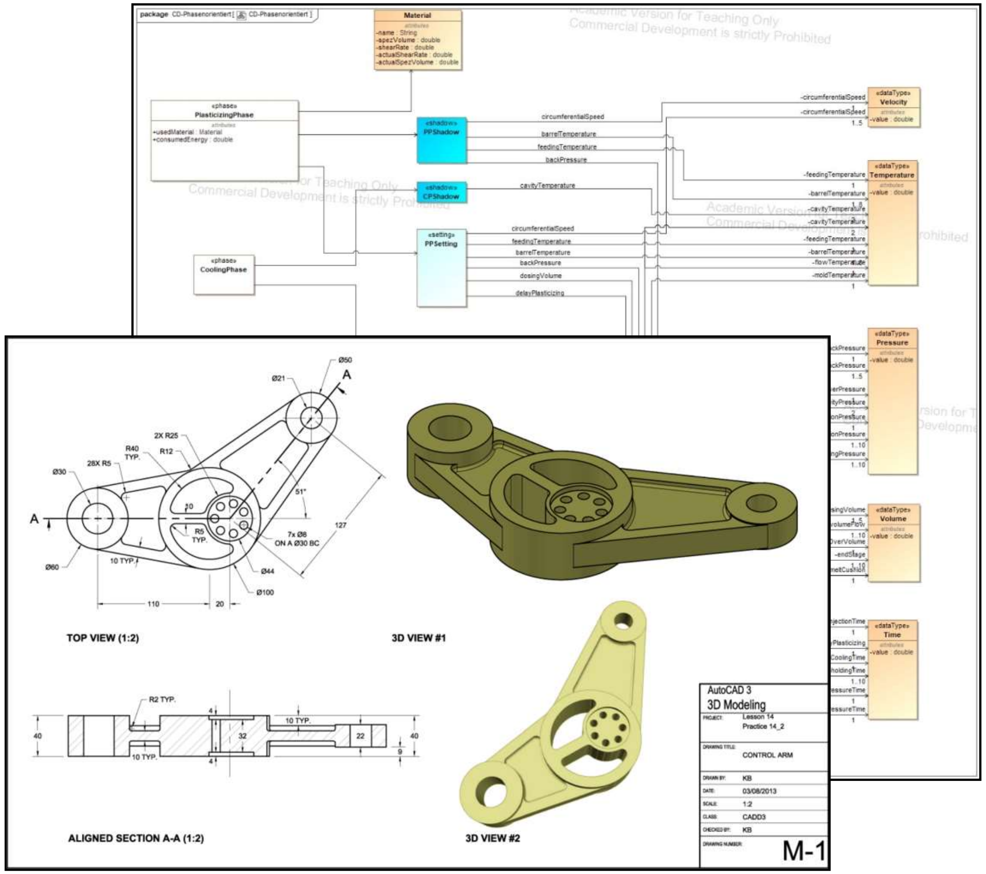
- Digitalization of engineering domains demands explicit languages
- Languages are a key for systems engineering, e.g.,
  - Physical modeling: Modelica, Simulink
  - CAD: STEP, NX CAD, ECAD
  - Simulation: Dymola
  - Knowledge: OWL, RDF
  - Integration: AutomationML
  - Circuits: VHDL
  - Building Information Models (BIM)

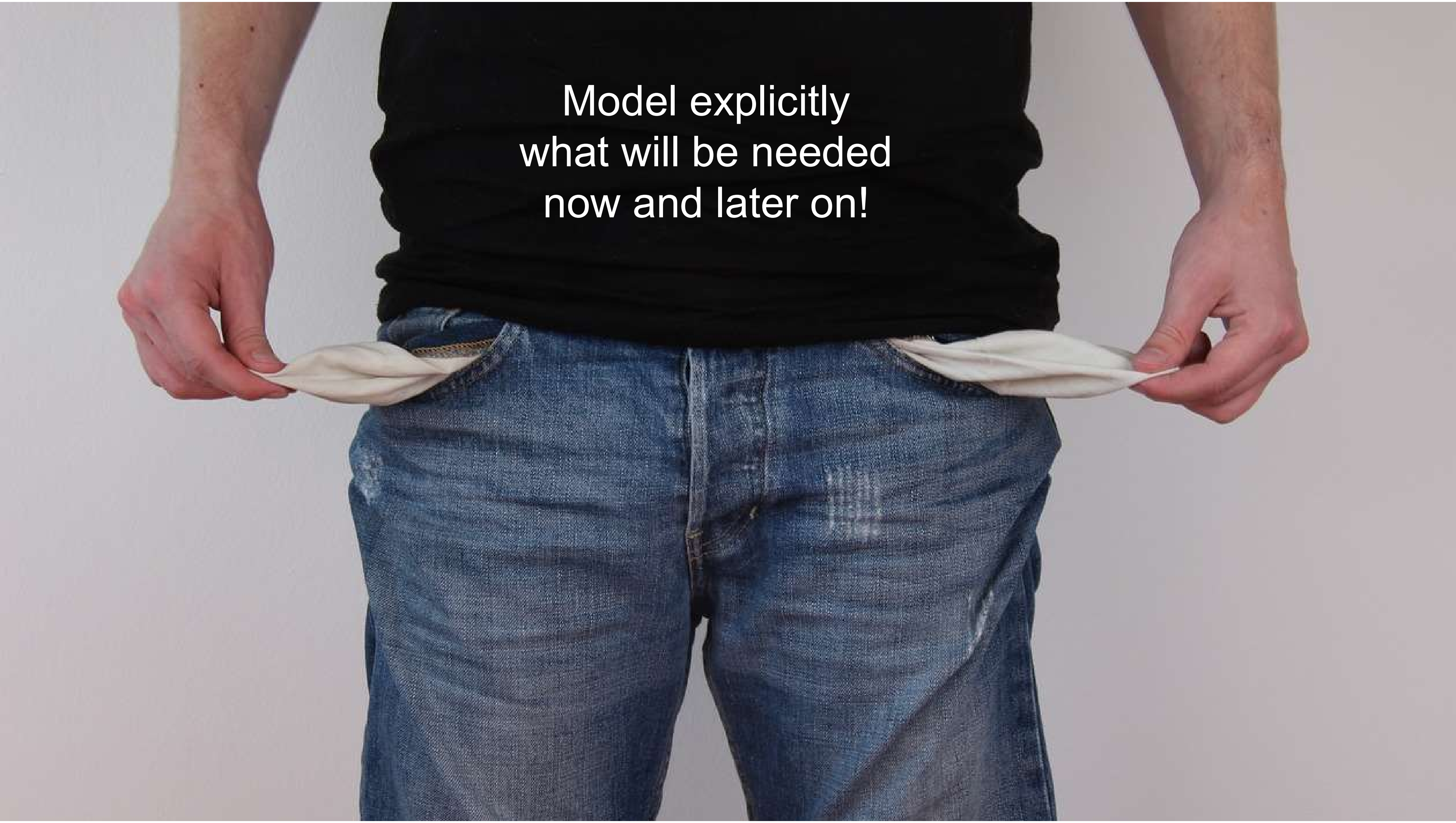


Reuse Engineering Models from System Design for Engineering Digital Twins

- Cyber-physical systems are complex
  - Consist of multiple components
  - Offer different functionalities
- Complexity reflected in their digital twins
  - Cover different functions and views
- Creating a Digital Twin requires
  - Domain knowledge about the physical system
  - Software engineering skills
  - Is time-consuming

Goal:
Reuse engineering models created during system design for systematic and efficient definition of larger parts of a Digital Twin



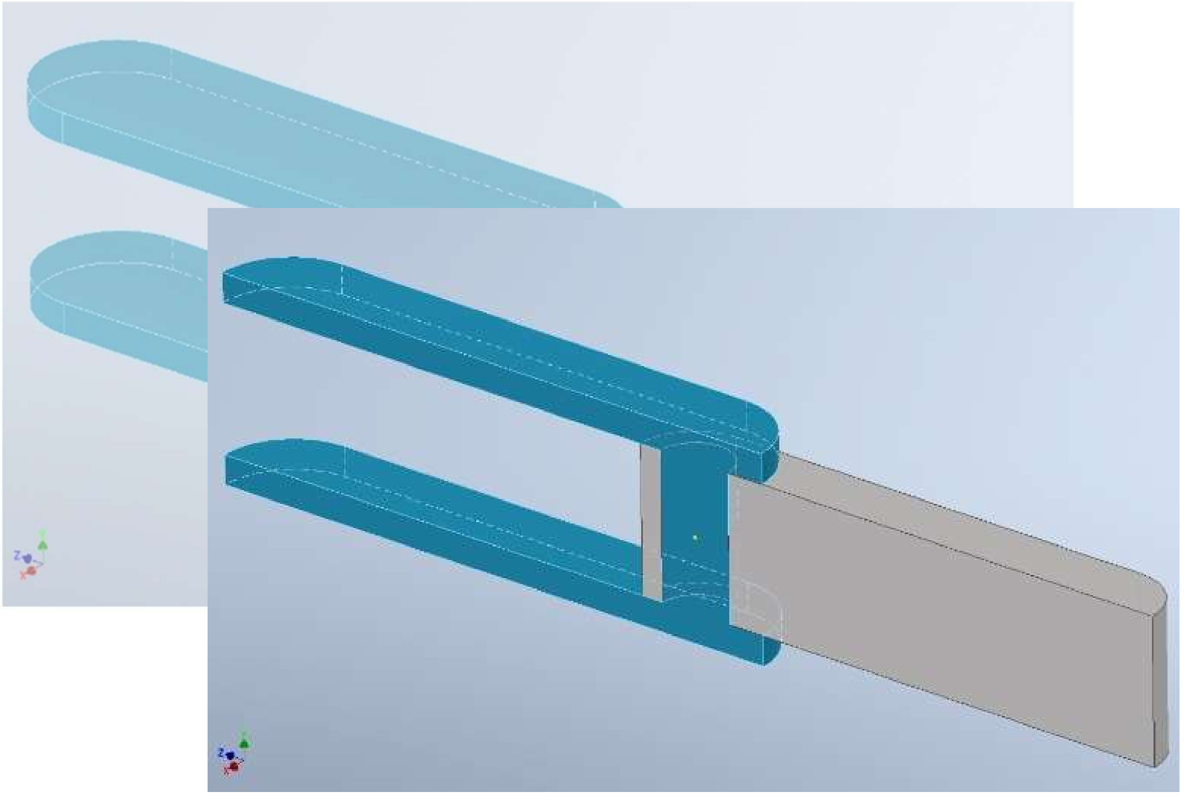

Model explicitly
what will be needed
now and later on!

Analyzing STEP Files for Deriving Digital Twins

- Different assemblies in CAD models form a functional unit

Challenge:
- Units are just an intellectual property of the domain experts, not the models
- Functional units often not reflected in the CAD model

Goal:
- Extracting contact points of different assemblies to detect and extract functional units
  - Ultimate goal: Construct Digital Twins with respect to functional interrelationships



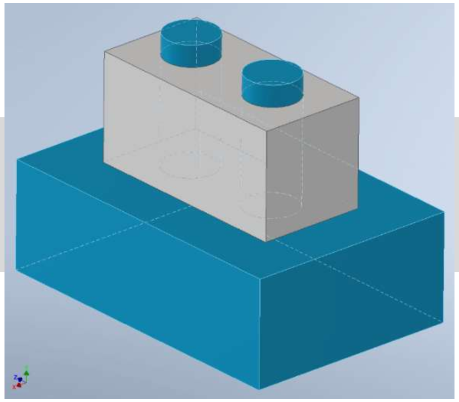
[CJW+22] B. Caesar, N. Jansen, M. Weigand, M. Ramonat, C. S. Gundlach, A. Fay, B. Rumpe: Extracting Functional Machine Knowledge from STEP Files for Digital Twins. ETFA 2022

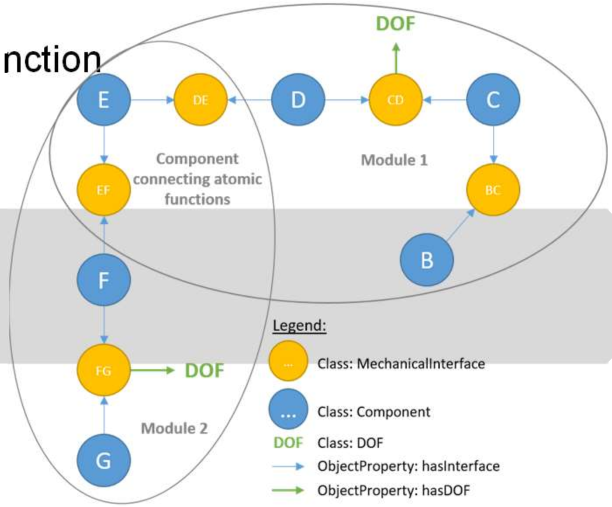
Ontology Mapping and Graph Analysis

1. Extraction of assemblies and mutual events between modeled concepts
   – translational and rotational constraints

2. Transformation of information into a graph based on an ontology design pattern (ODP)

3. Group the system components into functional modules
   – Atomic function enables the movement of two components relative to each other
   – Each module includes at least the components that contribute to fulfill one atomic function





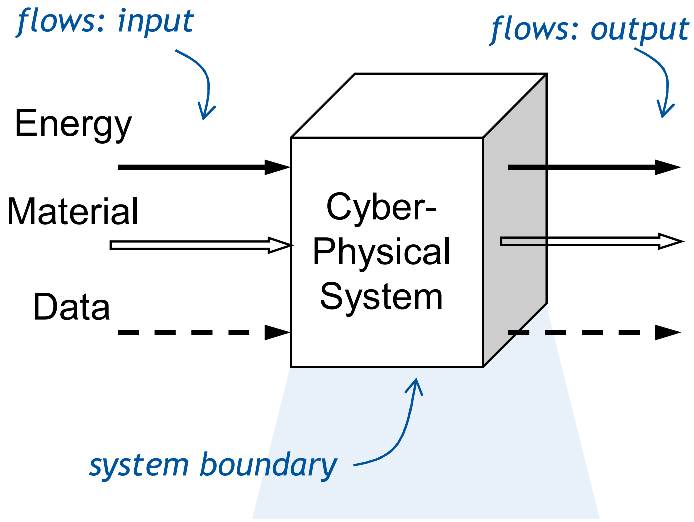
[CJW+22] B. Caesar, N. Jansen, M. Weigand, M. Ramonat, C. S. Gundlach, A. Fay, B. Rumpe: Extracting Functional Machine Knowledge from STEP Files for Digital Twins. ETFA 2022

System Specification as Function

- A system defines a function
  - it encapsulates a physical and computational structure
  - performs data, energetic and physical transformations
  - and is connected to its context through its interfaces.

- A system function is described through its input and output signature
  - types and forms of the
    - signals / data
    - energy flow
    - material flow

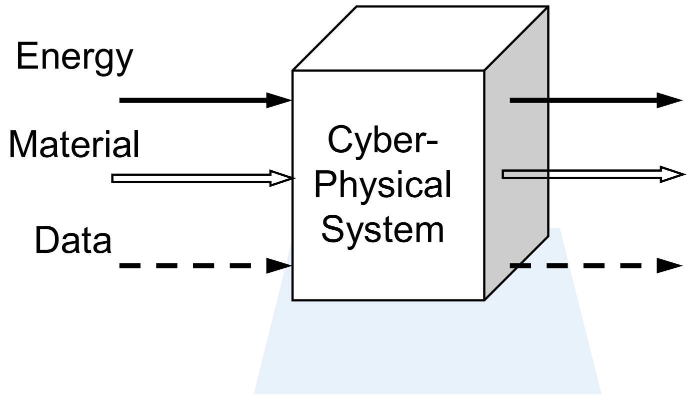
- The functionality is described through the
  - relation between input and output



⇨ This concept of function is our first universal specification and construction principle

The Underspecification Principle

- Deterministic and fully specified relations are normally not achievable
  - Delays happen
  - Energy fluctuates
  - Abstraction introduces lack of information
- Underspecification is the ability to describe the desired range of allowed behaviors (instead of a single, determined behavior)
- Advantages:
  - Easier to specify
  - Can be well combined with variant-building and methodical refinement



⇨ Controlled, explicit underspecification is the second universal specification principle

Composition

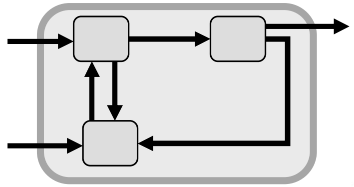
- Composition is an act or mechanism to combine simple elements to build more complicated ones

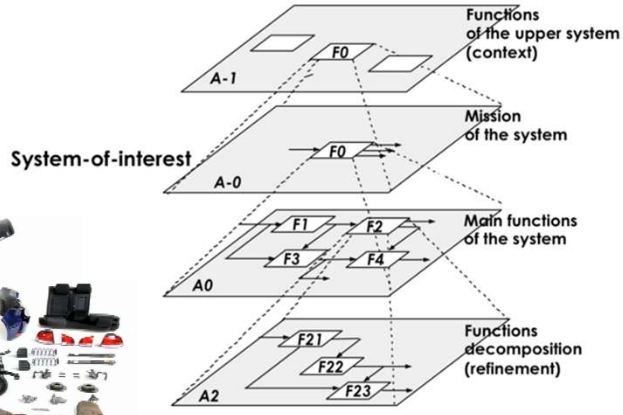
- Examples: function composition (math), product composition (mechanics), software composition (CS), …

- System is composed of components.

- Component is atomic or hierarchically composed of simpler components.

- Sub-system ~ not-atomic component





⇨ Composition is the third universal construction principle
It helps to manage complexity.

The Center for Systems Engineering integrates the experience concerning system and product development from different disciplines at RWTH Aachen university

WZL | RWTH AACHEN UNIVERSITY

Prof. Dr. Christian Brecher
Prof. Dr. Günther Schuh
Innovation Management and Production

MSE | RWTH AACHEN UNIVERSITY

Prof. Dr. Georg Jacobs
Systems Engineering

SE Software Engineering | RWTH AACHEN UNIVERSITY

Prof. Dr. Bernhard Rumpe
Software Engineering

Systems Engineering experts from different companies, branches and disciplines are part of the Center for Systems Engineering

BMW GROUP
bürkert FLUID CONTROL SYSTEMS
Complexity Management Academy
CONTACT Software
DASSAULT SYSTEMES
ELISE
evolut-on
Hettich
itemis
IME Aachen GmbH
kampf Jagenberg Group
KION GROUP
MAN Energy Solutions Future in the making
MR
M.TEC
ptc
Schuh & Company
SIEMENS
SIEMENS Healthineers
Vestas
ZF

We bring together interdisciplinary experts to shape the development of tomorrow – learning together and from each other in five benefit categories

CENTER
SYSTEMS
ENGINEERING

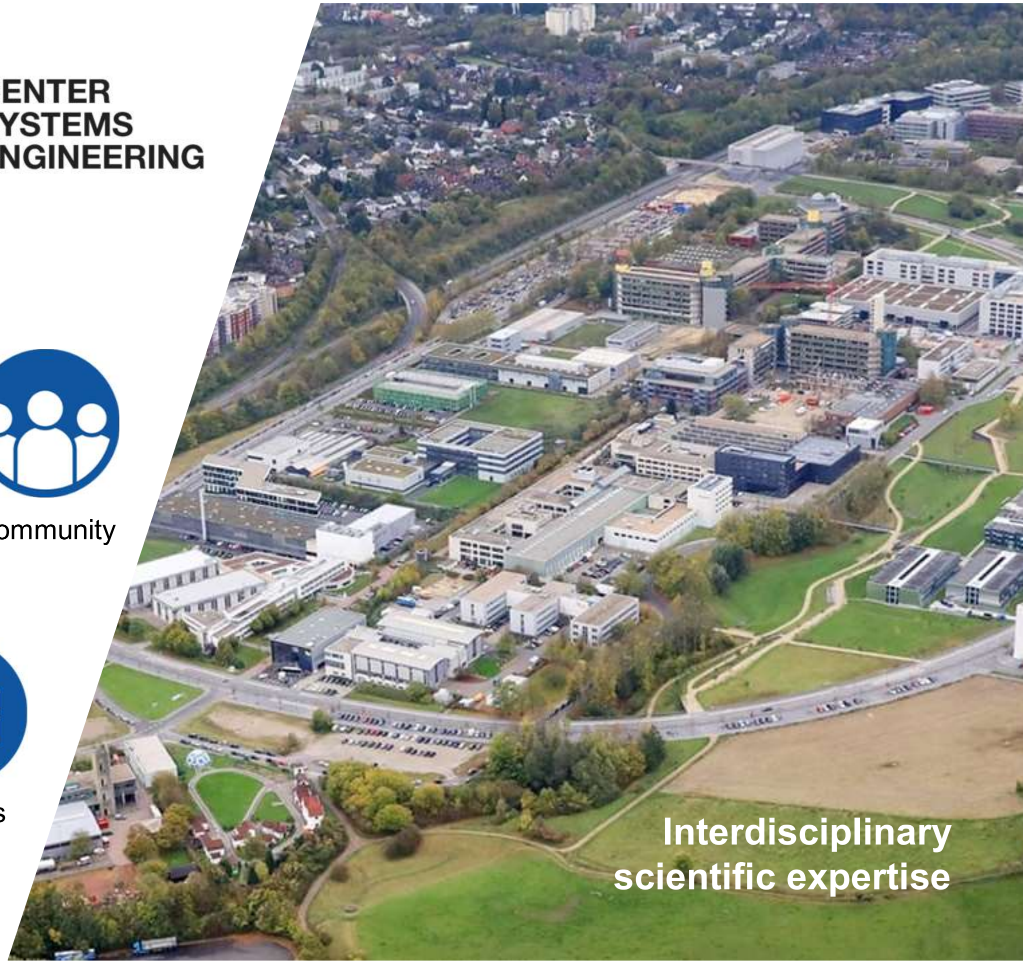
Membership

Community

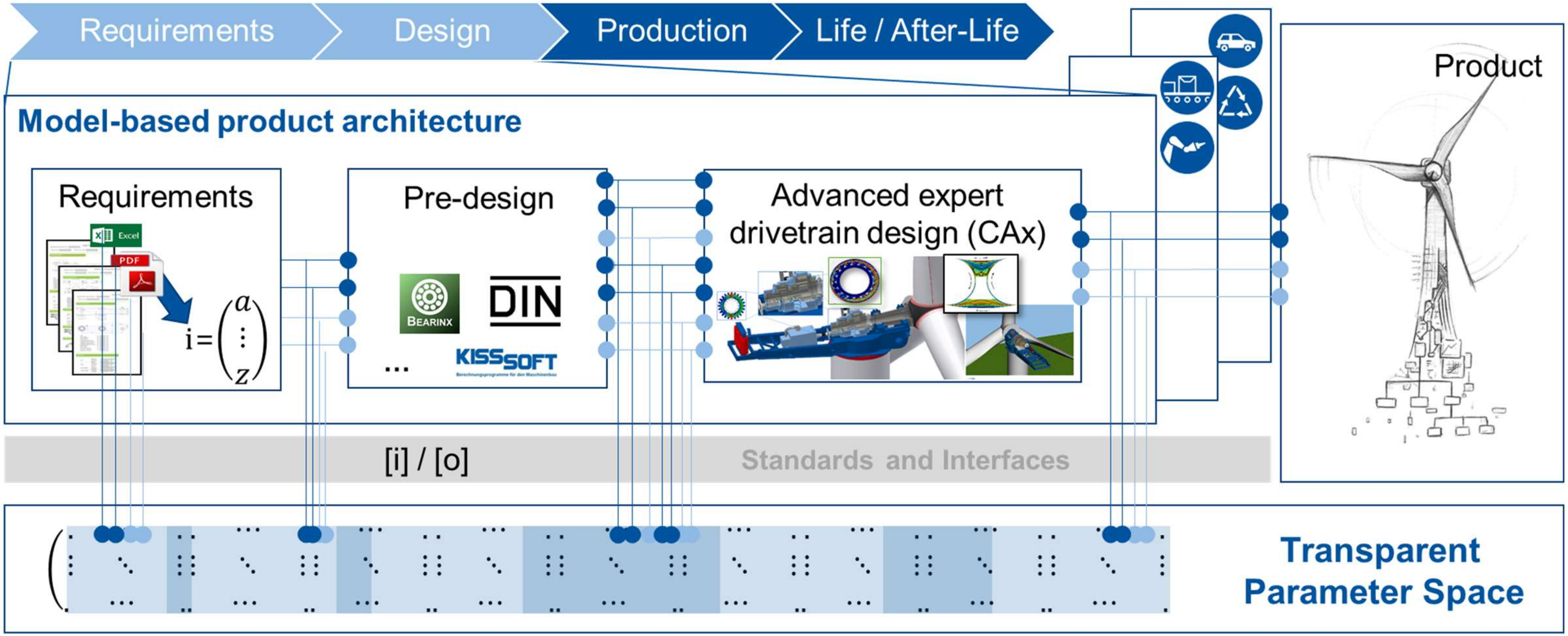
Research & Development

Services

Further Education

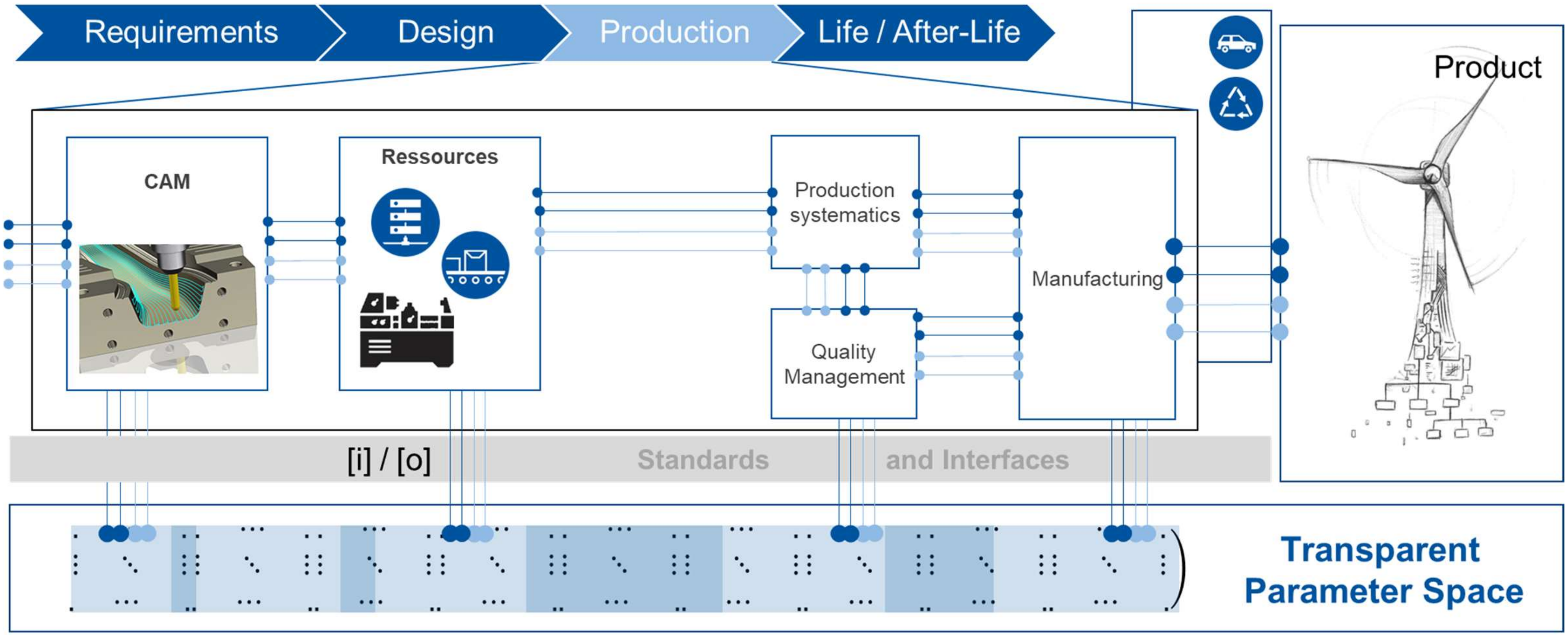


MBSE – Parameter triggered processes by global transparent modelling of the entire design process (I/II)



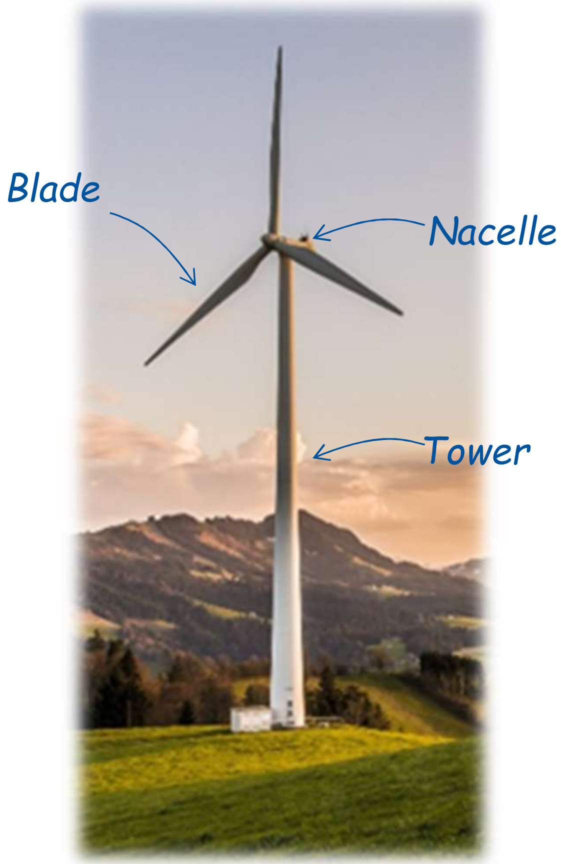
MSE

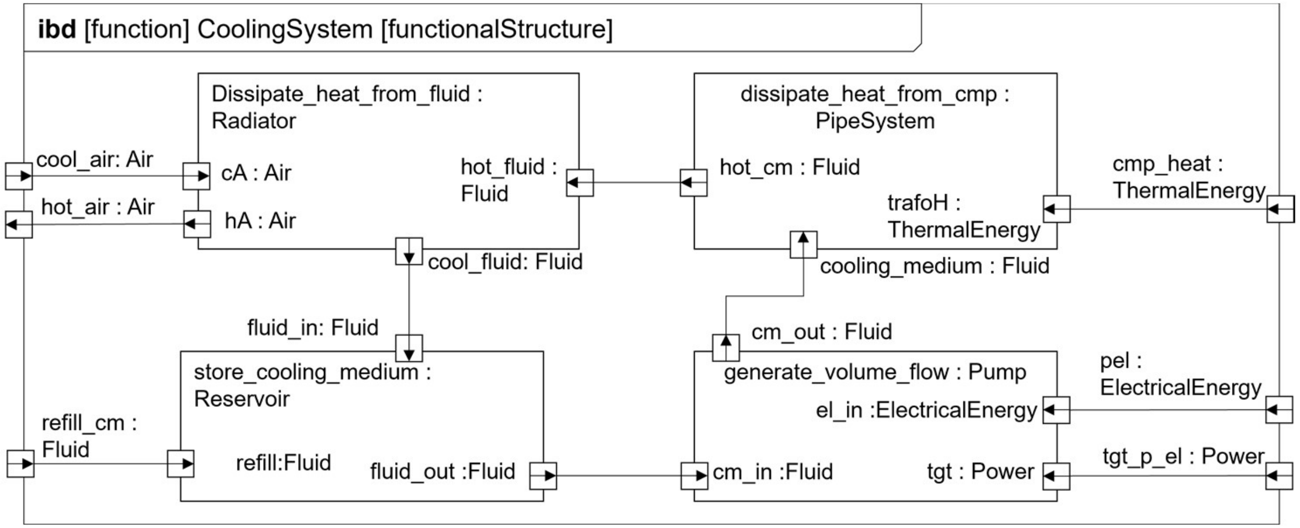
MBSE – Parameter triggered processes by global transparent modelling of the entire design process (II/II)



MSE

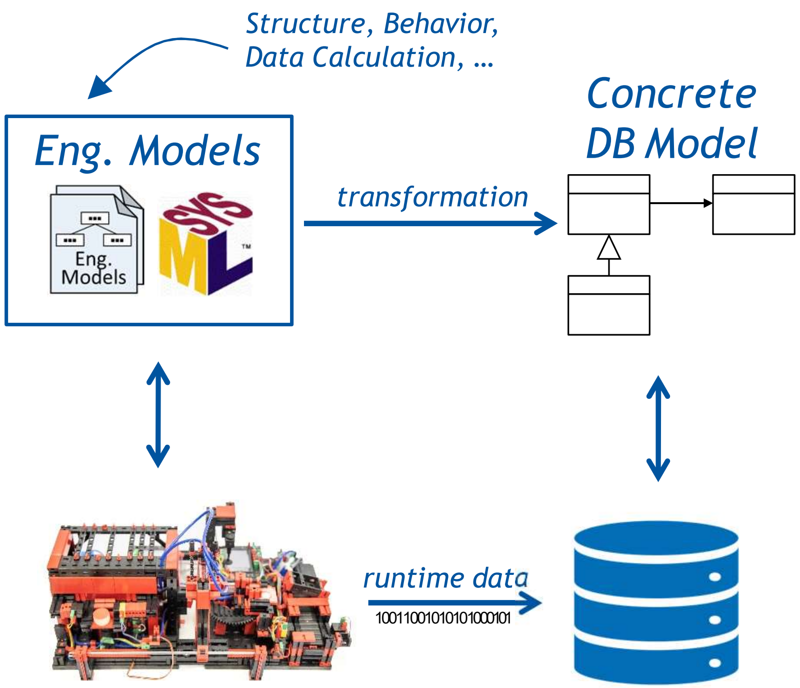
Digital Twin Cockpit for the Parameter Management in the Engineering of Wind Turbines

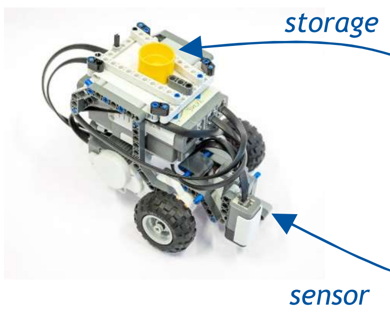




From Systems Modeling to Data Structures

- Example from the Fischertechnik Factory demonstrator



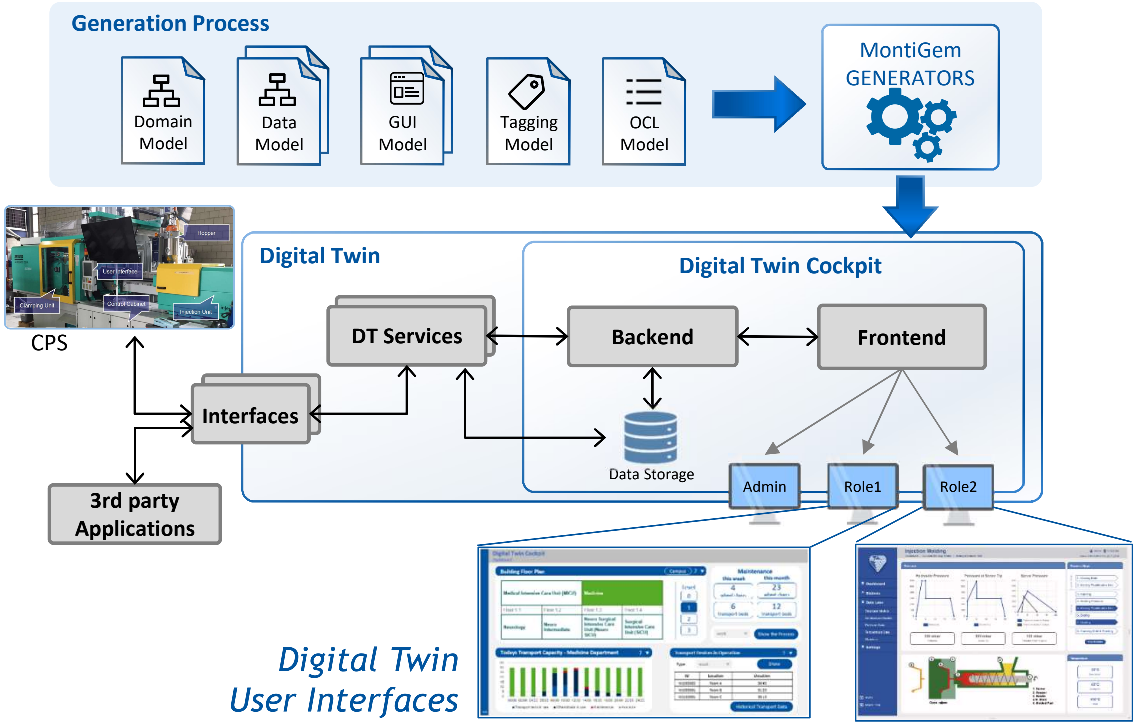


SysML

```
package Vehicles{

  import Items::Cup;
  import Stations.Station;

  part def Wheel;
  part def Engine;

  part def Vehicle{
    attribute func: Function;
    attribute reachedDestination:boolean;
    part wheels:Wheel[2..8];
    part engine:Engine;
    part control:Controller;

    bind reachedDestination = control.TargetPort.isFinished;
  }

  part def TransportCart specializes Vehicle {
    attribute function redefines func = Function.transport;
    part cartWheels:Wheel[4] subsets wheels;
    part def Storage{
      attribute capacity:int;
      port cupIn:CupPort;
      port cupOut:~CupPort;
    }
  }

  part def Sensor{
    attribute sensedObstacle:boolean;
    port sensorPort:SensorPort;
  }

  part def Controller{
    attribute target:String;
    bind target = TargetPort.station.position;

    port def TargetPort{
      attribute isFinished:boolean;
      in item station:Station;
    }
  }
```

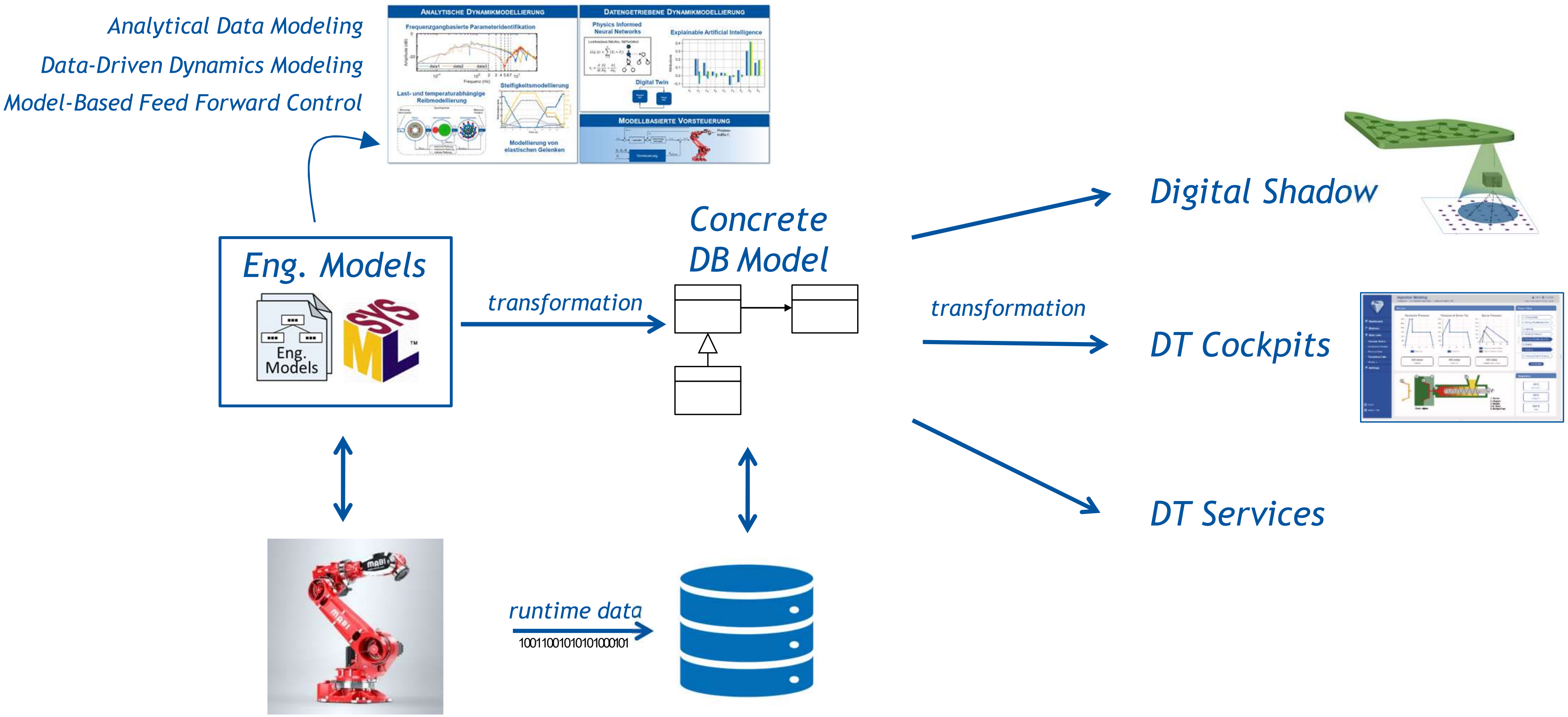
Creating Digital Twin Cockpits with MontiGem [DMR+20]

- 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]



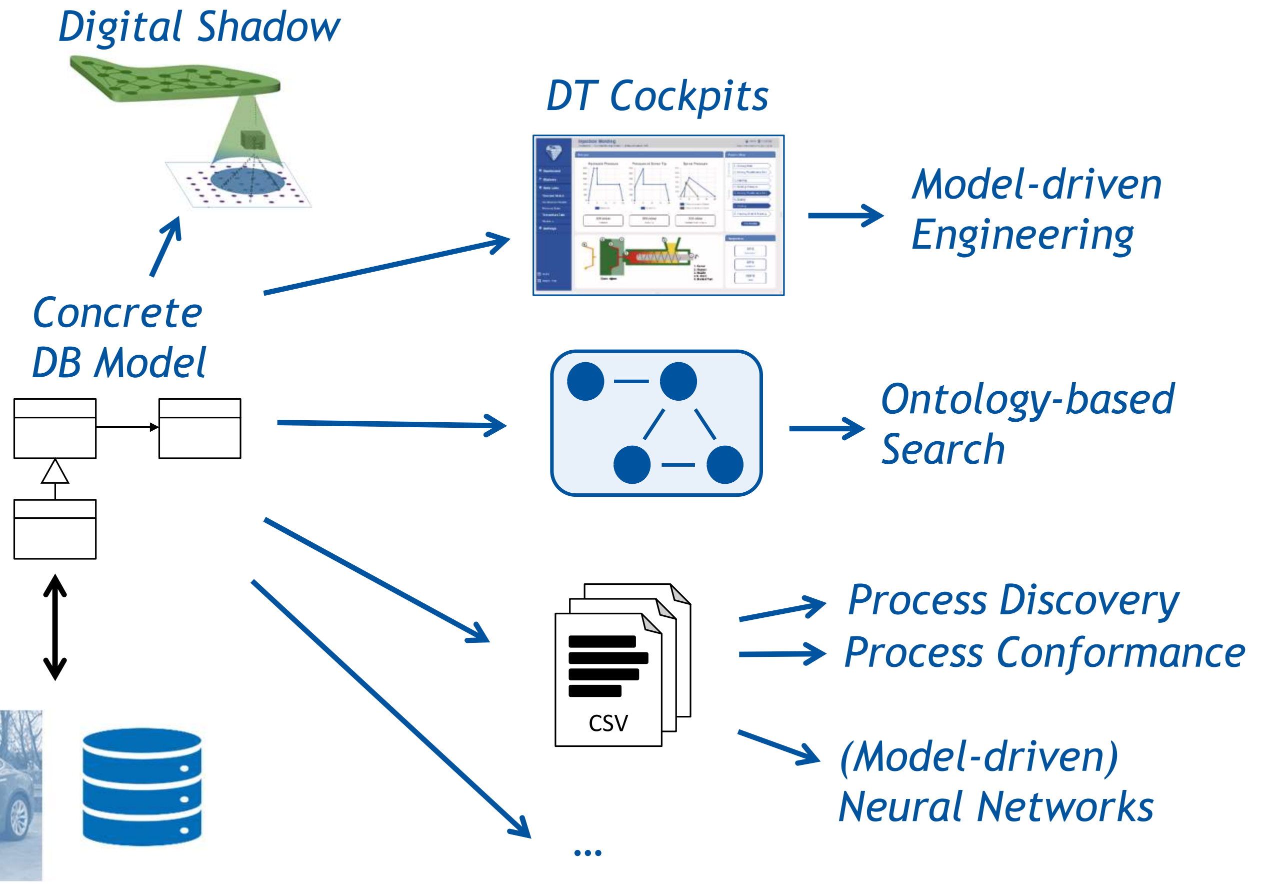
[DMR+20] M. Dalibor, J. Michael, B. Rumpe, S. Varga, A. Wortmann: Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits. ER'20

Transformations from 6-Axis Robot Engineering Models into parts of a Digital Twin



From Data Models to parts of Digital Twins

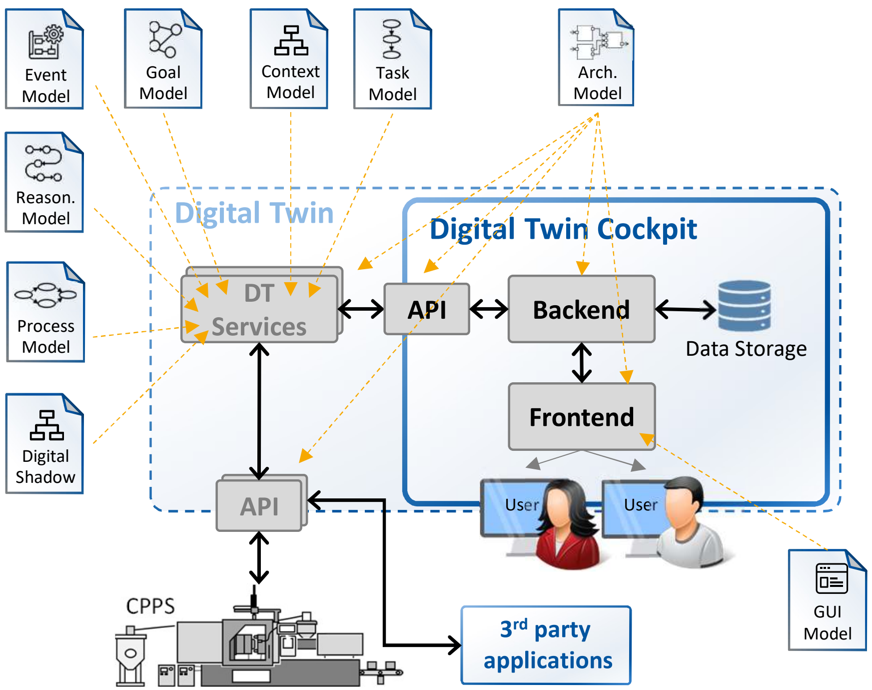
- Builds the foundation for derivation of
  - Digital Shadows
  - DT Cockpits
  - Semantic annotation
  - Digital Twin Services
    - Process Mining:
      Discovery & Conformance Checking
    - Neural Networks



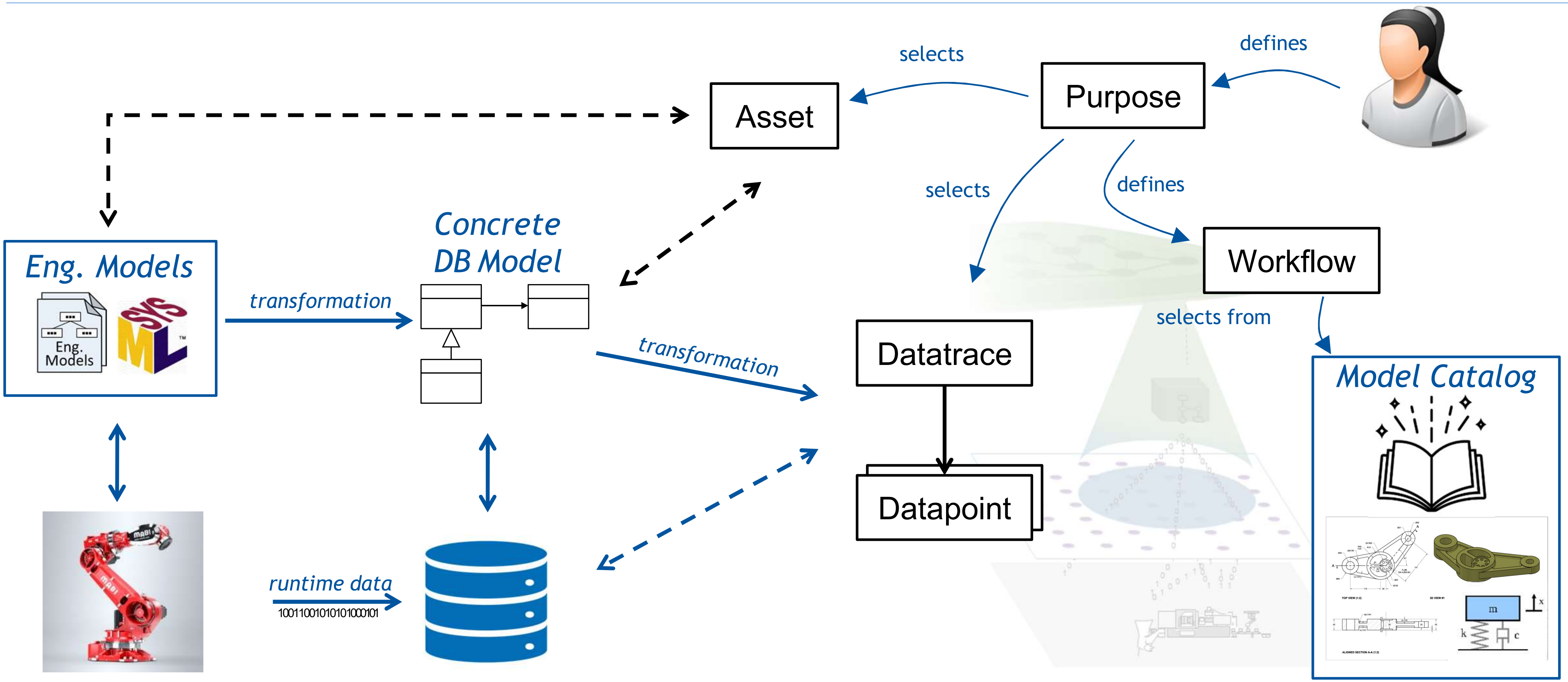
Use models about system behavior during runtime of a system!

Models@run.time in Digital Twins

- Examples
  - Aggregate and abstract *digital shadows* [BBD+21] from live data
  - *Process models* [BMR+22] to describe the CPPS behavior or human-CPPS interaction
    - e.g., derived using process discovery [BHK+21]
  - *Context models* to capture context data [MR23]
  - *Task models* to support human behavior [MR23]
  - Goal models to describe wished states [MRZ21]
  - Event models [DMR+20, DHM+22] to describe events of a system and possible actions to reach a certain state
  - Reasoning models,…

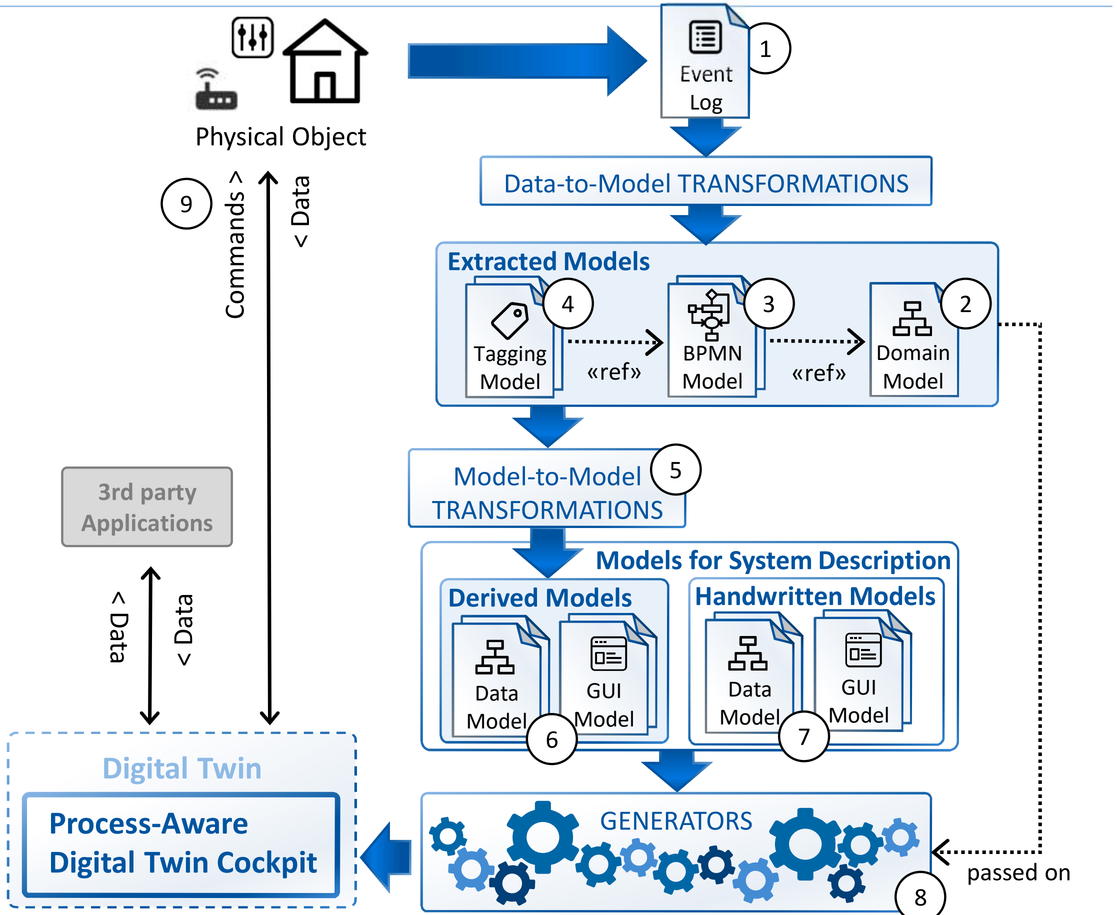


From System Models to Data Models and Digital Shadows



Extraction of Models from Sensor Data and Event Logs [BMR+22]

- Phase 1: Preparation
  - Extract event log (1) from sensor data of a physical object
  - Discover (2) domain information, (3) process models, (4) roles
  - Results: Domain CD, BPMN models, a tagging model
- Phase 2: Generation
  - Models (2,3,4) as input for (5) model-to-model transformation
  - Output: data models (views), GUI models (6)
  - (8) gen. PADTC source code
- Phase 3: Adaption
  - Add handwritten models (7), and handwritten code
- Phase 4: Runtime
  - PADTC connected to DT services
  - Live data (9) from the physical object or third-party applications & DT influence the physical object via commands
  - Domain users: interaction

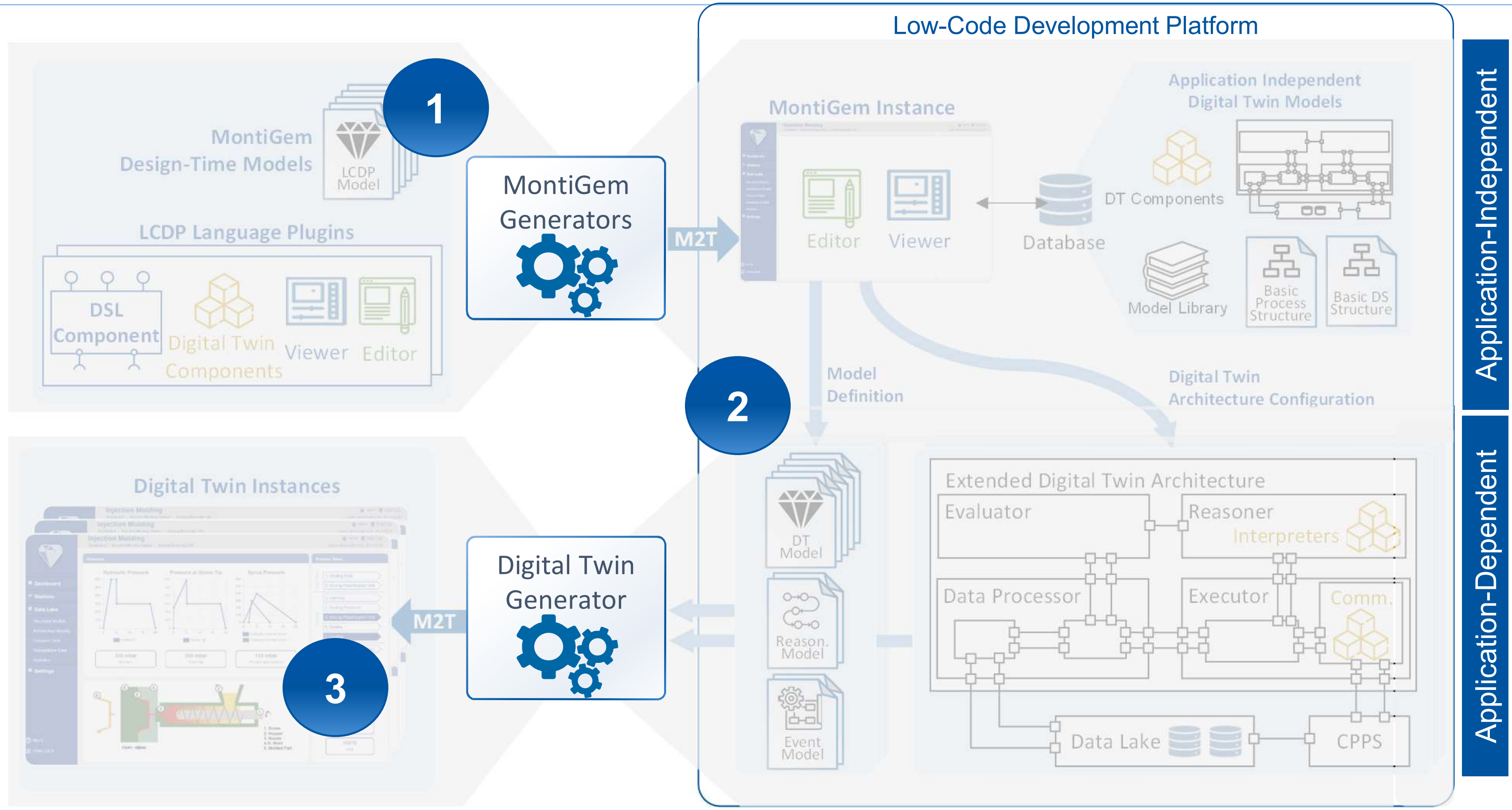


[BMR+22] D. Bano, J. Michael, B. Rumpe, S. Varga, M. Weske: Process-Aware Digital Twin Cockpit Synthesis from Event Logs. In: Journal of Computer Languages (COLA), Volume 70, Elsevier, 2022.

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

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 DTs
3) *Domain experts* operate on DTs

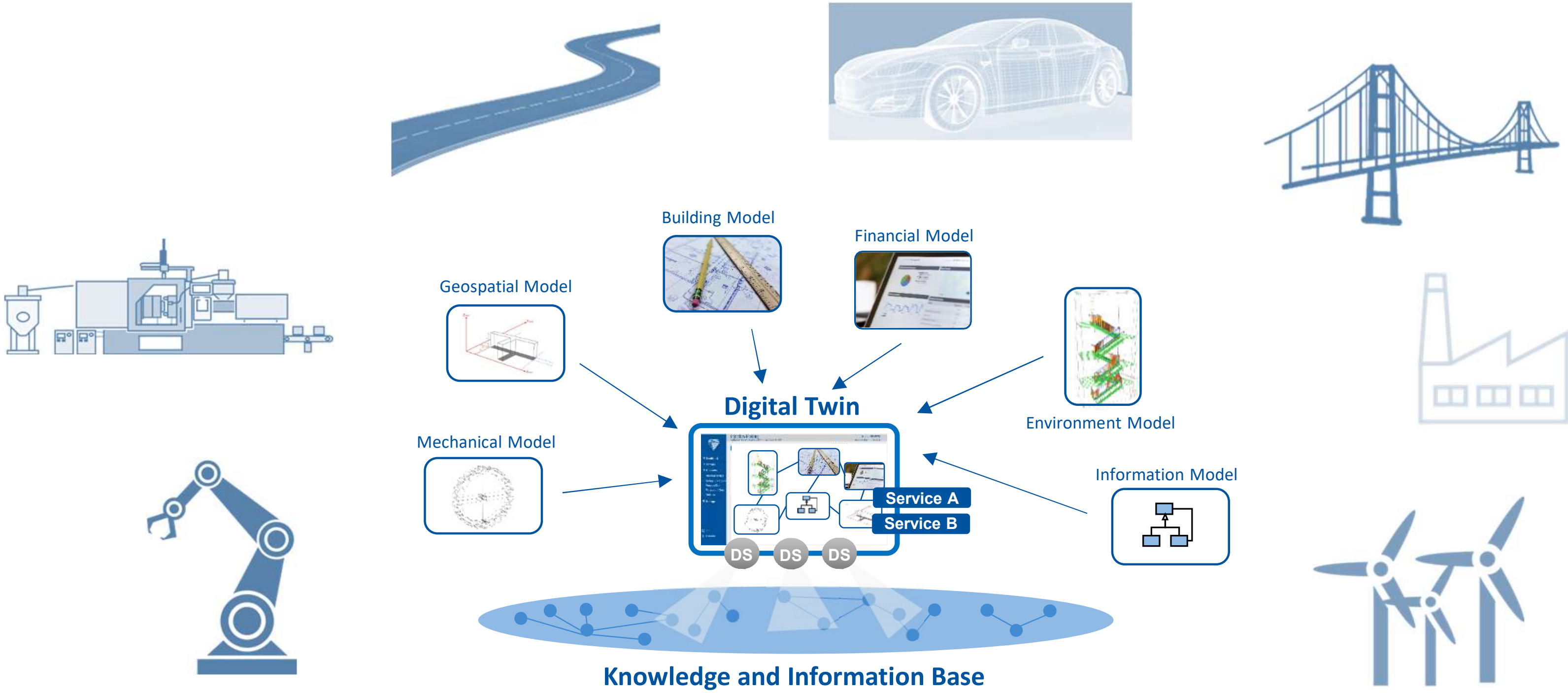


[DHM+22] M. Dalibor, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, S. Varga, A. Wortmann: Generating Customized Low-Code Development Platforms for Digital Twins. COLA 70, 2022

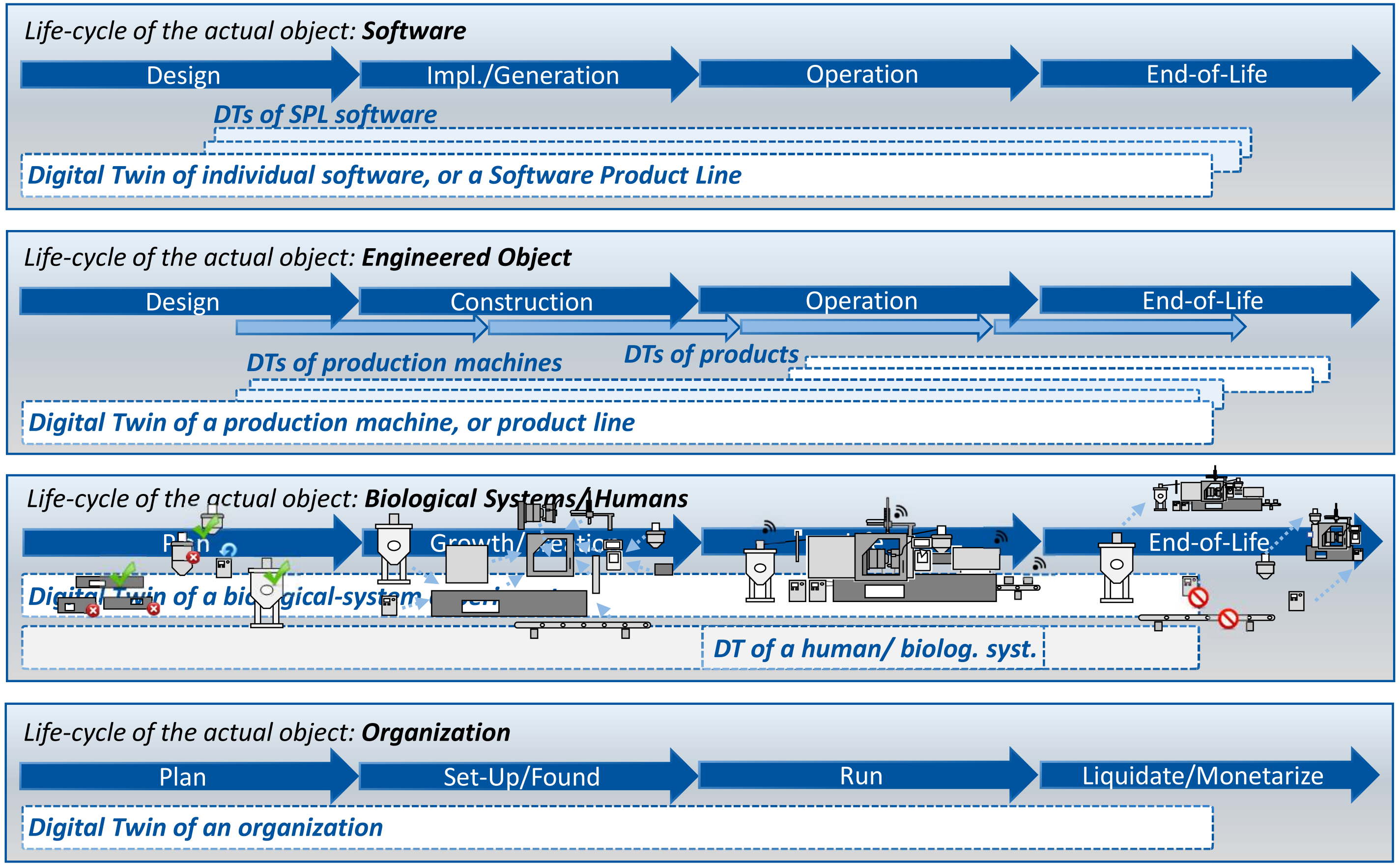
Future?

Creating digital twins for …

… elements in the physical world
that can be monitored, sensed, actuated and controlled
with a ***long lifespan***

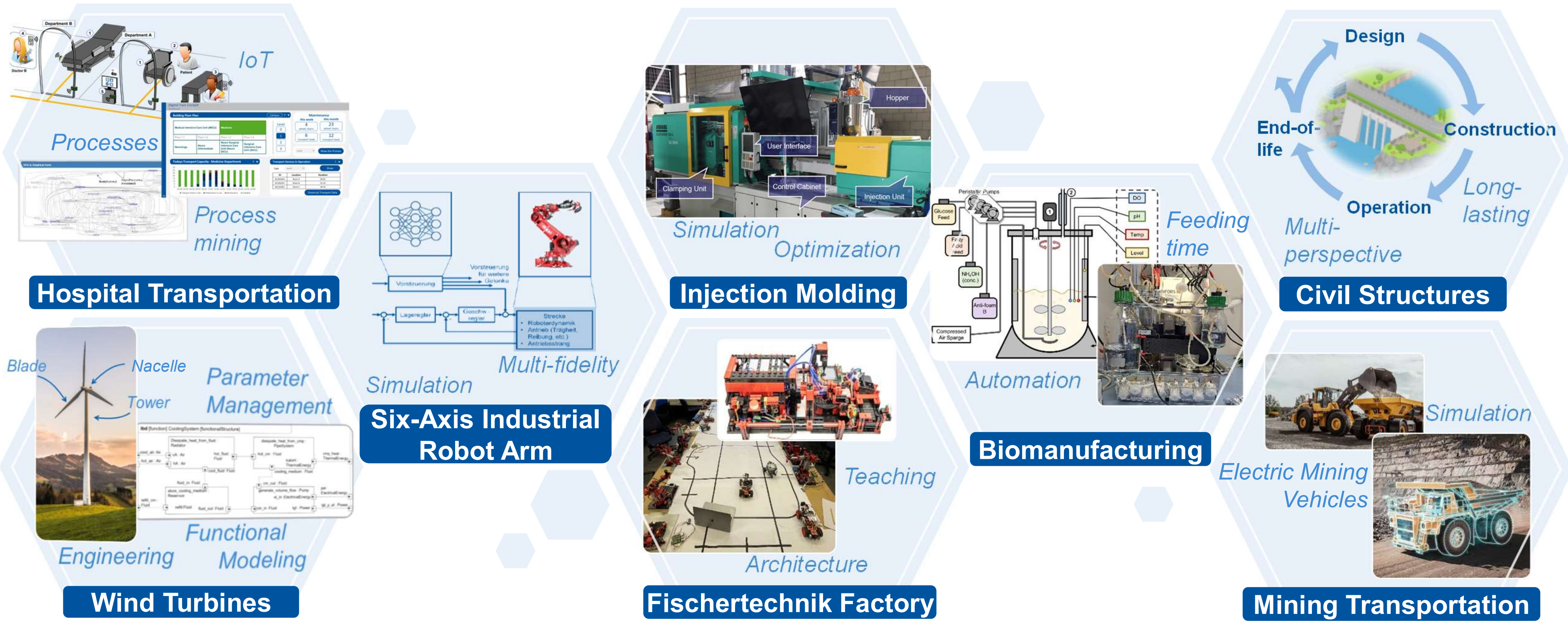


Digital Twins *in various application domains*




Human Factors?

Our Main Use Cases



GET THE MOST OUT OF YOUR SYSTEM MODELS!

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
- [FMR+22] K. Feichtinger, K. Meixner, F. Rinker, I. Koren, H. Eichelberger, T. Heinemann, J. Holtmann, M. Konersmann, J. Michael, E.- M. Neumann, J. Pfeiffer, R. Rabiser, M. Riebisch, K. Schmid: *Industry Voices on Software Engineering Challenges in Cyber-Physical Production Systems Engineering*. In: ETFA'22, IEEE, 2022.
- [Mic22] J. Michael: *A Vision Towards Generated Assistive Systems for Supporting Human Interactions in Production*. Modellierung'22

Digital Twins

- [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
- [DMR+20] M. Dalibor, J. Michael, B. Rumpe, S. Varga, A. Wortmann: *Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits*. ER'20.
- [KMR+20] J. C. Kirchhof, J. Michael, B. Rumpe, S. Varga, A. Wortmann: *Model-driven Digital Twin Construction: Synthesizing the Integration of Cyber-Physical Systems with Their Information Systems*. MODELS'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
- [MKD+23] J. Michael, I. Koren, I. Dimitriadis, J. Fulterer, A. Gannouni, M. Heithoff, A. Hermann, K. Hornberg, M. Kröger, P. Sapel, N. Schäfer, J. Theissen-Lipp, S. Decker, C. Hopmann, M. Jarke, B. Rumpe, R. Schmitt, G. Schuh: *A Digital Shadow Reference Model for Worldwide Production Labs*. In: Internet of Production: Fundamentals, Applications and Proceedings, Springer, 2023.

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
- [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

Software Engineering Chair

- Language Workbench MontiCore
- Generator Framework MontiGem
- Model-based Systems Engineering
- SysML, UML
- Logics-based AI

Current
Topics
se-rwth.de/topics

Research Overview

Automotive / Cyber-Physical Systems

- Autonomous Driving
- Simulation
- Deep Learning
- Data Science
- Modeling Embedded Systems
- Test-Driven Development
- Requirements Engineering

Model-Driven Systems Engineering

- Systems Modeling Languages
- Digital Twins
- Domain-Specific Application
- Software Architectures
- Semantic Tool Integration
- Variability & Product Lines
- Industry 4.0, CPPS, Robotics

Model-Based Assistance and Information Services

- Models in Information and Workflow Systems
- Human-Centered Assistance
- Digital Transformation
- DSLs for Rights, Roles, Privacy
- Digital Twin Cockpits
- Behavior/ Process Modeling

Modeling Language Engineering

- Development Tools
- Language Workbench MontiCore
- UML, SysML, Architecture DL
- Domain-specific languages (DSL)
- Generation, synthesis
- Testing, analysis, verification
- Software architecture, evolution
- Agile methods
- Compositional DSL engineering

Chair of Software Engineering
Computer Science 3
RWTH Aachen University

Ahornstraße 55
D-52074 Aachen

Prof. Bernhard Rumpe
rumpe@se-rwth.de

Dr. Judith Michael
michael@se-rwth.de