Modelling for and of Digital Twins

Bernhard Rumpe
Software Engineering
RWTH Aachen

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

Bernhard Rumpe

Judith Michael

Malte Heithoff

Nico Jansen

Steffen Hillemacher

… and formerly also Andreas Wortmann

12.10.2021, Aachen

RWTH: Facts and Figures

45.377	students
9.651	international students
10.866	students in the first core semester
547	professors
9.496	staff (FTE)
948,0	total financing volume in Mio. €
10.224	publications

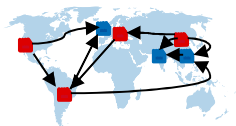
as of: 09/2018

Mission Statement of Our Group

- Improving software & systems development

- By identifying…

  methods, concepts, tools and infrastructures for
  - efficient, highly iterative development
  - of software intensive and high-quality products
  - in less time while
  - incorporating rapidly evolving requirements.



PEP definition tool

HW/SW product line

autonomous car

scientific model languages

electric car

- And we look for industrial strength evidence of success through:
  - development of complex software/systems in various problem domains
  - design of domain specific languages
  - architectural analysis and evaluation
  - model-based development methods and processes
  - realization of smart development tools, plugins, add-ons

Software Engineering Chair

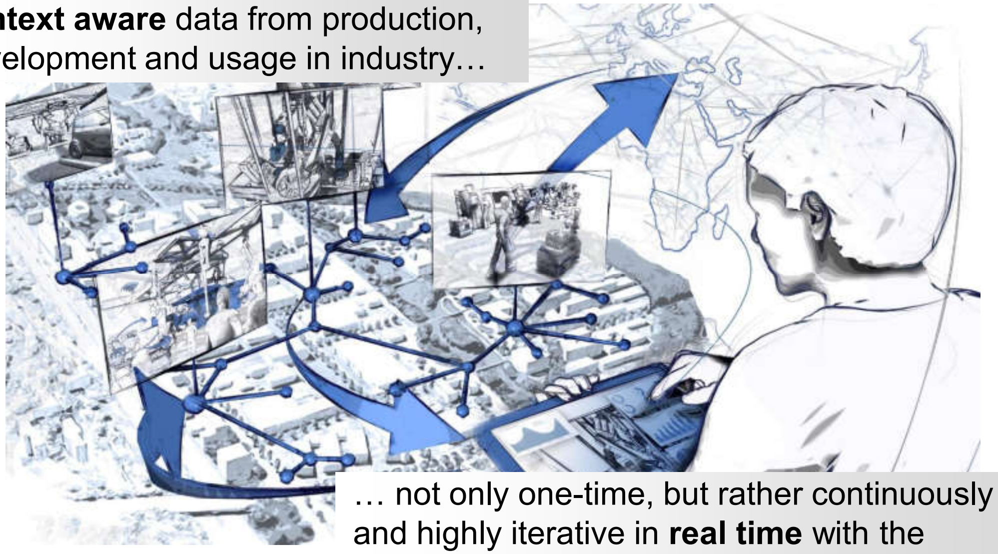
- Language Workbench MontiCore
- Generator Framework MontiGem
- Model-based Systems Engineering
- SysML, UML
- Logics-based AI

Current
Topics
se-rwth.de/topics

The Internet of Production develops techniques for digital shadows and digital twins

The vision of the **Internet of Production:**

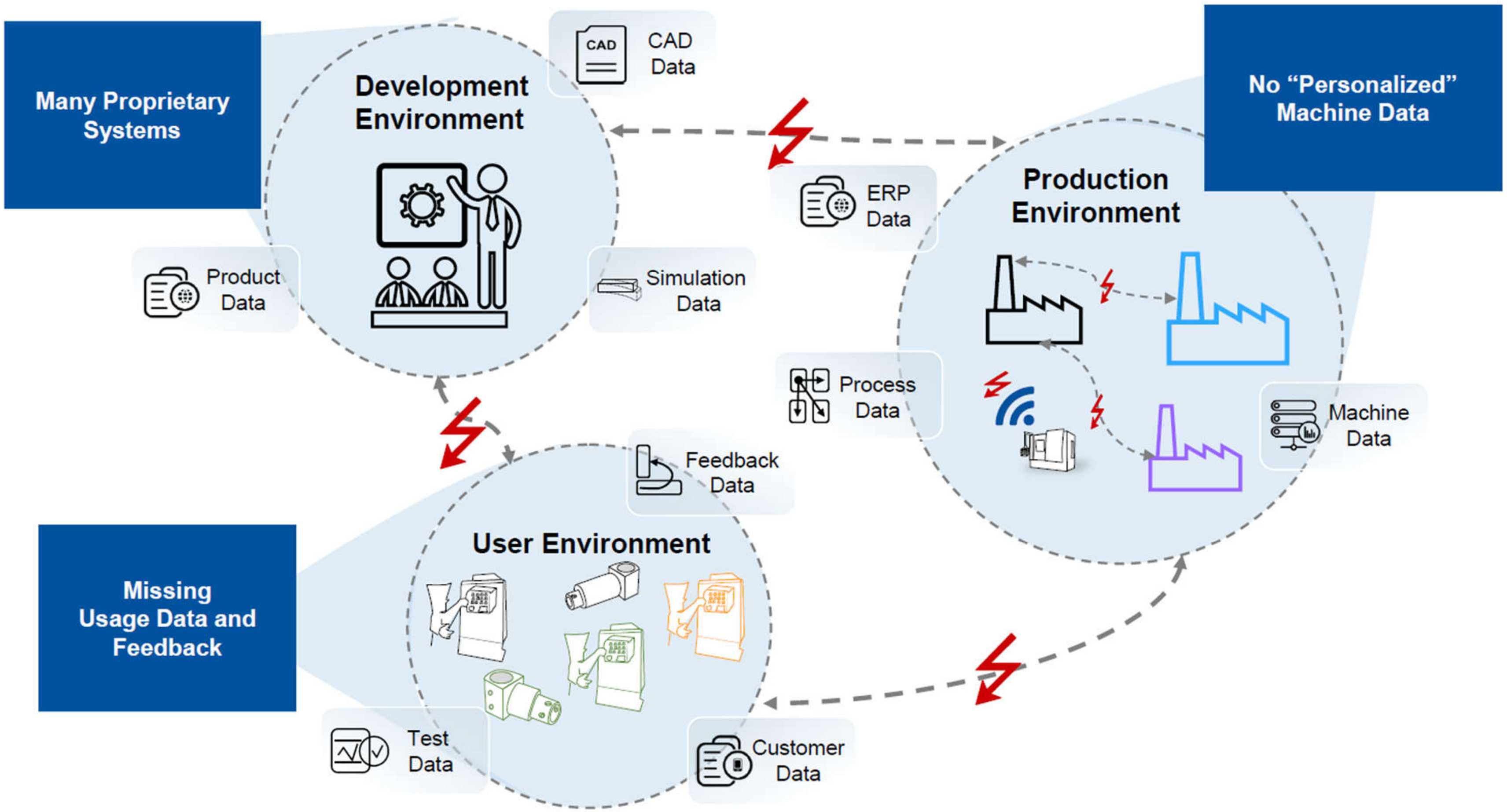
Providing **semantically adequate** and **context aware** data from production, development and usage in industry…



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

- The central scientific approach of the IoP are **digital shadows** as mediators between the vast amounts of heterogeneous data and detailed production engineering models, meaning:
  - 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

Data and Models must become available for cross-domain use



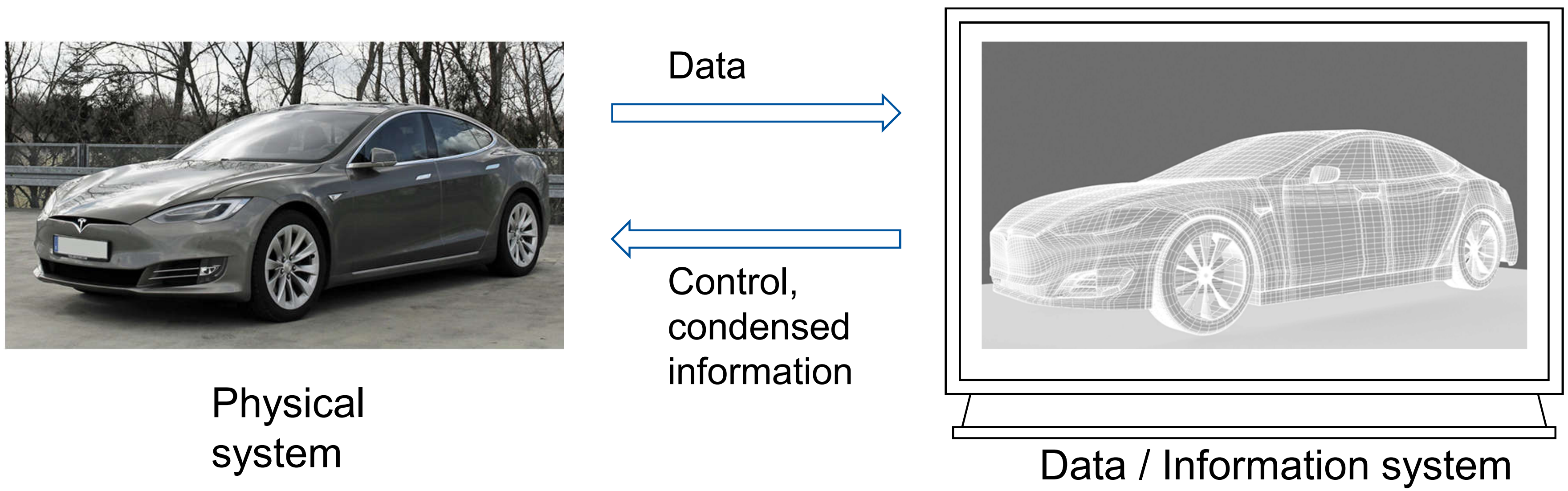
History of Digital Twins

- Goal: Increase system availability and performance of systems by
  - Analyzing physical processes and judging, predicting and optimizing virtually
  - Providing data from physical system to complete simulations, validate settings and dynamically adjust
  - Analyzing results and feeding back to respond to the changes
- Term “twin” originates from NASA: Build a physical copy of aircrafts to simulate and test control scenarios
- Today: Digital Twins normally are virtual representations of physical things:
  - digital models about the physical thing
  - data about/of the physical twin
- Realizing new technologies requires close collaboration of experts and connecting various models

Our Digital Twin Definition

A Digital Twin of a system consists of

- a set of models of the system,
- a set of digital shadows, and
- provides a set of services to use the data and models purposefully with respects to the original system.



What a Digital Twin is (not)

A Digital Twin of a system consists of
- a set of models of the system,
- a set of digital shadows, and
- provides a set of services to use the data and models purposefully with respects to the original system.

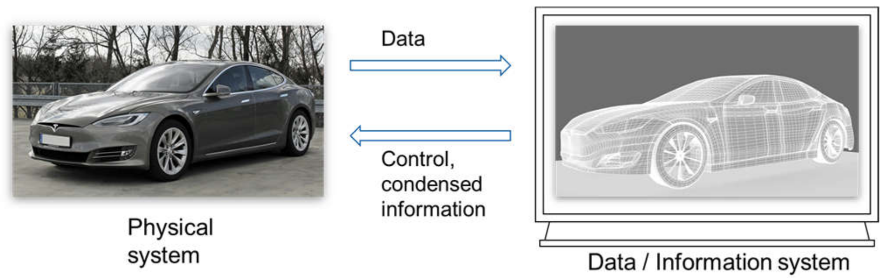
We found 118 definitions in literature

- A digital twin **is not** “a model“ (a digital twin can be but doesn’t need to be)
- A digital twin **is not** a database in the cloud (but it might comprise it)
- A digital twin **does not** include a CPPS (which has physical parts)
- A digital twin **is not a complete** digital representation of another system (digitizing implies abstraction)
- A digital twin **is not** “a digital avatar” (what does that even mean?)
- A digital twin **is not** “a digital equivalent of a product on the shop floor” (but it can be)

Digital Shadows as Part of the Digital Twin

A Digital Twin of a system consists of
- a set of models of the system,
- a set of digital shadows, and
- provides a set of services to use the data and models purposefully with respects to the original system.

- Physical world contains observable elements that can be monitored, sensed, and may be actuated and controlled
- Data Collection & Device Control interacts with the physical world to observe and influence its behavior
- Creates digital shadows based on data about the physical world and queries/specifications from the digital twin applications



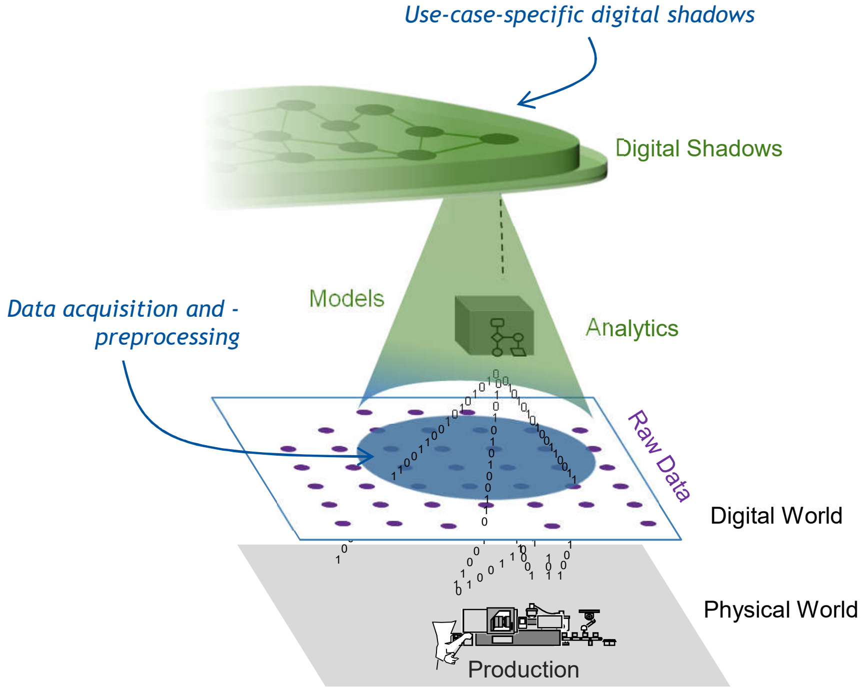
embodies software *pure software*

the complete system -
when viewed holistically

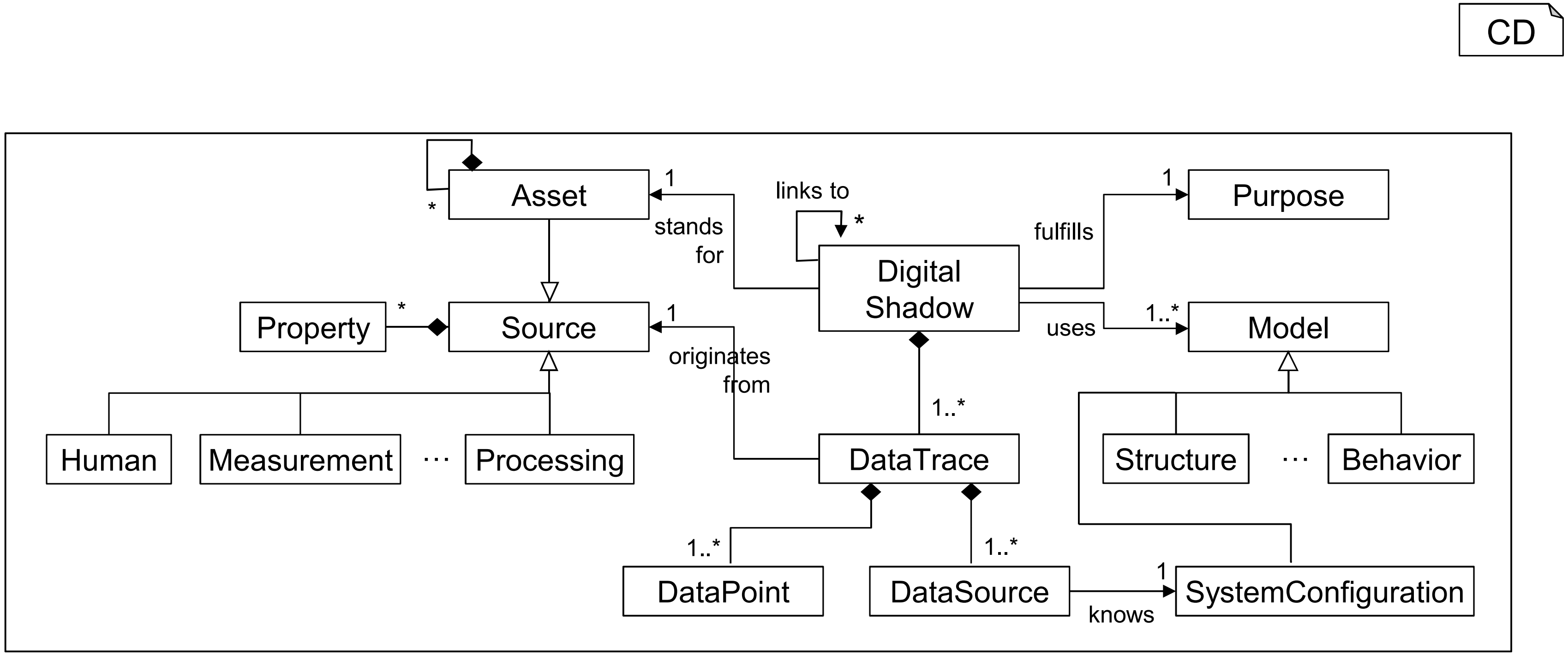
Digital Shadows

- A Digital Shadow is a set of contextual data traces and their aggregation and abstraction collected for a specific purpose with respect to an original system.

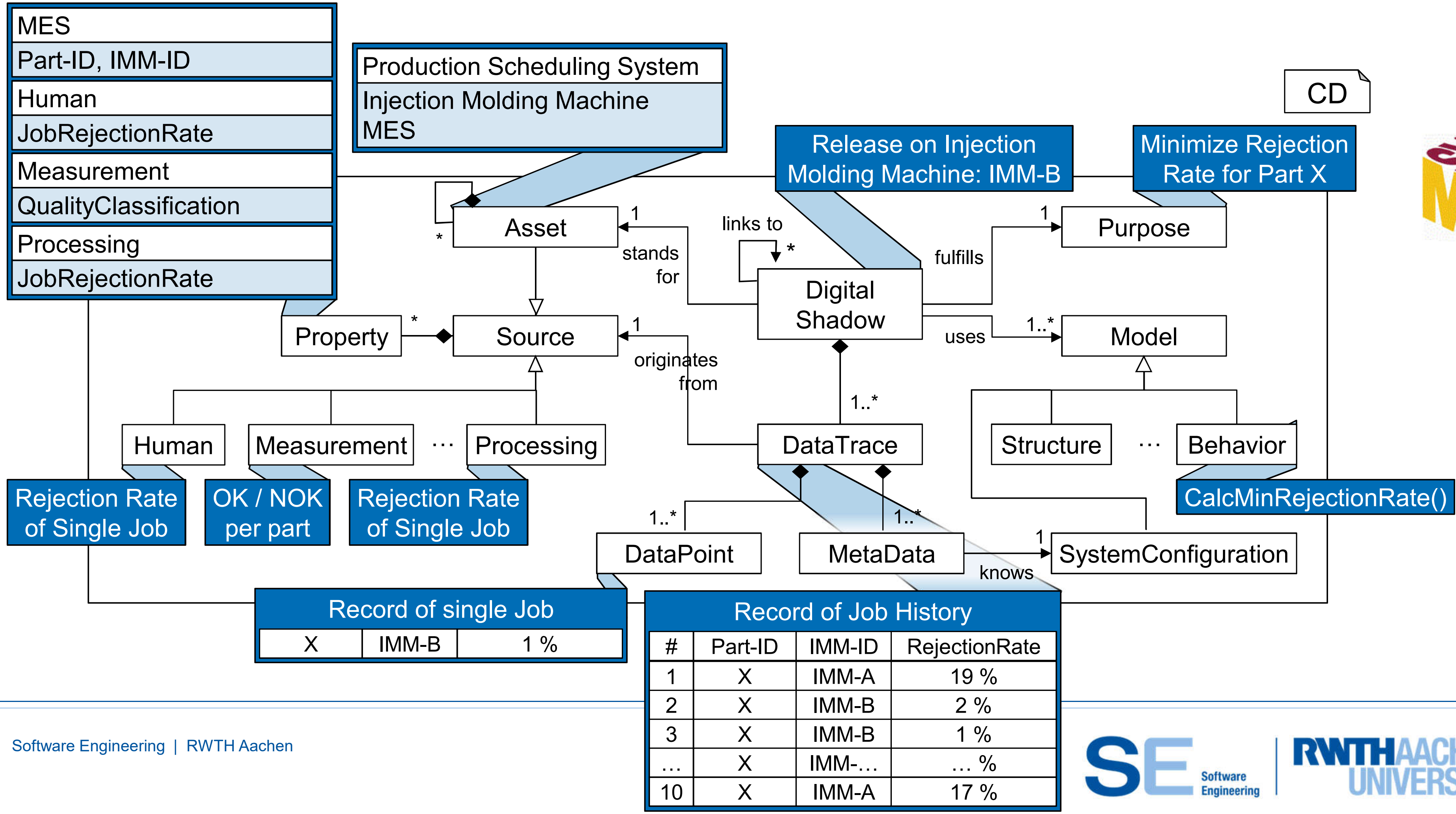
- A digital shadow is
  - a passive set of data
  - information source about a system's state and history
  - is collected, filtered and reduced for its dedicated purpose in varying forms of abstractions
  - a purely digital artifact
  - produced by a (physical) system.
- A system can have many different digital shadows describing a variety of different aspects of the system in different detail and at different times.
- Shadow may contain information about production systems, production processes, products, and human operations



Typical Digital Shadow Reference Model [BBD+21]



Conceptual Model for Digital Shadows & Example in the IOP [BBD+21]



Record of single Job

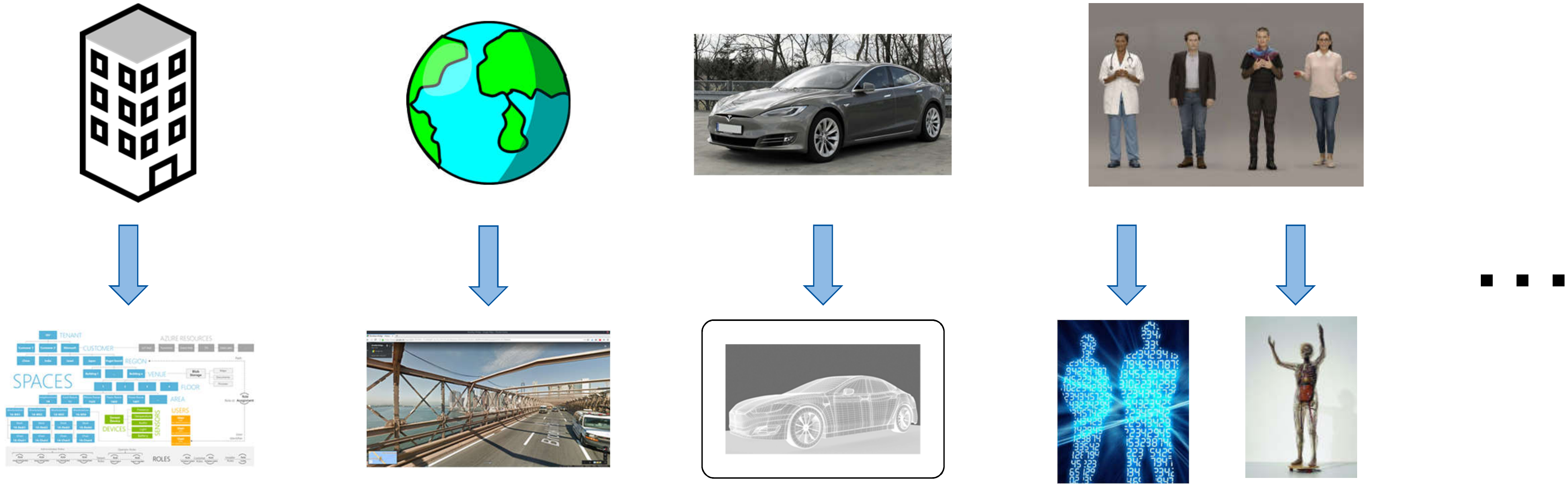
X	IMM-B	1 %

Record of Job History

#	Part-ID	IMM-ID	RejectionRate
1	X	IMM-A	19 %
2	X	IMM-B	2 %
3	X	IMM-B	1 %
…	X	IMM-…	… %
10	X	IMM-A	17 %

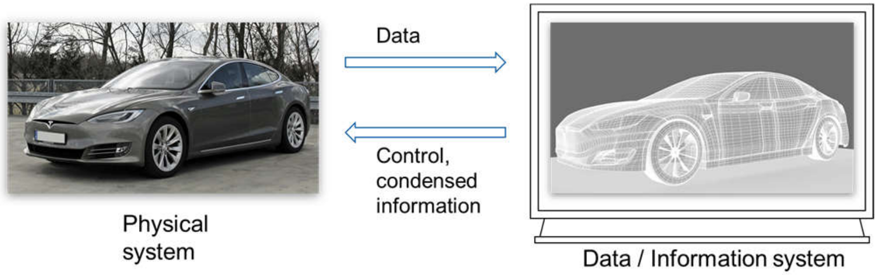
Kinds of Digital Shadows

- Different physical entities, very different, purpose specific kinds of data models
  - e.g., BIM, Google Earth, CAD, Conceptual Models



Purposeful Services are Part of a Digital Twin

A Digital Twin of a system consists of
- a set of models of the system,
- a set of digital shadows, and
- provides a set of services to use the data and models purposefully with respects to the original system.



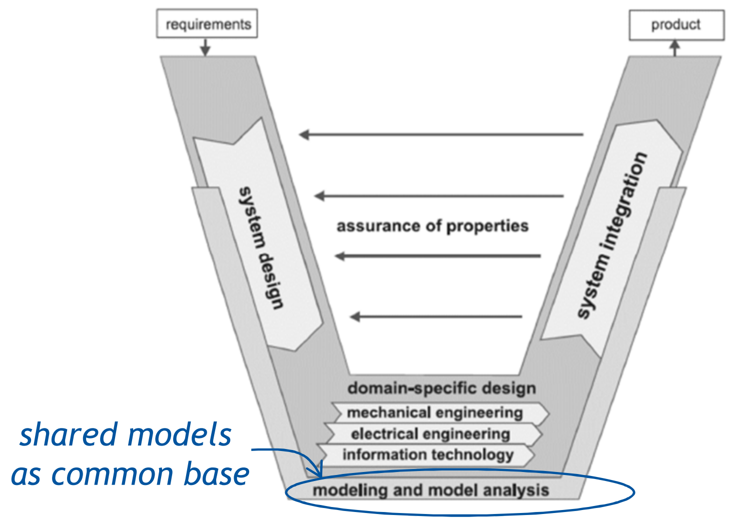
- Digital Twin Applications & Services take actions based on one or more situations that they sense in the environment
- User can access Digital Twin Applications & Services through appropriate interfaces
- Digital Twin has (some) strategic control over his physical twin
  - usually high-level, the real-time control strategies are embedded in the CPS

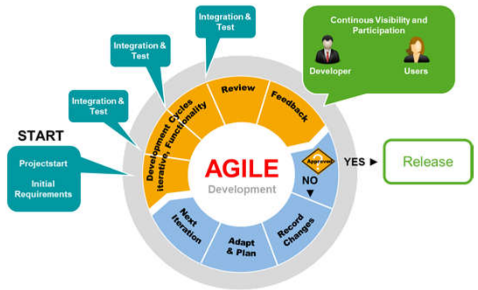
Development Process of CPS and Digital Twins

- Services of a Digital Twin are intensively connected to the CPS
  - Integrated parallel development
  - NOT only a spin-off product of the development of the physical system
  - may be configurable, can be parameterized or calibrated
  - new services may be coming over time

- Adaptivity through explicit models at runtime:
  1. Autonomous self-adaptation, e.g. induced by changes of the context, optimizations identified for example through continuous measurements or by a slow degradation of the system itself
  2. The user wishes to adapt the system behavior
  3. The manufacturer adapts the system behavior according to identified optimizations, fixing of bugs and failures, or upgrade of functionality

- Challenge:
  Development cycles/methods for CPS and IT differ radically





Systems Realize Functions Related to Their Purpose

- From the definition: A system has one or more purposes with respects to its operational context.
- Consequence: **Systems realize functions.**
- Functions of systems
  - Train → Mobility
  - Car → Individual mobility, also: storage
  - DaVinci medical robot → Enable remote operation
  - Freighter → Transport
  - Smart phone → Communication, photography, …
  - Manufacturing system → Produce goods
  - …





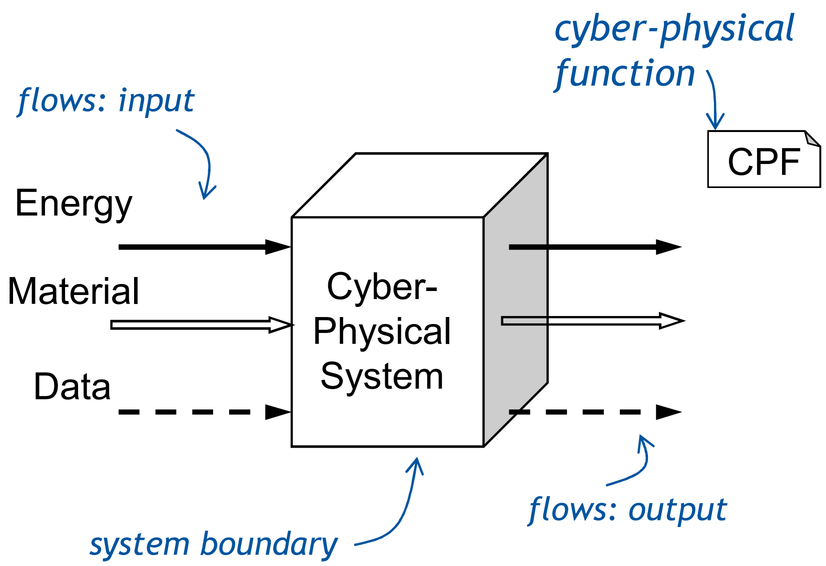
Systems thinking is based on functional specification, design and implementation. Geometry (i.e., physical shape): form follows function.

System Specification through Functions

- A system defines a cyber-physical 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

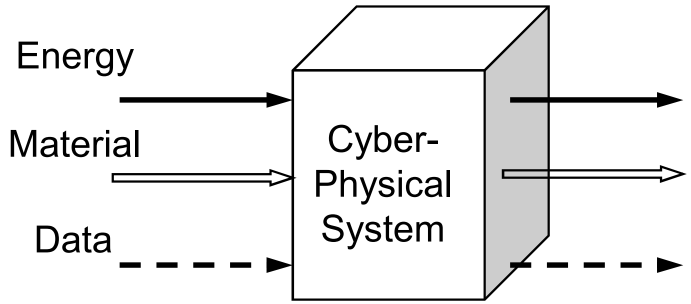
- The functionality is mathematically described through the
  – relation between input and output



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

The Underspecification Principle

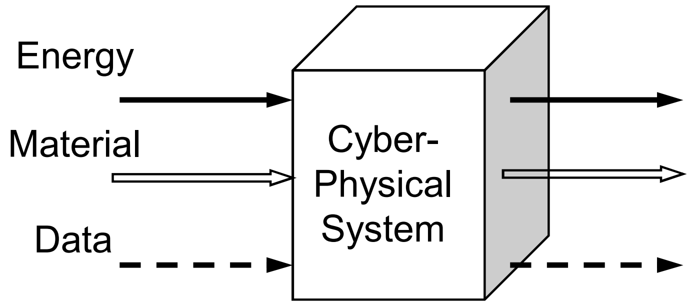
- Deterministic and fully specified relations are normally not achievable
  - Delays happen
  - Energy fluctuates
  - Abstraction introduces lack of information
- Underspecification is the ability to describe the desired range of allowed behaviors (instead of a single, determined behavior)
- Advantages: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



⇨ Controlled, explicit underspecification is the second universal specification principle

Signature of Input and Output of a Function

- The signature of a function describes the forms of interactions of a system component with its environment.



- Interactions are broken down to streams of elements, which describe the flow and can be of the kinds
  - data,
  - energy or
  - material

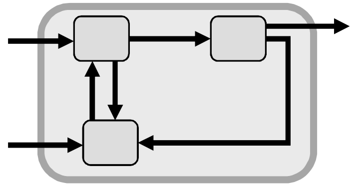
- The Interface of a Cyber-Physical System is defined through its function signature

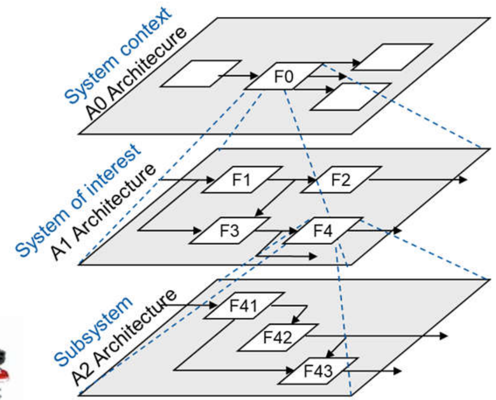
- Interactions are organized through input and output channels.

⇨ The concept of stream is our third universal specification and construction 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 ~ non-atomic component

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

Cyber-Physical Systems and Digital Twins

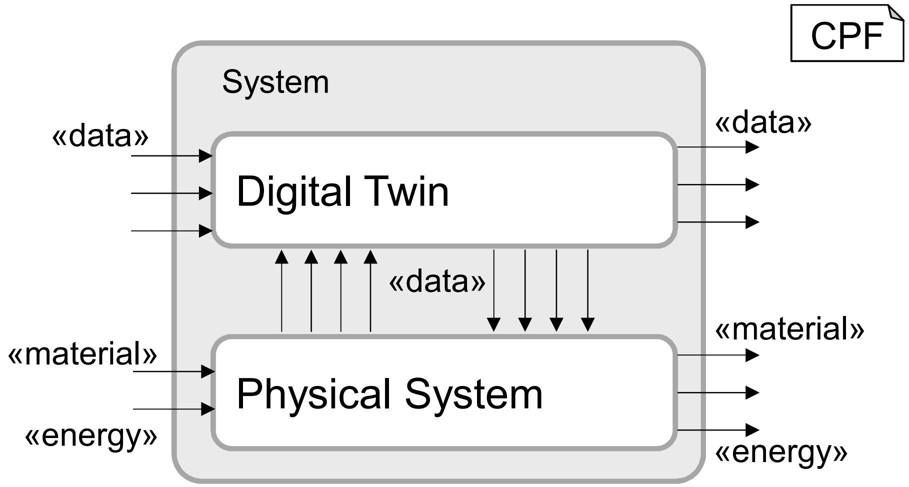
Cyber-physical systems (CPS) are engineered systems where functionalities are emerging from the networked interaction of physical and computational processes. [BDS19]

- Literature currently is unclear, whether a digital twin is part of the CPS or beneath to it

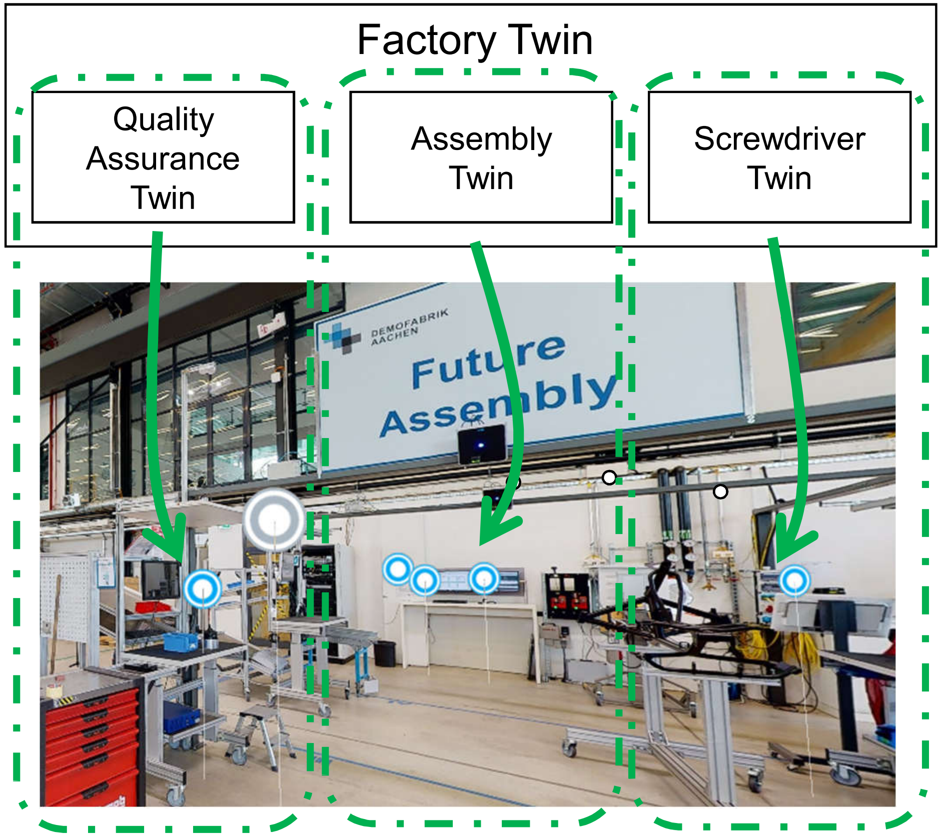
- A useful viewpoint:
- The overall system contains the physical part, sensors, actuators the embedded controls, data collection, services and user interface

- The overall system is engineered in an integrated project considering physical and IT part in parallel

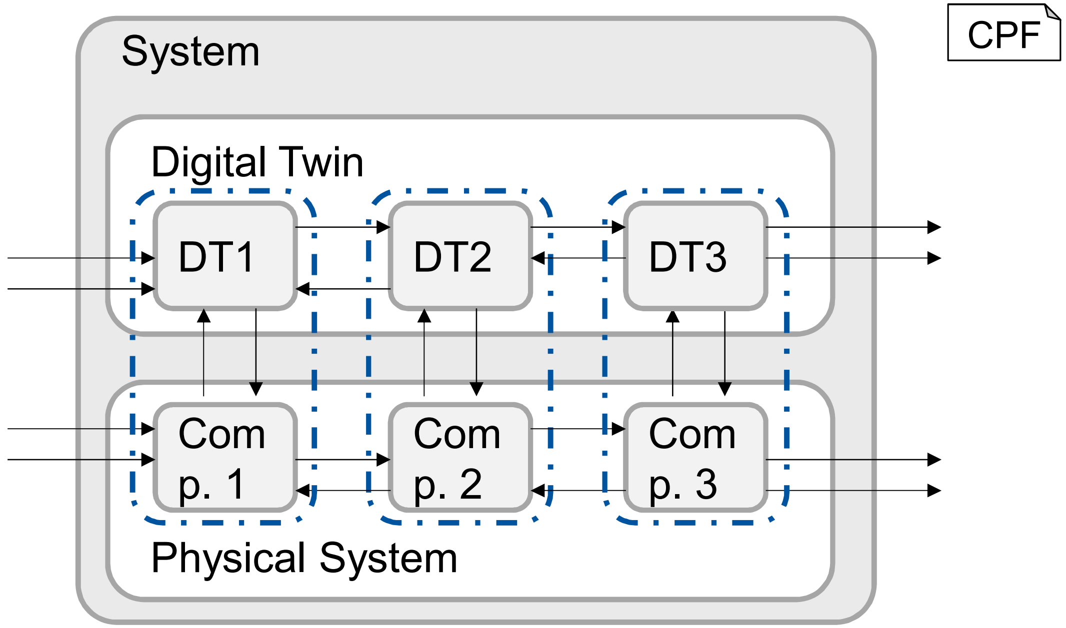
- Digital Twin builds a logical entity, but its software components may be distributed



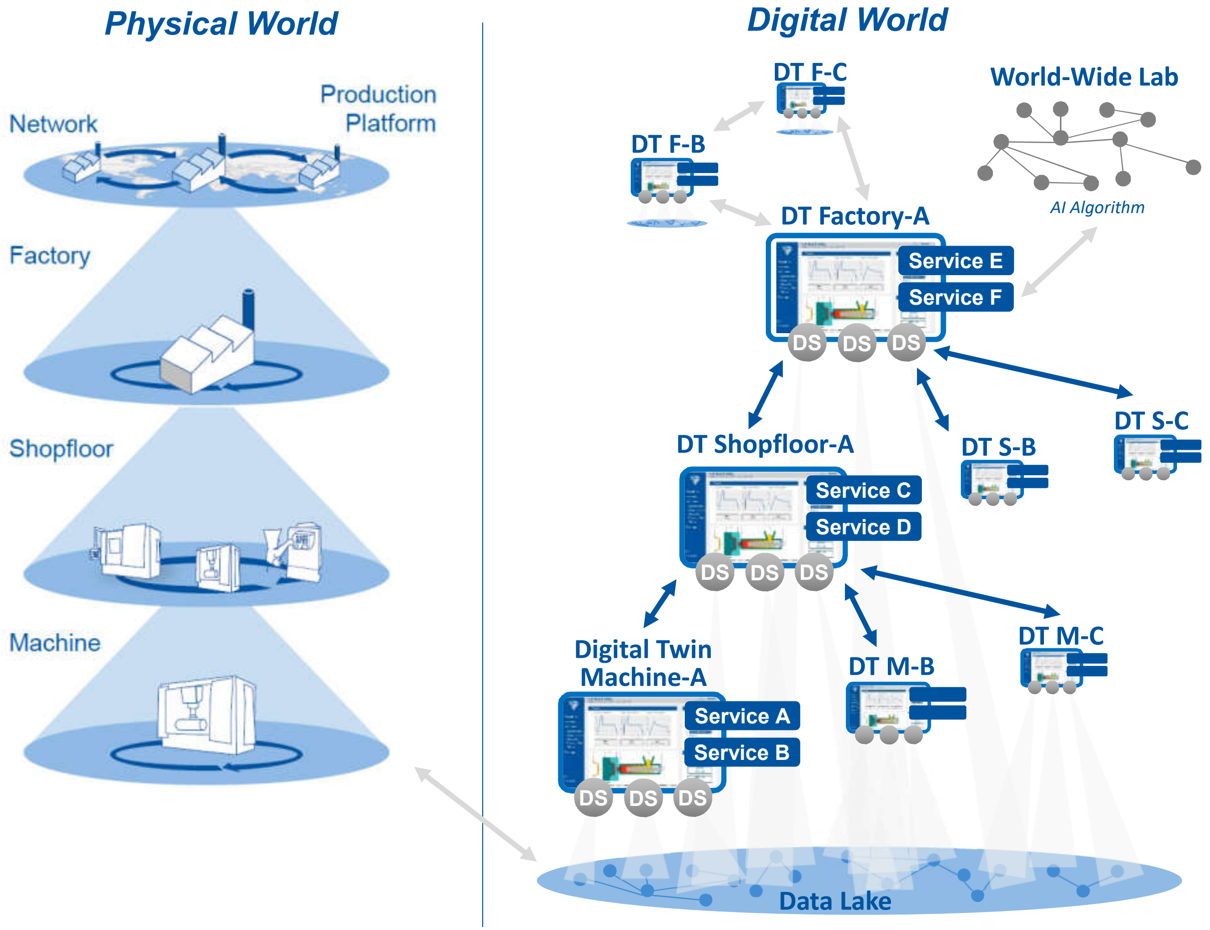
Composition of Digital Twins



- Factory decomposed along two dimensions
  - DT's are composed
  - Physical components are composed to a system
  - But ideally: Physical component + its twin is developed / purchased / shipped together



The SE-Vision within the IoP



Our aim:

Efficient development of digital twin services based on digital shadows

Provide Engineering Tools & Methods

Needed Tools are e.g.:

Digital Shadow Type Creator
- generate DS-Types which can be used during runtime to create DSs
- select data sources from the data lake

Low Code DT Configurator
- configurable DTs

Application specific to the DT Platform
- services for data extraction from engineering models
- API's to other services, e.g. AI algorithms
- integrated process mining services

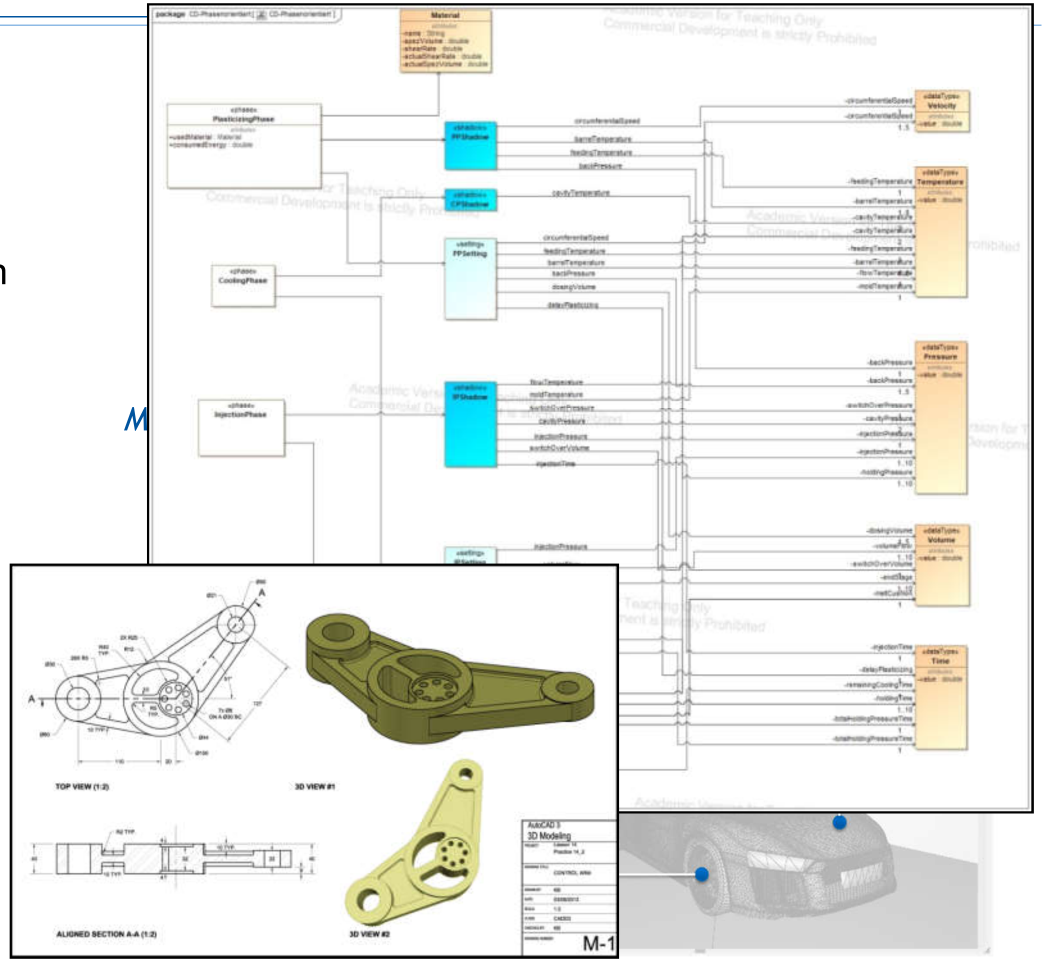
Model-Driven Digital Shadow Creation

- Cyber-physical systems are complex
  - Consist of multiple components
  - Offer different functionalities

- Reuse engineering models that are created during system design for systematic efficient definition of larger parts of a Digital Twin

- Generate a Digital Shadow Caster that accesses the CPS and displays potentially interesting Digital Shadows from Engineering Models

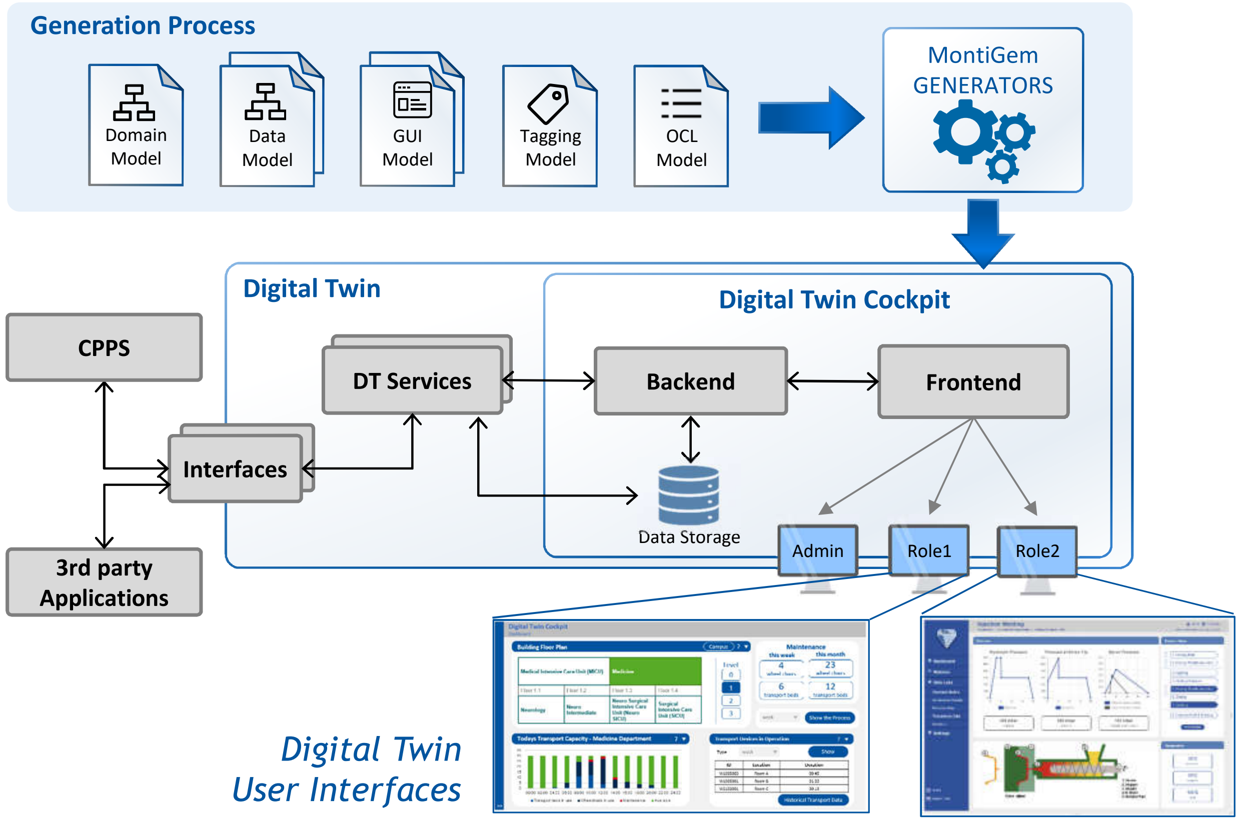
- Extract structural information about the CPS
  - How is the CPS composed
- Spatial Information
  - Where are the CPS and its internal components located
- Expected behavior
  - How should the system react to a specific situation
  - Derive, when the system does not behave as intended



Creating Digital Twin Cockpits with MontiGem [DMR+20]

Generating digital twin cockpits from models with the generator framework MontiGem

- Successfully applied to research and real-life projects
  - MaCoCo, Engineering Wind Turbines, InviDas *see: https://www.se-rwth.de/projects*
  - Digital twin cockpit: Injection Molding [DMR+20]
- Components of the digital twin cockpit application
  - Database, backend and frontend of a web application, communication infrastructure
- Used models, e.g.,
  - Domain model: data structure
  - GUI models: user interfaces
  - Data models: representation of parts of data in GUIs
  - OCL models: validation of data input

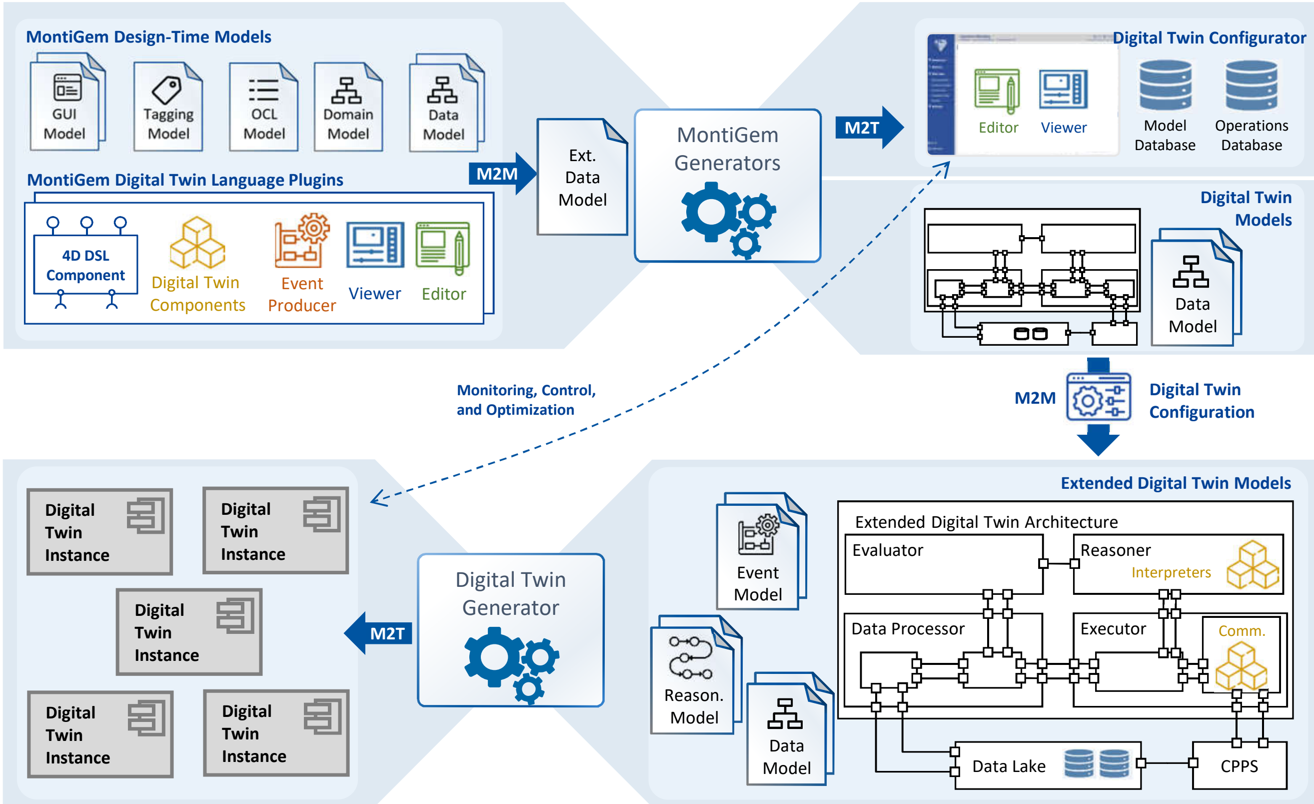


[DMR+20] M. Dalibor, J. Michael, B. Rumpe, S. Varga, A. Wortmann: Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits. ER'20

Low-Code Platforms for Model-Driven Digital Twins [MW21]

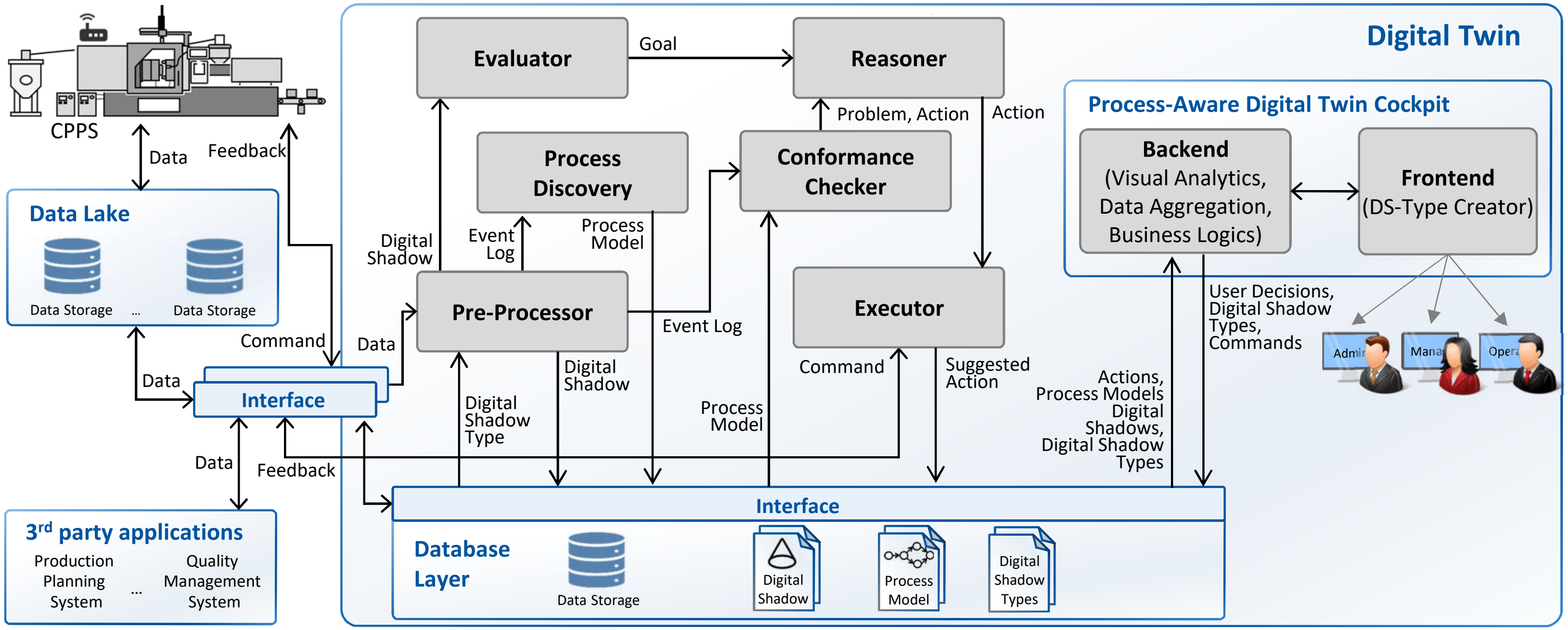
- Digital twins configured and operated by shop-floor experts: rarely professional software engineers

- 2-step development process
  - 1) generate the low-code platform that contains a digital twin configurator:
  - Shop-floor experts configure their digital twin
  - 2) generate derive the digital twin from the configuration:
  - Used and operated together with the system



Process Prediction with Digital Twins [BHK21]

- Aim: improve the operation of digital twins
  - process discovery from event logs and
  - process prediction from process models at runtime
- Models at designtime
  - Application-specific models, e.g. domain model, GUI models, …
  - Application independent models, e.g., architecture, basic DS and process structure,…
- Models at runtime
  - process models, goals, actions,…



Presentation in Oct. 21 at MODELS@run.time Workshop

with Wil van der Aalst, István Koren, Merih Seran Uysal, Tobias Brockhoff (PADS RWTH Aachen University) and Andreas Wortmann (University of Stuttgart)

Models for the Engineering of Digital Twins

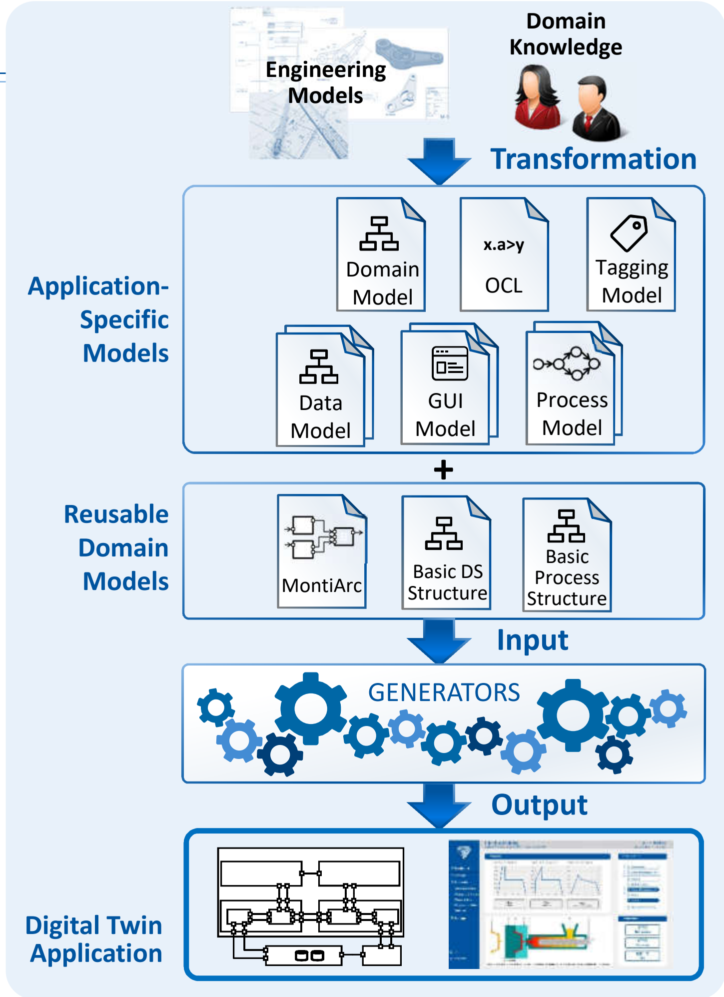
Derive Information from engineering models to create system models

Models at design time

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

Models at run-time

- e.g., processes, goals, actions, events, …



[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

[BHK+21] 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@run.time WS (MODELS'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

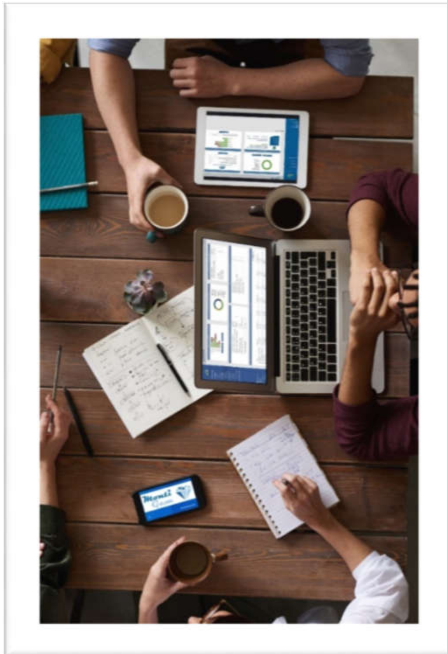
[GMN+20] A. Gerasimov, J. Michael, L. Netz, B. Rumpe, S. Varga: *Continuous Transition from Model-Driven Prototype to Full-Size Real-World Enterprise Information Systems*. AMCIS'20

Summary

- Efficient development of digital twin services in a usable cockpit
  - based on digital shadows
  - based on engineering models
- Model-driven engineering of digital twins…
  - Facilitates the creation and configuration
  - Empowers non-software engineers
  - Helps us to handle data and models cross-domain and on a sufficient aggregation level
- There‘s a million things still to tackle

Bernhard Rumpe *Judith Michael* *Malte Heithoff* *Nico Jansen* *Steffen Hillemacher*

… and formerly also Andreas Wortmann



Current
Projects
se-rwth.de/projects

12.10.2021, Aachen

Selected References

Modeling in Industry 4.0

- [WBCW20] A. Wortmann, O. Barais, B. Combemale, M. Wimmer: *Modeling Languages in Industry 4.0: an Extended Systematic Mapping Study*. Software and Systems Modeling, 19(1), 2020

Digital twin architecture

- [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

Low code platforms for digital twins

- [MW21] J. Michael, A. Wortmann: *Towards Development Platforms for Digital Twins: A Model-Driven Low-Code Approach*. APMS'21

MontiGem

- [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

Find more at: https://www.se-rwth.de/publications

Backup

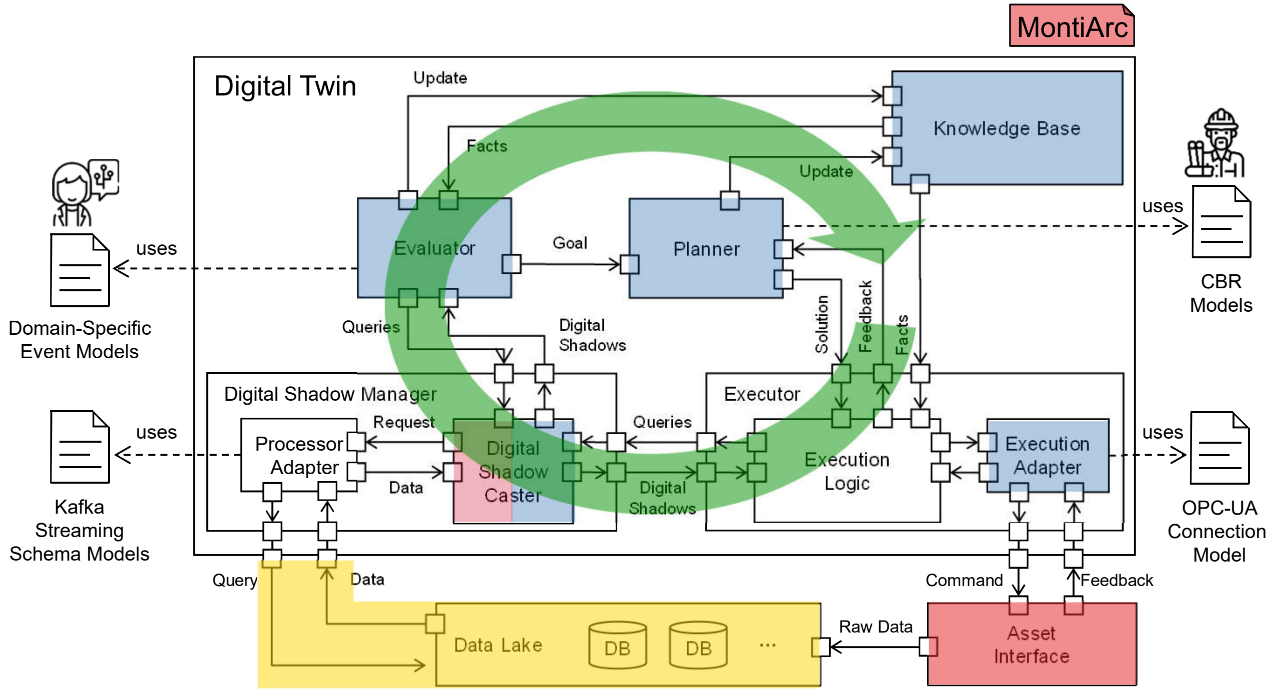
Current
Projects
se-rwth.de/projects

A Possible Model-Driven Digital Twin Architecture [BBD+21a, DMR+20, BDH+20]

That realizes a MAPE-K self-adaptive loop over the CPPS

Digital twin: models + contextual data + services used purposefully w.r.t the original system.

- **MontiArc reference architecture**
  - Uses **data** lake and asset
  - Creates digital shadows
  - Evaluates state and acts
- Domain-specific **models**
  - trigger shadow creation
  - connect to data lake
  - case-based reasoning
  - connect to CPS
- **Services:** Representation, monitoring, optimization

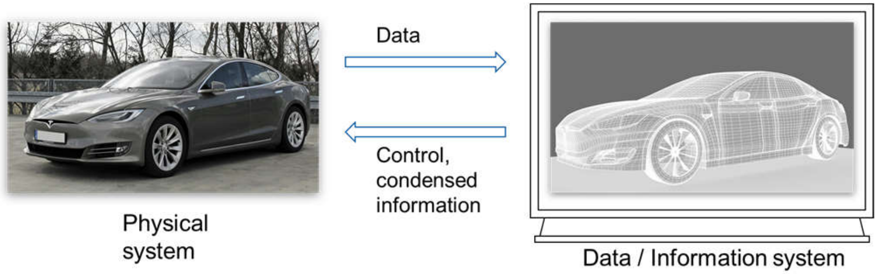


Digital Twin vs. Digital Shadow vs. Model

A Digital Twin of a system consists of
- a set of models of the system,
- a set of digital shadows, and
- provides a set of services to use the data and models purposefully with respects to the original system.

A Digital Shadow is a set of contextual data traces and their aggregation and abstraction collected concerning a system for a specific purpose with respect to the original system.

A model is essentially a reduced or abstracted representation of the original system in terms of measure, precision and functionality. (Stachowiak 1973)

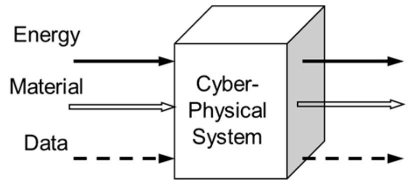


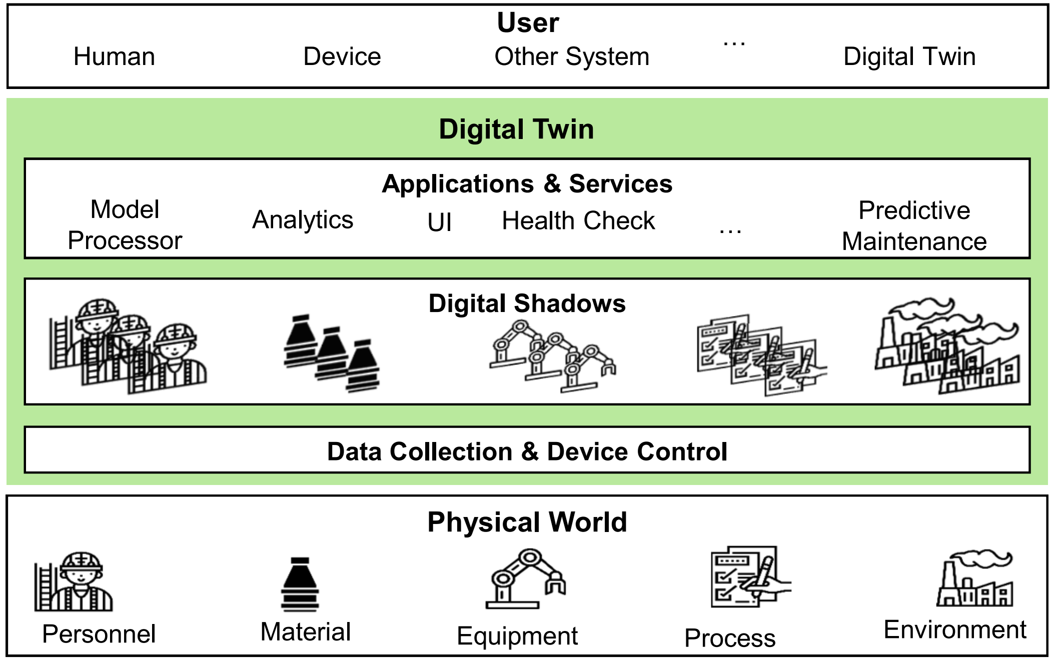
- Three different terms …
- They share some characteristics, but:
- Twin is an active software system (through services)
- Model prescribes the system under development and at operation
- Shadow is passive data produced at runtime/operation

Cyber-Physical Systems and Digital Twins

Cyber-physical systems (CPS) are engineered systems where functionalities are emerging from the networked interaction of physical and computational processes. [BDS19]

- Literature currently is unclear, whether a digital twin is part of the CPS or beneath to it
- A useful viewpoint:
- The overall system contains the physical part, sensors, actuators the embedded controls, data collection, services and user interface
- The overall system is engineered in an integrated project considering physical and IT part in parallel
- Digital Twin builds a logical entity, but its software components may be distributed





Cyber-Physical Systems and Digital Twins

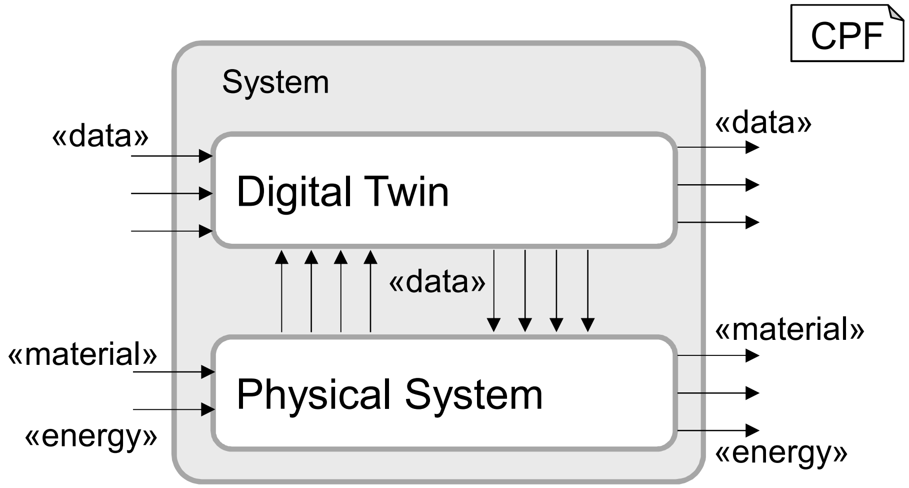
Cyber-physical systems (CPS) are engineered systems where functionalities are emerging from the networked interaction of physical and computational processes. [BDS19]

- Literature currently is unclear, whether a digital twin is part of the CPS or beneath to it

- A useful viewpoint:
- The overall system contains the physical part, sensors, actuators the embedded controls, data collection, services and user interface

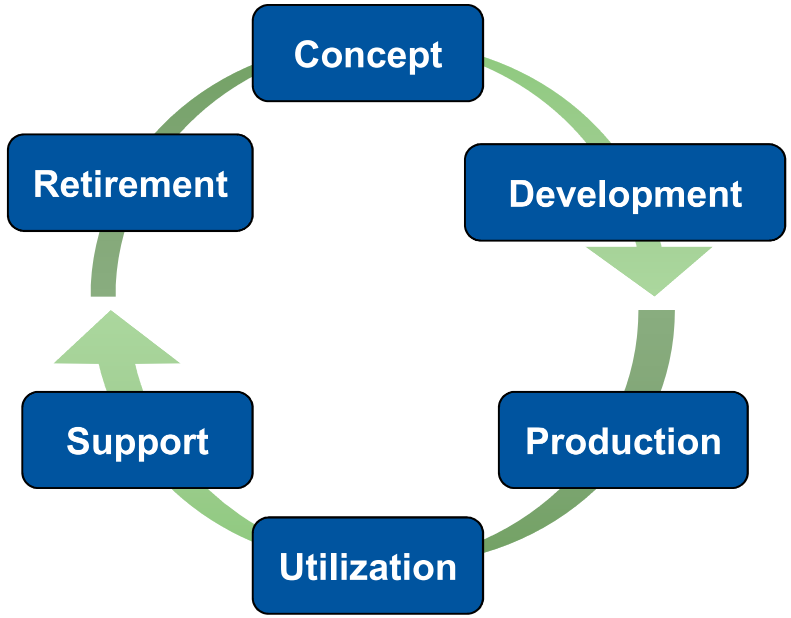
- The overall system is engineered in an integrated project considering physical and IT part in parallel

- Digital Twin builds a logical entity, but its software components may be distributed



Digital Twins Support all Lifecycle Phases of a System

- Depending on the lifecycle phase a Digital Twin offers different services to control/adapt/represent the physical system
- During Concept and Development, a Digital Twin simulate the behavior of a CPS
  - Support communication engineers and designers while working together across departmental boundaries
  - Evaluate product variants to support design decisions
- During Production, a Digital Twin
  - Supervise the production process, e.g., individual deviations from norm that require special treatment
  - Trace the applied materials, components, processing steps
- During Utilization and Support a Digital Twin
  - Provides information on system state, history and usage
  - Enables optimization of a machine during operation
  - Facilitates the improvement of future products
  - Enables predictive maintenance

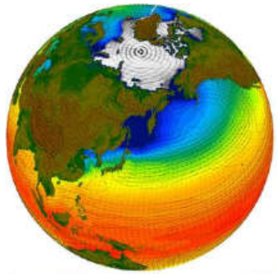


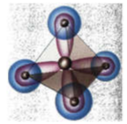
System Lifecycle from ISO/IEC 15288 (Systems Engineering standard)

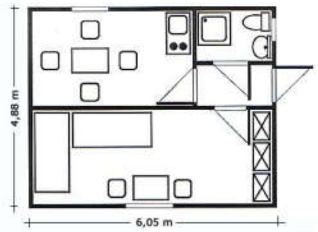
Models are Most Useful if Grounded on a Theory of Modeling

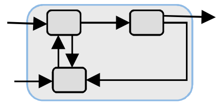
- In a software & systems development process:
  - Models are composed
  - Models evolve
  - Models are compared
  - Models are analyzed
  - Models are used to derive/generate/synthesize systems

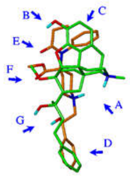
- This works best if there is
  - A) a precise understanding of what a well-defined model is (syntax)
  - B) a precise definition of what a model means (semantics)
  - C) an elaborated underlying mathematical theory
    consisting of theorems and laws that give us
    tools at hand to analyze, transform, test, etc. the models
  - Syntax and semantics are covered by modeling language definitions
  - Theory is an underlying supplement for the semantics definition

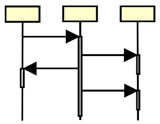


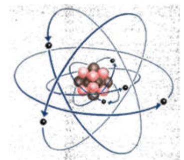








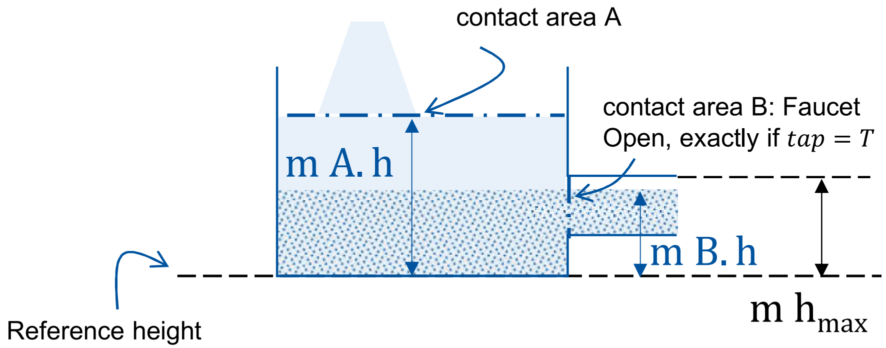




Example: Physical Function that Stores a Fluid in a Reservoir

- Interface:
  - Two contact areas A, B that transport a flowing water described by the filling level (m) measured from the reference height and the volume flow rate (m^3s^{-1})

- Specification of the behavior:
  - We assume ideal flow (no friction) :
    - $B.h = \begin{cases} h_{max}, & tap = T \wedge A.h > h_{max} \\ A.h, & tap = T \wedge A.h \leq h_{max} \end{cases}$
    - $B.q = \begin{cases} \sqrt{2g \cdot h}, & tap = T \\ \sqrt{2g \cdot B.h}, & tap = F \end{cases}$ (Bernoulli‘s Principle)



Example: Physical Function defined by a Friction Gear

- A friction gear is a specific wheel and axle, i.e. a rigid rotating shaft that transmits rotational energy at the wheel through friction locking
  - The friction gear has two contacts: One on the axle and one on the wheel

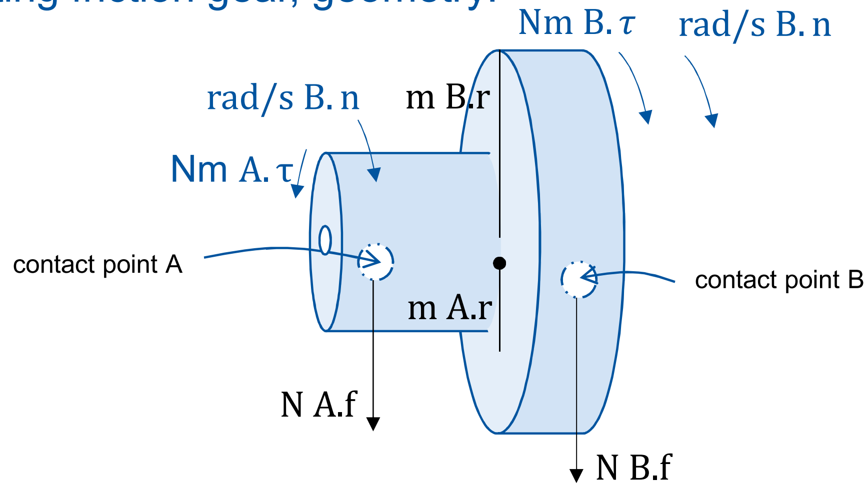
- Interface:
  - Two contact points A, B that transport the torque and the wheel speed
  - We define each interface as
    - In $\mathrm{A: (Nm\ \tau, rad/s\ n)}$
    - Out $\mathrm{B: (Nm\ \tau, rad/s\ n)}$

- The torque (described in Nm = J) transmission is (mechanical equilibrium): $\mathrm{A.\tau = -B.\tau}$

- Rotational speed is equal (rigid body) : $\mathrm{A.n = B.n}$

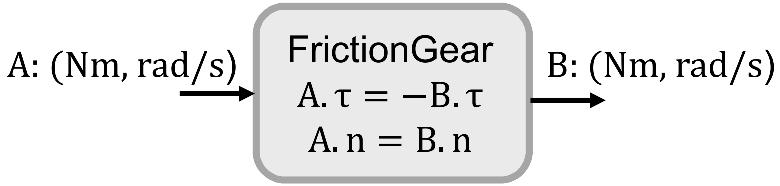
- Transported energy (described in W): $\mathrm{p = 2\pi \cdot \tau \cdot n}$

rotating friction gear, geometry:



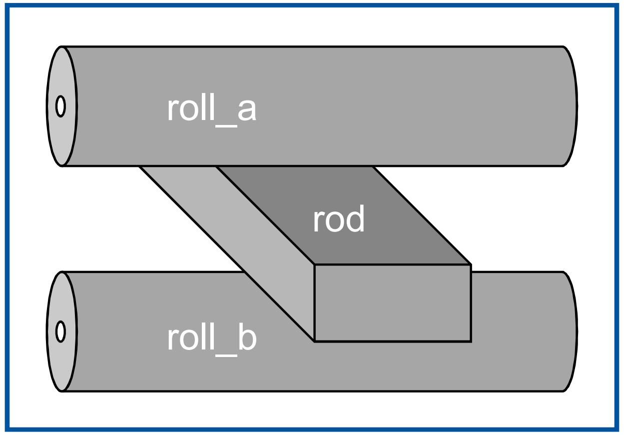
- $\tau 1, \tau 2$ are the torque acting on / transmitted by the gear,
- $n1, n2$ are the wheel speeds

friction gear, function:

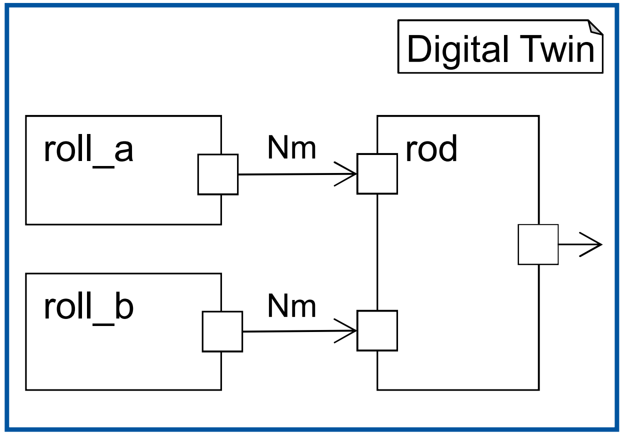


Deriving Digital Twins from 3D Interactions

- Collect functional interrelationships and degrees of freedom for deriving a Digital Twin model
  - (Semi-) automatable generation process
- The twin represents possible mechanical interactions between the particular assemblies
- Extensible Digital Twin via enrichment with additional information
  - Integration of further engineering models
  - Aggregation of external data

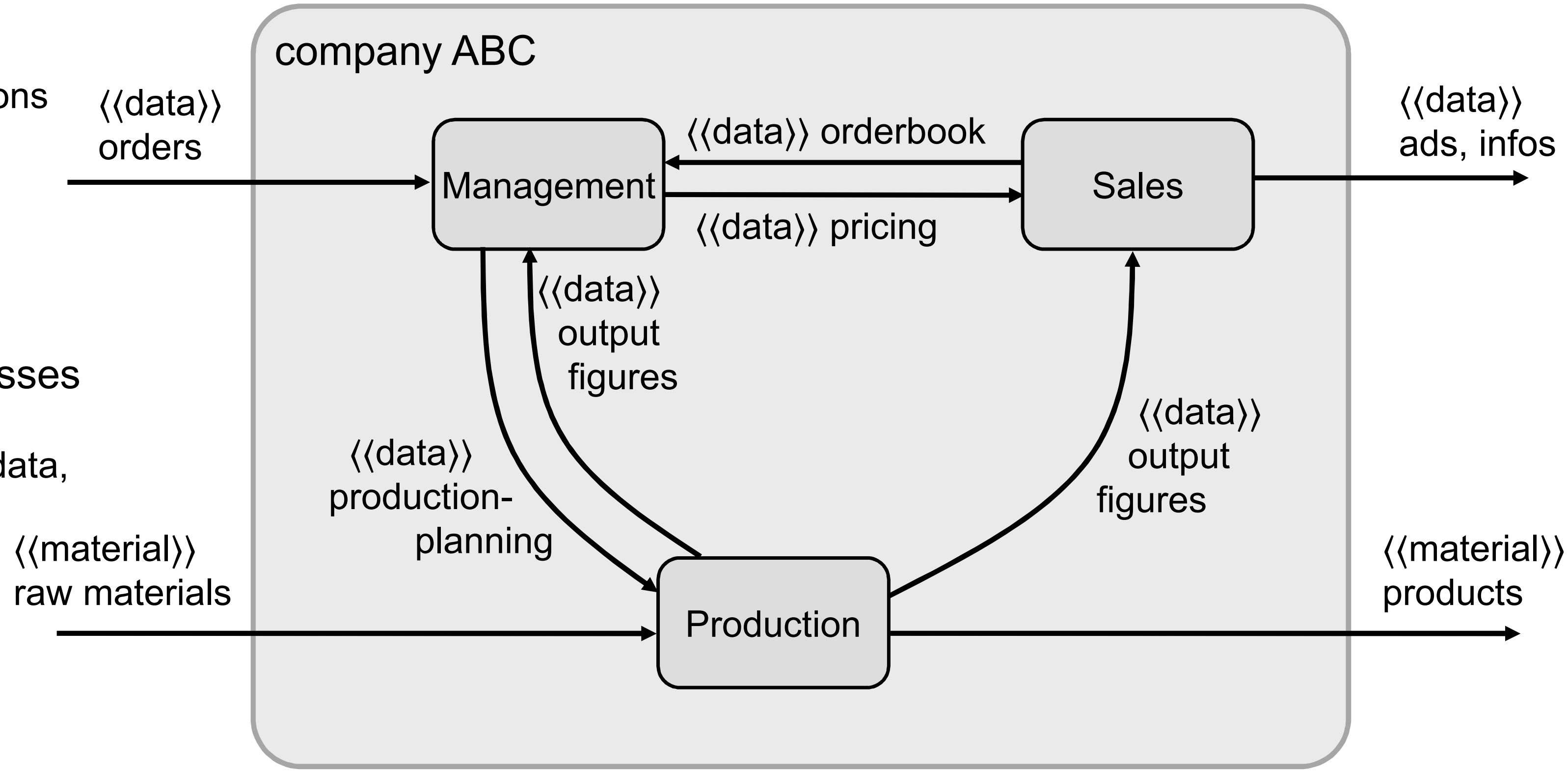



```
T[roll_a, rod, (0, 1, 0)]
T[rod, roll_b, (0,-1, 0)]
R[roll_a, rod, (1, 0, 0)]
R[rod, roll_b, (1, 0, 0)]
...
```



Example: Dataflow and Material Flow in a Company

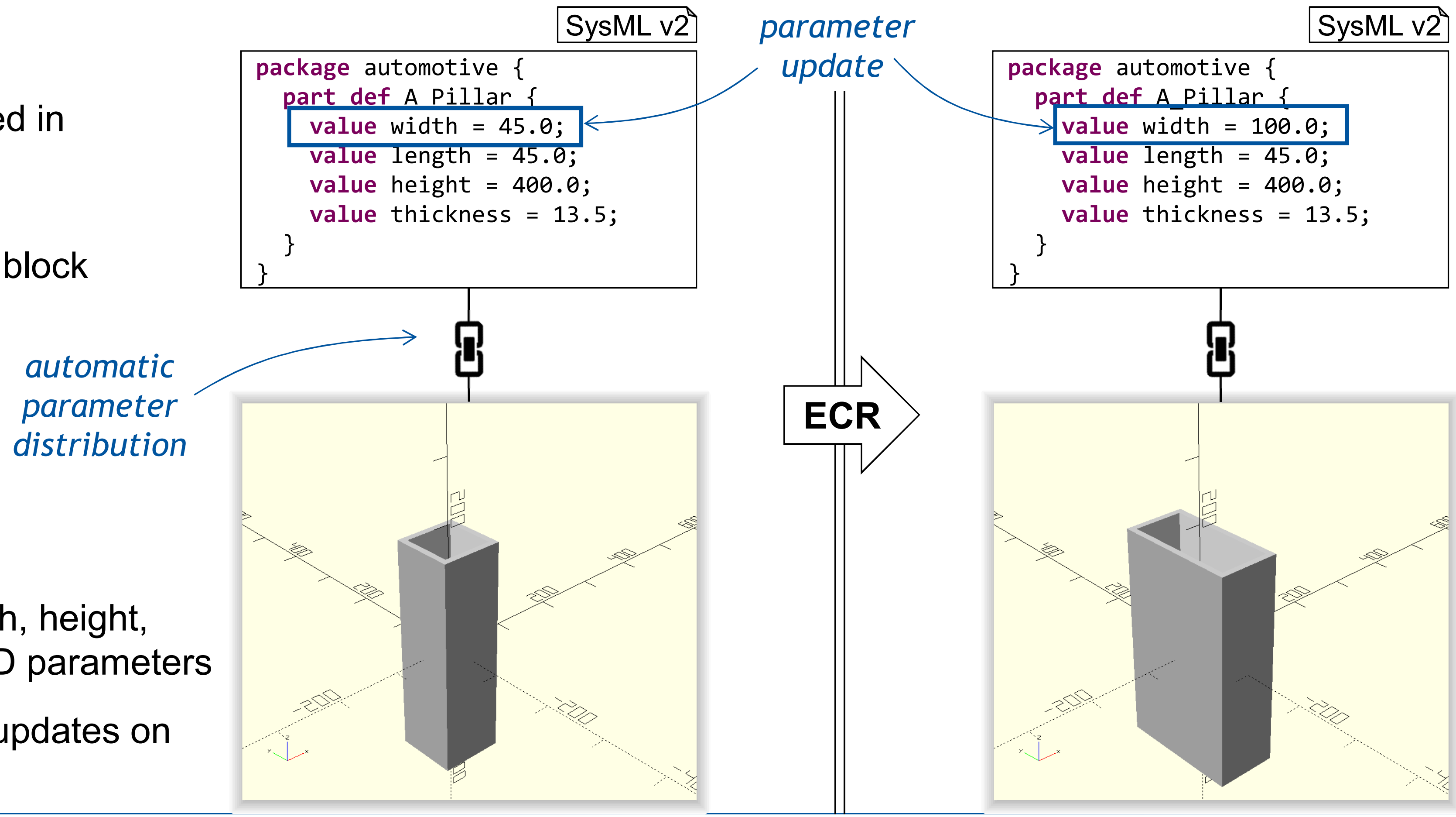
- Companies,
  business processes,
  production processes:
  - Can be specified as functions
- Production takes time:
  - Material is processed,
  - Material is stored, etc.
- Company business processes
  - Use history for prediction
  - Use data to produce new data, work directives, etc.



Case Study – Bridging SysML and CAD

Use Case:

- A-pillar of a car modeled in
  - SysML block diagram
  - OpenScad model
- Transfer symbols from block diagram to CAD model



Results:

- Integrating width, length, height, and thickness with CAD parameters
- Automatic distribution updates on manual changes

Detailed Explanation to the Digital Twin Definition

For a detailed reading

A DT is not(!) directly a model.

- However, DT and models share a lot of characteristics. A DT has a purpose with respects to the original (i.e., the system) and it contains a lot of knowledge about the system.
- A DT therefore contains one or a set of models of the physical system.

A DT is not identical to the physical system. While twins in the real world have the same origin and share therefore a lot of characteristics, twins are also not identical in the real world.

- DT is digital: digitalization in itself means that something physical is represented by a non-physical, pure data-based entity. Product and DT can therefore not be confused with each other.
- Digitalization also means that the DT is an abstraction. A digital version of an analogous product or picture is always an abstraction, no matter how detailed it is.

A DT provides services. Services are active. They are part of a digital system consisting of state, e.g., a data base, algorithms, and potentially a graphical user interface.

- The services of the DT may thus be used by other digital systems or by humans.

The Digital Twin can change the behavior of the physical system

The DT contains a set models. These models are typically either (1) derived from engineering models, (2) directly the engineering models, or (3) derived as abstractions from the data traces collected over time of operation.

The DT has a purpose with respects to the original system. This characteristic is shared with the definition of models. Quite a variety of purposes of the DT can be thought of.

- The DT can for example be used to analyze, optimize, simulate, predict behavior throughout the complete lifecycle of a physical system.
- The DT can be used to understand the behavior of the system, as well as the context it is operating in, in which state it is during development, in which mode it is during operation, etc.

The DT contains sets of digital shadows. A digital shadow is itself a passive data trace or an abstraction of it. A DT may collect more data traces over time.

- While the DS is passive, the DT contains all relevant functionalities (algorithms), storage and retrieval to actually perform the abstractions and aggregations.
- The DT provides services that use a set of functionalities to visualize and thus allow to analyze the collected DS.

To execute its services the DT connects the data traces with the engineering models, such that the databases become meaningful.

- e.g., the models describe actual and virtual sensors of the system and the data traces describe the values collected by these sensors.
- Sensors in a broad sense can be automatic (e.g. temperature of the system), or input by humans (descriptions, modes), or measurements by external, temporary sensors.
- The DT uses (real-time) data traces and (design-time) models to enable learning, reasoning, dynamically recalibrating or self-adapting its behavior, as well as assistance for improved decision making.

Kinds of Digital Twins -1

For a detailed reading

Depending on the purpose, there are different kinds of digital twins. The following lists elaborates some main types of DTs, without assuming this list is complete:

"Development Digital Twin" is a central part of the development process. It contains a coherent set of development artefacts, which describe the system under development.

- These development artifacts are typically models of various viewpoints and in various abstractions, describing structure, behavior, interactions, decomposition forms, geometry, functionality (both mechanical and computational), requirements, design decisions, architecture, etc.
- The development DT may also contain assembly plans (mechanical) and build procedures (IT)
- The development DT may also contain information about the development process, which consists of tasks, planned and executed iterations, artifact states, milestones in traditional processes, scrums in agile processes, etc.
- A development DT could consist of informal documents with textual descriptions, but using a model-based development process is probably more efficient.

"Usage Digital Twin" is a set of services that helps the owner respectively the user of a system to understand the current state, the history as much as possible and potentially also to plan future actions

- E.g. a car DT mainly deals with information that is relevant to users, drivers and potentially fleet owners, such as gas filling, next time to service, customization of the display e.g. with additional services, managing the navigation system, etc.
- A production line DT deals with maintenance and production information for the current workers as well as the management.
- If the usage DT also has influence on the system itself, then one could also speak of a "Controlling Digital Twin". To enable control, a feedback channel to the system is needed, allowing to adapt its behavior according to certain given restrictions.

Kinds of Digital Twins -2

For a detailed reading

"Diagnostic Digital Twin" is a set of services that helps maintenance to understand the state of operation, failures and repair operations.

- A car diagnostic DT in all its features is used during maintenance and could typically be equipped with additional tooling to execute detailed diagnostics and repair operation.
- Diagnostic DTs may also be of interest for the manufacturer.
- In a production line, usage and diagnostic DT may coincide (or even more variants of them may exist).

"Autonomous Collaboration DT" is an active set of services that connects itself with other DTs of this type to collaborate on achieving certain functionality

- The Autonomous Collaboration DT acts like an autonomous agent, needs to understand its context in sufficient detail, and react on changing environments in a robust way. Example: collaborative autonomous driving.
- The Autonomous Collaboration DT also has the ability to control the underlying system. If a fleet (set) of systems shall act together and in a sufficiently synchronized and reliable way, then the fleet (set) of Autonomous Collaboration DT coordinates this reliably.

"Monitoring DT" is a set of services that collects relevant data traces and proofs correct behavior versus certain juristic and technical restrictions.

- A monitoring DT e.g. traces, whenever a rented car has not left the official rental area, a production engine wasn't operated outside its parameters, etc.
- A monitoring DT may e.g. be used by owners, insurances, etc.
- Airplanes e.g. are equipped with (not yet connected) black boxes.

"Business DT" is used by the owner or manufacturer to understand the past, current and future financial conditions of the operation for a system.

- This may include the capacity utilization of a fleet of rental cars, or the degeneration and replacement conditions of spare parts.

"Production DT" is used by the manufacturer during the production or assembly of a product to understand its state respectively the states of all its components.

Remarks:

- The development DT has the characteristics of a “type”, because it may have a lot of instances (real systems built). It therefore also does not have data traces, except potentially from simulations used during development.
- All other kinds of DTs are based on a concrete instances of the product. They can share runtime data in form of data traces, which are unique to the concrete product.
- Thus development DT and all other DT significantly differ in use and offered services, even though it may be that an extraction/derivation of the development DT is needed in the other DTs.

Digital Twin and CPS

For a detailed reading

Digital Twin and CPS

Definition: Cyber-physical systems (CPS) are engineered systems where functionalities are emerging from the networked interaction of physical and computational processes. [BDS19]

- The boundary between CPS and digital twin is blurred, since a CPS already contains software
- The DT can be considered as a logical unit. This does not mean that it may only be executed by one computational unit (Hardware), but can be distributed, virtualized and increasingly often lives mainline in a cloud.
- There exist other software system in parallel to the DT, such as SAP management components, the DT can communicate with those
- The existence of the DT and the CPS should largely overlap, they exist roughly at the same time
- Logically, there is only one digital twin. If there would be several twins, they would not be twins, but triplets or more.
  - If two digital siblings would try to control one physical twin, that could become complicated

- Logically, there is only one digital twin. If there would be several twins, they would not be twins, but triplets or more.
  - If two digital siblings would try to control one physical twin, that could become complicated

- Logically, each CPS has a digital twin on its own. But digital twins of several CPS may
  - share data, e.g. for reasons of learning from each other
  - exchange certain information for cooperation (which actually means they are composed)
  - share the same implementing system (platform, software) or even the same engineering models, because they originate from the same development project

For a detailed reading

Further Discussion on Digital Twins -1

The DT is always the DT of an original. The definition mainly applies to the DTs of systems, which includes cars, production, medical devices, buildings and even city quarters, but not processes, nor humans or things occurring in nature, although adaptions could be easily possible.

- E.g. the digital twin of a human could be of interest in society, medical or also industrial context, e.g. describing capabilities, workload, etc.

A conceptually well defined digital twin can be technically distributed over a set of nodes including the system, local devices (edge computers) and the cloud.

- Engineering challenges result here, but this needs to be tackled.

Because the services of a DT are intensively connected to the functional capabilities of the physical system, it may well be that the development process has to concentrate on physical system and digital twin in parallel. The digital twin therefore is NOT only a spin-off product of the development of the physical system.

- We speak of "product service system" as the integrated, holistic system consisting of the physical product and the associated services that are available through its digital twin.

Because the system may be configurable, can be parameterized or calibrated and new services may be coming over time, the DT might be highly "self-adaptive"

- "Models@Runtime" or in this case the explicit management of models in a digital twin allows to identify sources of data traces as parts of the system's models.

- "Models@Runtime" also provide a highly customizable and therefore adaptive mechanism. Adaptivity through explicit models provides another set of engineering challenges.

- We distinguish three different main forms of adaptation:
- (1) Autonomous self adaptation, e.g. induced by changes of the context, optimizations identified for example through continuous measurements or by a slow degradation of the system itself.
- (2) The user wishes to adapt the system behavior.
- (3) The manufacturer adapts the system behavior according to identified optimizations, fixing of bugs and failures, change of juristical side conditions or upgrade of functionality.

- Managing "Models@Runtime" enforces additional infrastructure, such as appropriate meta-models or explicit system and self-monitoring.

- Parts of the "Models@Runtime" may be physically embedded within the system to achieve appropriate performance and reliability. Others may be hosted externally (e.g. in the cloud).

For a detailed reading

Further Discussion on Digital Twins -2

Complex systems are decomposed.

Individual subsystems or components of the systems
- (1) may have an individual history,
- (2) are developed in the individual sub-projects potentially even outsourced to suppliers,
- (3) may be reused from predefined sets of available components, or
- (4) may be replaced over time.

Therefore, the DT should be "structurally composed" similar and in accordance to the product.

In addition to the product that only knows its current state, the DT should know about all historical structural changes and replacements.

The DT therefore as to cope with several dimensions of variability:
- (1) replacements of subcomponents,
- (2) recalibration and change of parameters of operation of the system,
- (3) change of available services and service-variants, and
- (4) deviations of the engineered system from the produced system.

These dimensions of variability all exist in addition to the typical engineering variability of different variants of systems.

Data Lake

- Repository of data stored in its natural/raw format, usually object blobs or files
  - Include structured data from relational databases (rows and columns)
  - Semi-structured data (CSV, logs, XML, JSON)
  - Unstructured data (emails, documents, PDFs)
  - Binary data (images, audio, video)
- Data Lake stores all data - regardless of relevance, structure and purpose
- Data stored independent of source, and structure
  - Remain in original form and only prepared when needed
  - "Schema on Read"-Principle: data is only structured when it is read
- Data Lake e.g., based on Hadoop
  - Distributes the storage and computation of the data over many nodes of a cluster
  - Data in large quantities can be processed quickly