University of Stuttgart

Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW)

Digital Twins

And the Asset Administration Shell

Prof. Dr.
rer. nat. habil.
Andreas
Wortmann

Presenting the Joint Work of all Five of Us

Every information about an asset in a single place

Jingxi Zhang

Institute for Control Engineering of Machine Tools and Manufacturing Units

University of Stuttgart

Carsten Ellwein

Institute for Control Engineering of Machine Tools and Manufacturing Units

University of Stuttgart

Malte Heithoff

Software Engineering

RWTH Aachen University

Judith Michael

Software Engineering

RWTH Aachen University

Andreas Wortmann

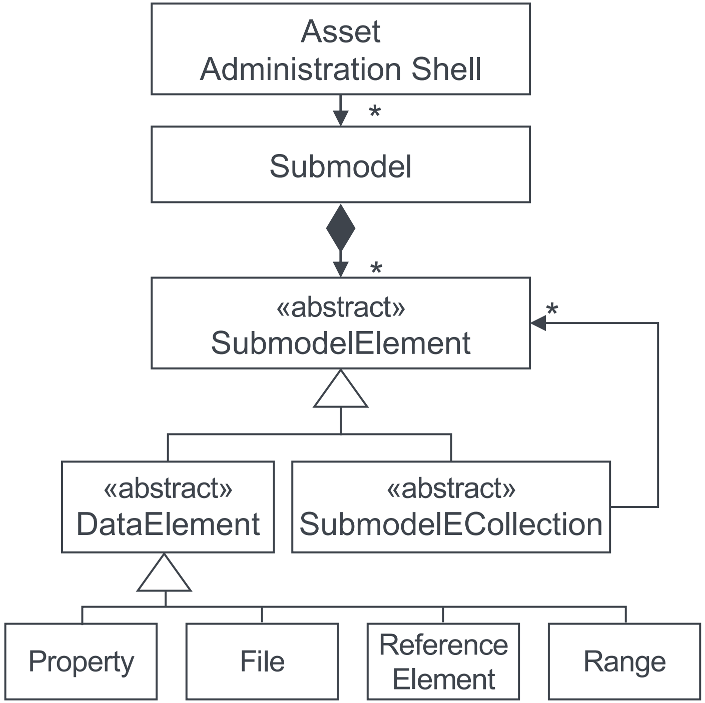
Institute for Control Engineering of Machine Tools and Manufacturing Units

University of Stuttgart

The Asset Administration Shell is an Industrial Modelling Success Story

Every information about an asset in a single place

- Essentially a metamodeling framework for asset information (including data, models, and more)
- Standardized by Industrial Digital Twin Association (IDTA)[1]
- 129 members, incl. ABB, Bosch, Danfoss, Dassault, Hitachi, Huawei, Mitsubishi, PTC; Siemens, SAP, Trumpf, VW, …
- Core metamodel[2] building on industry standards
  - Data model based on ISO 13584-42, IEC 61360
  - ECLASS[1] ISO/IEC compatible data modeling standard
- Goal: Standard for Digital Twins in Industry
- Eclipse BaSyx for implementation (among others)


Asset Administration Shell
*
Submodel
*
«abstract» SubmodelElement
*
«abstract» DataElement
«abstract» SubmodelECollection
Property
File
Reference Element
Range

1. https://industrialdigitaltwin.org
2. https://industrialdigitaltwin.org/wp-content/uploads/2023/04/IDTA-01001-3-0_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf

The AAS aims to Locate all Relevant Information About an Asset Centrally

Submodels are the main content of asset administration shells

Capabilities

- End milling
- Drilling

Maintenance

- Cutting time
- Latest service

Operational Data

- Sensor data
- Energy consumption

Technical Data

- Max. spindle speed
- Axis count

Digital Nameplate

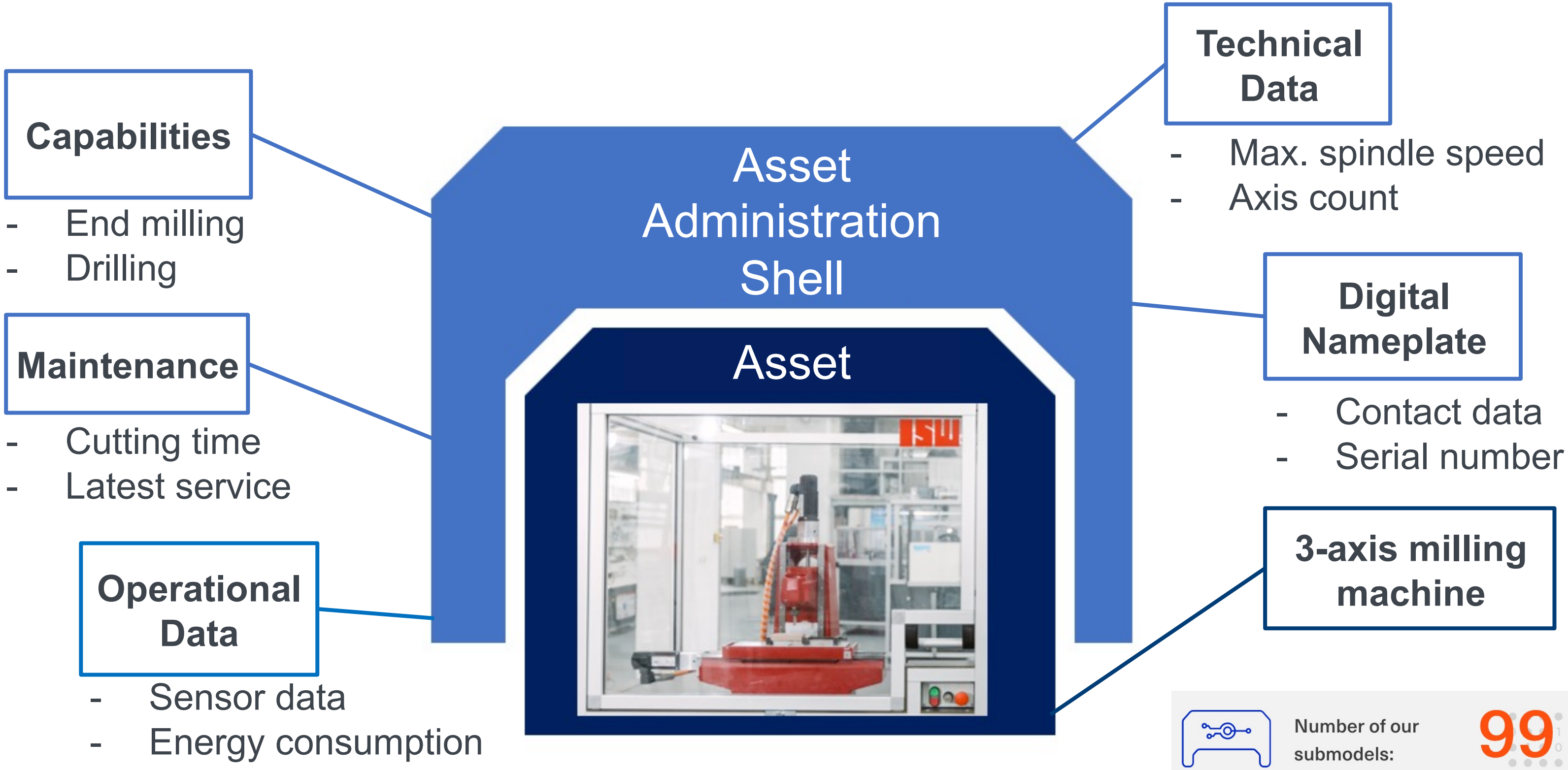
- Contact data
- Serial number

3-axis milling machine

Number of our submodels: 99

The AAS aims to Locate all Relevant Information About an Asset Centrally

Submodels are the main content of asset administration shells



- End milling
- Drilling

- Cutting time
- Latest service

- Sensor data
- Energy consumption

- Max. spindle speed
- Axis count

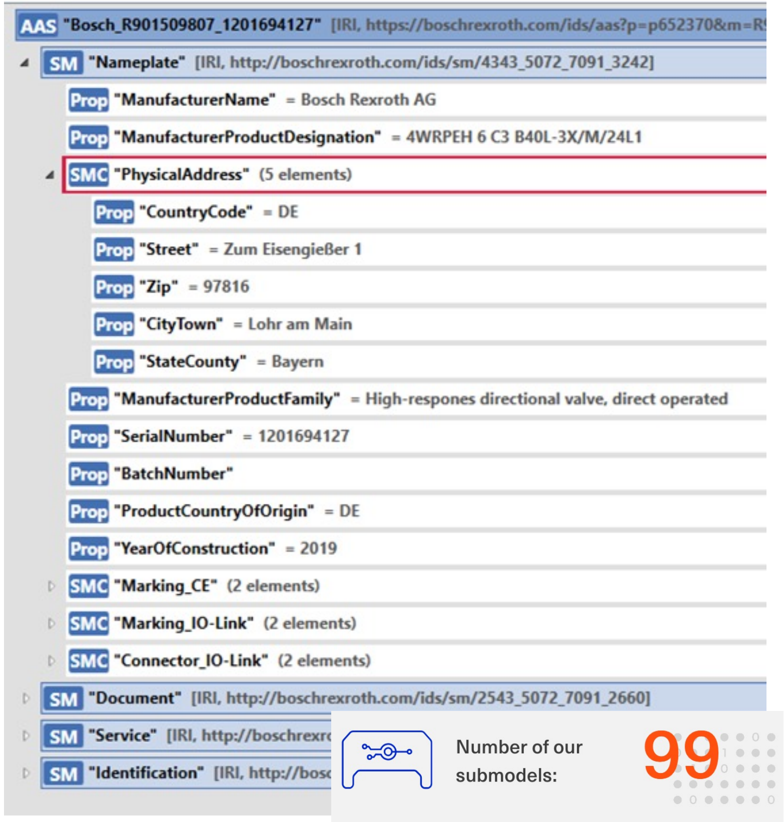
- Contact data
- Serial number

Number of our submodels: 99

AAS Submodel Templates Facilitate Structuring Asset Information

Standardizable data models within the IDTA

- Ca. 40 submodel templates published[1], incl.
  - Digital Nameplate
  - Provision of Simulation Models
  - Handover Documentation
  - Bill of Material
  - Asset Interfaces Description
- Rest under development, incl.
  - Software Bill of Materials
  - Nameplate for Software in Manufacturing
  - Digital Battery Passport
- Or build your own submodel templates



```
AAS "Bosch_R901509807_1201694127" [IRI, https://boschrexroth.com/ids/aas?p=p652370&m=R
◢ SM "Nameplate" [IRI, http://boschrexroth.com/ids/sm/4343_5072_7091_3242]
    Prop "ManufacturerName" = Bosch Rexroth AG
    Prop "ManufacturerProductDesignation" = 4WRPEH 6 C3 B40L-3X/M/24L1
  ◢ SMC "PhysicalAddress" (5 elements)
      Prop "CountryCode" = DE
      Prop "Street" = Zum Eisengießer 1
      Prop "Zip" = 97816
      Prop "CityTown" = Lohr am Main
      Prop "StateCounty" = Bayern
    Prop "ManufacturerProductFamily" = High-respones directional valve, direct operated
    Prop "SerialNumber" = 1201694127
    Prop "BatchNumber"
    Prop "ProductCountryOfOrigin" = DE
    Prop "YearOfConstruction" = 2019
  ▷ SMC "Marking_CE" (2 elements)
  ▷ SMC "Marking_IO-Link" (2 elements)
  ▷ SMC "Connector_IO-Link" (2 elements)
▷ SM "Document" [IRI, http://boschrexroth.com/ids/sm/2543_5072_7091_2660]
▷ SM "Service" [IRI, http://boschrexr
▷ SM "Identification" [IRI, http://bosc
```

Number of our submodels: 99

[1] https://industrialdigitaltwin.org/en/content-hub/submodels

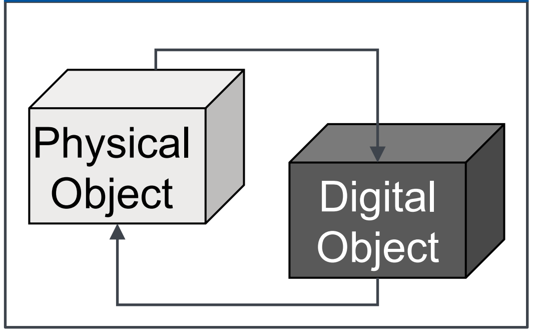
Question

Is the Asset Administration Shell a Digital Twin?

A Digital Twin is a Software System using Data, Models, and Services

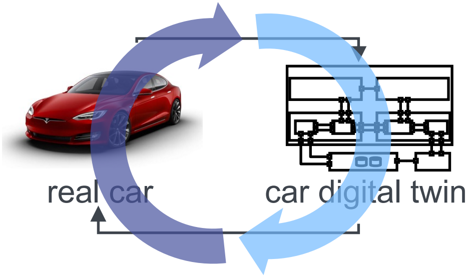
That can represent and control its (cyber-physical) counterpart

Flow-Based DT Model[1]



Physical Object

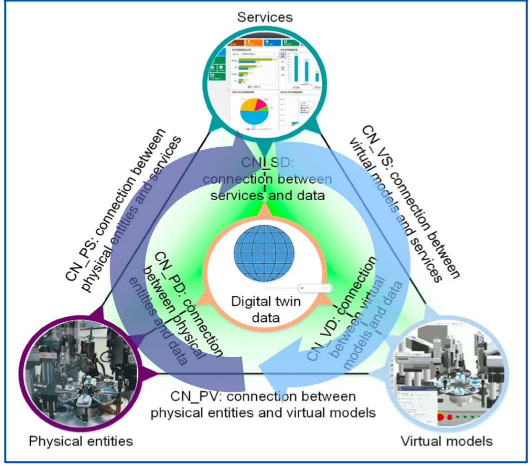
Digital Object



real car

car digital twin

5D Digital Twin Model[2]



Services

CN_SD: connection between services and data

CN_PS: connection between physical entities and services

CN_VS: connection between virtual models and services

CN_PD: connection between physical entities and data

Digital twin data

CN_VD: connection between virtual models and data

CN_PV: connection between physical entities and virtual models

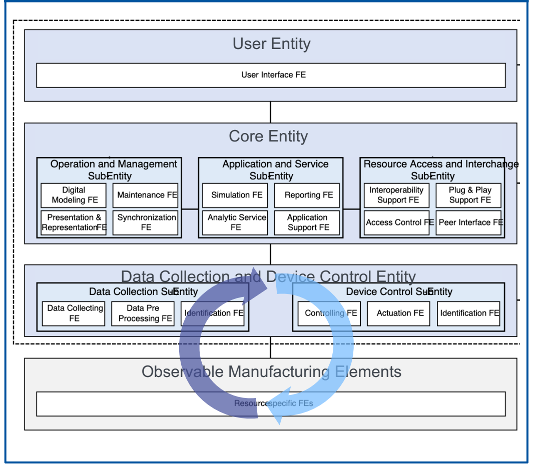
Physical entities

Virtual models

5 Dimensions

(1) CPS, (2) Data, (3) Models, (4) Services, (5) Connections

DT for Manufacturing[3]



User Entity

User Interface FE

Core Entity

Operation and Management SubEntity: Digital Modeling FE, Maintenance FE, Presentation & Representation FE, Synchronization FE

Application and Service SubEntity: Simulation FE, Reporting FE, Analytic Service FE, Application Support FE

Resource Access and Interchange SubEntity: Interoperability Support FE, Plug & Play Support FE, Access Control FE, Peer Interface FE

Data Collection and Device Control Entity

Data Collection SubEntity: Data Collecting FE, Data Pre Processing FE, Identification FE

Device Control SubEntity: Controlling FE, Actuation FE, Identification FE

Observable Manufacturing Elements

Resource specific FEs

Observable Manufacturing Elements:

Physical, biological, chemical, virtual, …

[1] Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W: Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine, 2018.

[2] Qi et al.: Enabling technologies and tools for digital twin. In: Journal of Manufacturing Systems, Elsevier, 2019

[3] ISO 23247. Digital Twin Framework for Manufacturing, 2021.

There are Three Types of Asset Administration Shells

At different levels of maturity

Type 1 AAS

- Shells are serialized files
- Contain static information
- Data model governed by AAS meta model
- Describe types and instances of assets as-designed
- No automated dataflows from/to asset

Type 2 AAS

- Runtime instances: may contain static and dynamic information from asset
- Interact w. other components
- Ex: frontend for device services, live sensor data, …
- Properties, operations, events via generic runtime interface
- Automated dataflows only from asset

Type 3 AAS

- Extend type 2 AAS
- Have active behavior
- Can start to communicate & to negotiate on their own
- Well-defined language and message structures (VDI/VDE 2193)
- Automated dataflows from/to asset

Question

How Much Digital Twin do the Three Asset Administration Shell Types Support?

We opted to Analyze very Popular and Important Digital Twin Definitions

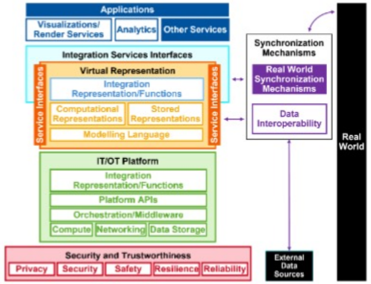
Omitted many less useful defintions[1]

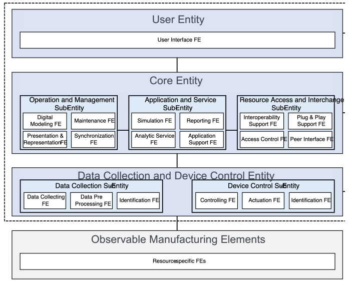
ACADEMIC DEFINITIONS

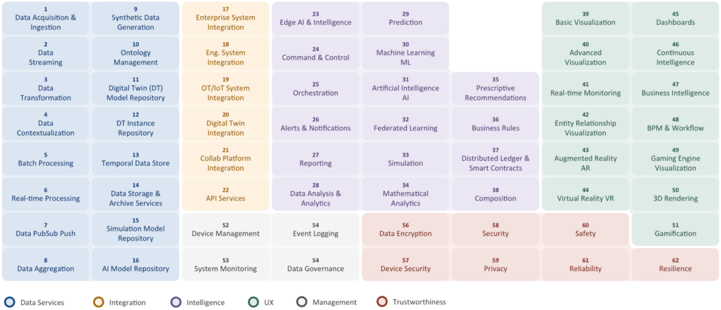
- Based on data flows[2] (Kritzinger et. al)
- Based on dimensions[3] (Qi et. al)

INDUSTRIAL DEFINITIONS

- ISO DT framework for manufacturing[4]
- Digital Twin Consortium (DTC) platform stack architecture for digital twins[5]
- DTC digital twin capability periodic table[6]







1 Data Acquisition & Ingestion	9 Synthetic Data Generation	17 Enterprise System Integration	23 Edge AI & Intelligence	29 Prediction		39 Basic Visualization	45 Dashboards
2 Data Streaming	10 Ontology Management	18 Eng. System Integration	24 Command & Control	30 Machine Learning ML		40 Advanced Visualization	46 Continuous Intelligence
3 Data Transformation	11 Digital Twin (DT) Model Repository	19 OT/IoT System Integration	25 Orchestration	31 Artificial Intelligence AI	35 Prescriptive Recommendations	41 Real-time Monitoring	47 Business Intelligence
4 Data Contextualization	12 DT Instance Repository	20 Digital Twin Integration	26 Alerts & Notifications	32 Federated Learning	36 Business Rules	42 Entity Relationship Visualization	48 BPM & Workflow
5 Batch Processing	13 Temporal Data Store	21 Collab Platform Integration	27 Reporting	33 Simulation	37 Distributed Ledger & Smart Contracts	43 Augmented Reality AR	49 Gaming Engine Visualization
6 Real-time Processing	14 Data Storage & Archive Services	22 API Services	28 Data Analysis & Analytics	34 Mathematical Analytics	38 Composition	44 Virtual Reality VR	50 3D Rendering
7 Data PubSub Push	15 Simulation Model Repository	52 Device Management	54 Event Logging	56 Data Encryption	58 Security	60 Safety	51 Gamification
8 Data Aggregation	16 AI Model Repository	53 System Monitoring	54 Data Governance	57 Device Security	59 Privacy	61 Reliability	62 Resilience

○ Data Services ○ Integration ○ Intelligence ○ UX ○ Management ○ Trustworthiness

[1] List of 112 definitions of the term digital twin: http://www.wortmann.ac/digital-twin-definitions
[2] Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W: Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine, 2018.
[3] Qi et al.: Enabling technologies and tools for digital twin. In: Journal of Manufacturing Systems, Elsevier, 2019
[4] ISO 23247. Digital Twin Framework for Manufacturing, 2021.
[5] https://www.digitaltwinconsortium.org/2023/11/understanding-dtcs-digital-twin-platform-stack-architectural-framework/
[6] https://www.digitaltwinconsortium.org/initiatives/capabilities-periodic-table/

Common Requirements on Digital Twins

Based on popular definitions, characterizations, and architectures

Req.	The digital twin …	Sources
R1	can **receive data** from twinned counterpart	All
R2	can **send data** to its twinned counterpart	All
R3	has a **user interface**	DTC, ISO
R4	can **represent** its counterpart digitally	All
R5	can **synchronize** (selected) properties with its counterpart.	Flows, ISO
R6	can **report information** to selected recipients aside from the AAS, e.g., by sending a message to the asset's operator	DTC, ISO
R7	can communicate with **other digital twins**	DTC, ISO
R8	can interact with **third-party systems**	DTC, ISO
R9	provides **services** to act on data and models.	5D, DTC, ISO
R10	can **reason** about data from/about the twinned counterpart as well as about data obtained from other systems	DTC, ISO

What IDTA Requires for a Type 1 AAS Based on Various White Papers

Type 1 AAS are serialized files that contain static information

Req.	Evaluation	Eval	Explanation
R1	**receive data**	○	Just an XML file without any activity
R2	**send data**	○	
R3	**user interface**	○	
R4	**represent**	●	Can represent asset information statically
R5	**synchronize**	○	
R6	**report information**	○	
R7	comm. w. **digital twins**	○	
R8	interact w. **3rd-party systems**	○	
R9	**services**	○	
R10	can **reason**	○	

IDTA requires ● suggests ◐ does not require ○ that feature

What IDTA Requires for a Type 2 AAS Based on Various White Papers

Type 2 AAS are runtime things: may have static/dynamic information from asset

Req.	Evaluation	Eval	Explanation
R1	**receive data**	●	Shall receive asset data via any connection
R2	**send data**	○	
R3	**user interface**	○	
R4	**represent**	●	Can represent system statically and dynamically
R5	**synchronize**	◐	Supports unidirectional, timed synchronization
R6	**report information**	◐	Can use references between submodels to emulate
R7	comm. w. **digital twins**	○	
R8	interact w. **3rd-party systems**	○	
R9	**services**	◐	There is an API for manipulating submodels
R10	can **reason**	◐	There is an API for manipulating submodels

IDTA requires ● suggests ◐ does not require ○ that feature

What IDTA Requires for a Type 3 AAS Based on Various White Papers

Type 3 AASs encompass Type 2 and yield software modules for added-value

Req.	Evaluation	Eval	Explanation
R1	**receive data**	●	Active bidirectional communication with asset
R2	**send data**	●	Active bidirectional communication with asset
R3	**user interface**	○	
R4	**represent**	●	Can represent system statically and dynamically
R5	**synchronize**	◐	Supports unidirectional, timed synchronization
R6	**report information**	◐	Can use references between submodels to emulate
R7	comm. w. **digital twins**	●	Possible via “Industry 4.0 language”
R8	interact w. **3rd-party systems**	●	Required via “Industry 4.0 language”
R9	**services**	◐	There is an API for manipulating submodels
R10	can **reason**	◐	There is an API for manipulating submodels

IDTA requires ● suggests ◐ does not require ○ that feature

Type 3 Asset Administration Shells are Digital Twins

The others not

Type 1 AAS

- Shells are serialized files
- Contain static information
- Data model governed by AAS meta model
- Describe types and instances of assets as-designed
- No automated dataflows from/to asset

→**Idealized, static, description of an asset**

Digital Model

Type 2 AAS

- Runtime instances: may contain static and dynamic information from asset
- Interact w. other components
- Ex: frontend for device services, live sensor data, …
- Properties, operations, events via generic runtime interface
- Automated dataflows only from asset

→**Well-informed Dashboard**

Digital Shadow

Type 3 AAS

- Extend type 2 AAS
- Have active behavior
- Can start to communicate & to negotiate on their own
- Well-defined language and message structures (VDI/VDE 2193)
- Automated dataflows from/to asset
- **Software interfacing asset**

Digital Twin

Challenges and Opportunities for Future Research

Also exploitation

1. **Deriving Twins.** Engineering model, e.g., AutomationML or SysML, comprise vast asset knowledge (connectivity, sensors, actuators, error modes, …) , yet much of this needs to be re-developed for the digital twins → needs means to derive parts of digital twins from engineering models
2. **Component Reuse**. Digital twins have many components to process data, interact with models, communicate to the asset, and run services, yet reusing these components between digital twins is hardly possible → needs digital twin reference architecture with well-defined component interfaces
3. **Digital Twin Reuse.** Systems often comprise subsystems, hence their twins should comprise sub-twins. The composition of digital twins is far from solved → needs systematic method to compose smaller digital twins into larger ones.
4. **Low-Code Configuration**. Digital twins will be configured, used, and understood by domain experts with little programming knowledge. Expecting them to grasp OO data models or stack traces is futile → needs DSLs and interfaces to properly configure and represent cross-cutting twin concerns.

University of Stuttgart
Institute for Control Engineering of Machine
Tools and Manufacturing Units (ISW)

Prof. Dr. rer. nat. habil. Andreas Wortmann

email wortmann@isw.uni-stuttgart.de
web www.wortmann.ac
phone +49 (0) 711 685-84624

University of Stuttgart
Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW)
Seidenstrasse 36 • 70174 Stuttgart • Germany

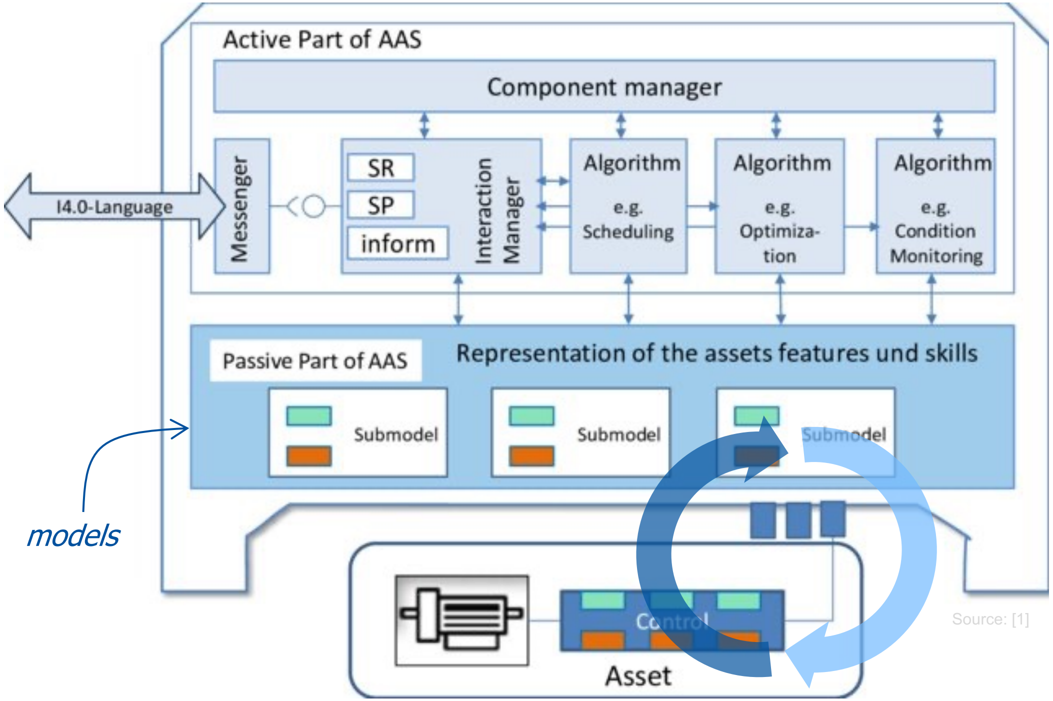
download these slides

www.wortmann.ac
/presentations

A General Architecture of Type 3 Asset Administration Shells[1]

A potential blueprint for digital twins

- Component manager orchestrates AAS behavior
- Service requester/provider interact with environment
- Algorithms perform computations
- Interact with models and asset
- Reference implementation, e.g., with BaSyx
- A lot of JSON…



Source: [1]

[1] Belyaev, A., Diedrich, C. (2019). Aktive Verwaltungsschale von Industrie 4.0 Komponenten," in Automationkongress 2019, Baden-Baden.