How to use Knowledge and Learning in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. Fundamentals of field level control
3. AI for field level control
4. Enabling Adaptive Factory and Order Controlled Production using MAS & DT
5. Conclusion and future work

Birgit Vogel-Heuser, Ordinaria

- Lehrstuhl Automatisierung und Informationssysteme (AIS)
  Fakultät Maschinenwesen
- Work@ Munich School of Robotic and Machine Learning
- Munich Data Science Institute (MDSI)

Technische Universität München (TUM)
www.ais.mw.tum.de; vogel-heuser@tum.de

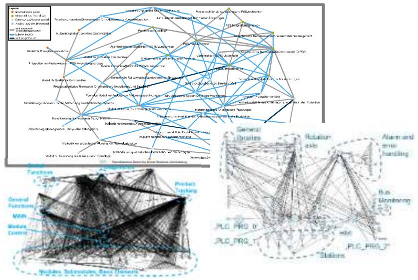
2nd Summer School on AI for Industry 4.0, France, Germany and Switzerland, 2021

Automation
and Information Systems
Technical University of Munich

Research Fields

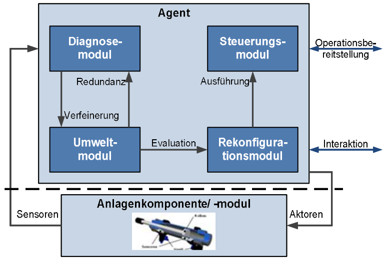
Institute of Automation and Information Systems
Faculty of Mechanical Engineering (AIS)
https://www.mw.tum.de/ais/startseite/
and Munich School of Robotic and Machine Learning (MSRM)
https://www.msrm.tum.de/msrm/research/principal-investigators/

The AIS-TUM focuses on the modeling of intelligent, distributed embedded systems in automation technology. The focus is on reliability and human-machine interaction.



MODEL-BASED DEVELOPMENT

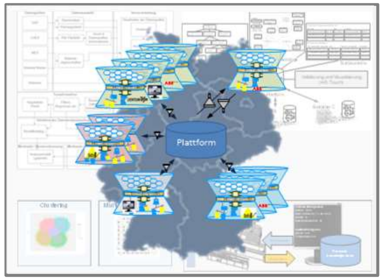
Model-based engineering of variant-rich, interdisciplinary aPS



INTELLIGENT PRODUCTION SYSTEMS

Intelligent, reconfigurable, distributed CPPS

KonMaFS DAVID advacode



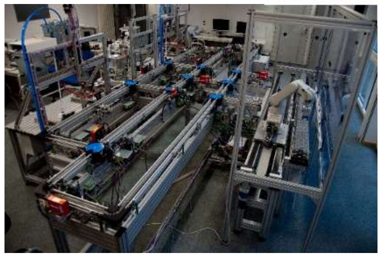
BIG DATA IN aPS

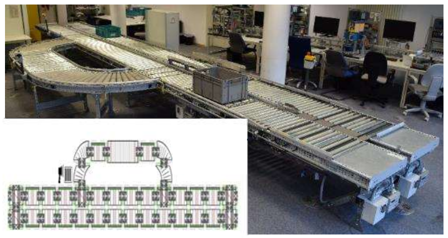
Methods for aggregation, analysis and processing of large amounts of data

AI Valve

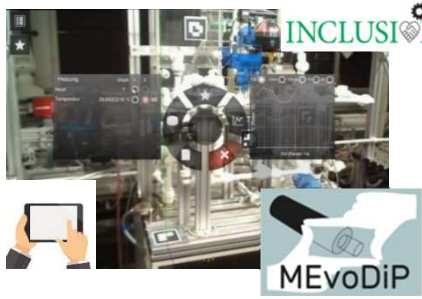
HUMAN-MACHINE INTERACTION

Representation of complex data sets to support humans in taking in information





InSeLDiP



INCLUSIVE

MEvoDiP

https://www.mw.tum.de/ais/forschung/

Learning goals and Time Schedule

Understand and be able to apply

- The specific requirements and challenges of field level agents like dependability, real-time characteristics regarding Cyber Physical Production Systems (CPPS)
- The most beneficial applications for field level agents
- How to develop DT by MAS and model their knowledge?
- Frequently used MAS pattern on field level

Time Schedule (CET time)

09:00 – 09:50 Background theory

09:50 – 10:00 5 min Q & A and 5 min feedback

Industry 4.0 architecture and scenarios AF and OCP

Data processing for humans

Assistance systems for engineering

Data processing and integration for humans

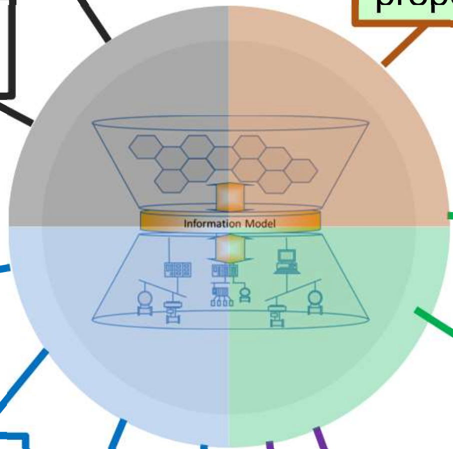
Communication and data consistency

Appropriation of necessary data for configuration, production, negotiation

World wide distribution of data, **high availability, access protection**

Data consistency about different „stakeholders“ in different engineering phases and crafts

Digital networks and interfaces for communication (between machine, human and plant, plant and plant)



Architecture models (e.g., reference architecture for I4.0 / RAMI4.0) for a category of aggregation/modules related to properties, capabilities, interfaces…

Intelligent products and production units

Production units with inherent capabilities

Data analysis of process and alarm data and connection with engineering data

I4.0 scenarios

Adaptable Factory (AF):
Self-adapting flexible production units
- to new product requirements and
- **faulty equipment**, structural changes

Order Controlled Production (OCP):
Description of PPR, e.g., ontology, for independent analysis, presentation, organization and execution of a production process

Source: B. Vogel-Heuser, G. Bayrak, U. Frank: Forschungsfragen in "Produktautomatisierung der Zukunft". acatech Materialien. 2012.

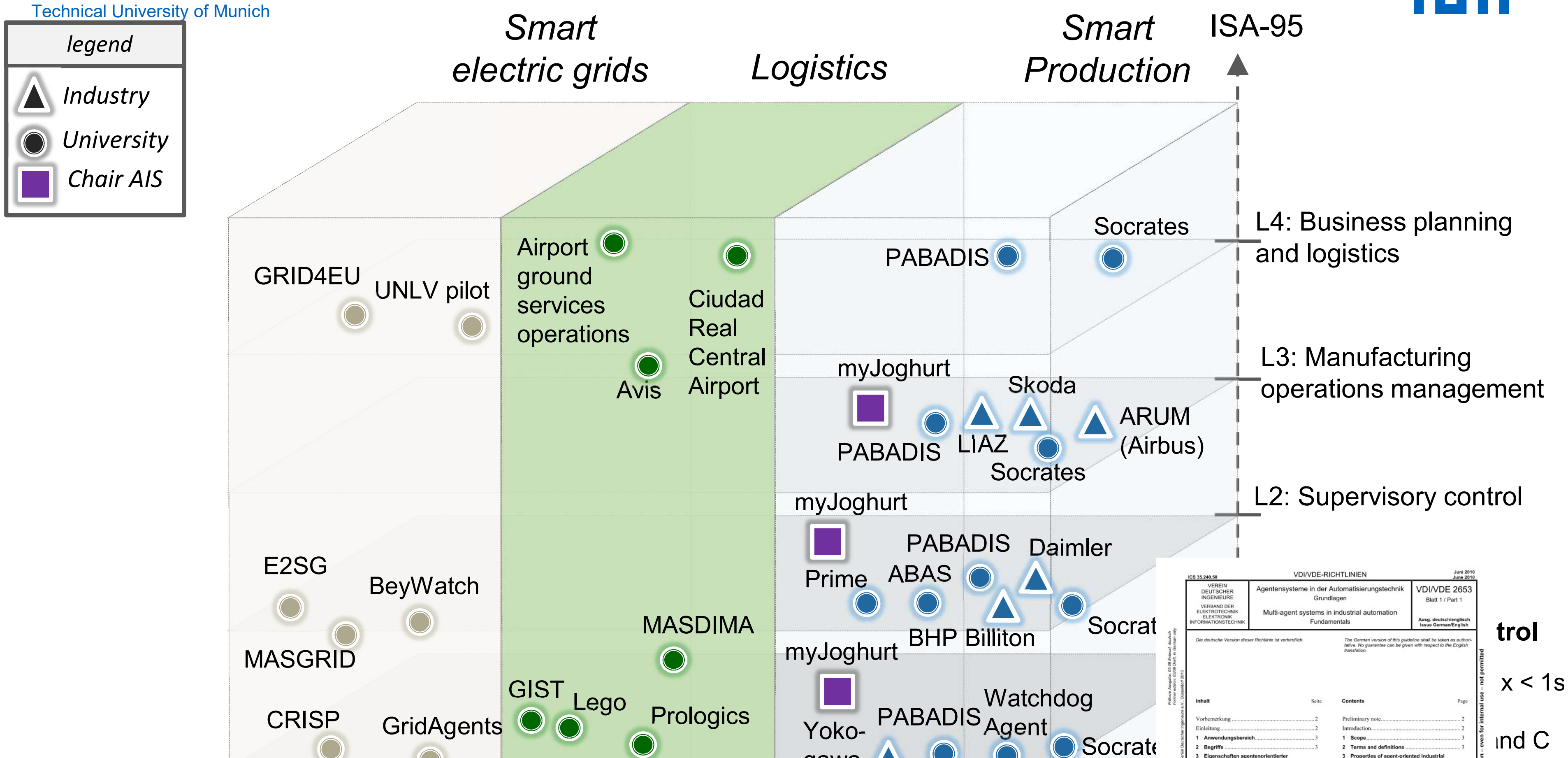
How to use Knowledge and Learning in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. **Fundamentals of field level control**
3. AI for field level control
4. Enabling Adaptive Factory and Order Controlled Production using MAS & DT
5. Conclusion and future work

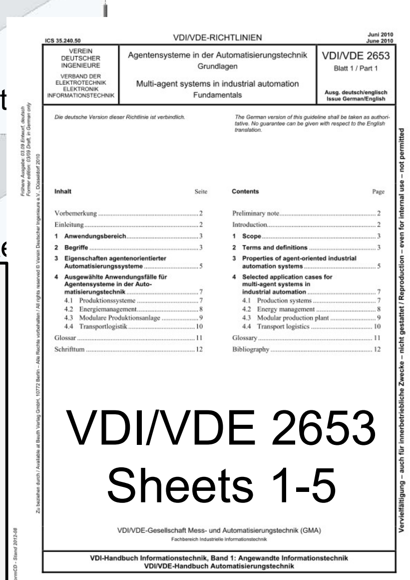
2nd Summer School on AI for Industry 4.0, France, Germany and Switzerland, 2021

Automation
and Information Systems
Technical University of Munich

State of the art- agents in CPS/CPPS



A **technical agent** is an **encapsulated** (hardware / software) entity with **specified objectives**.
An agent endeavours to reach these objectives through its **autonomous behaviour**, in **interacting** with its **environment** and with **other agents** (VDI/VDE 2653-1)



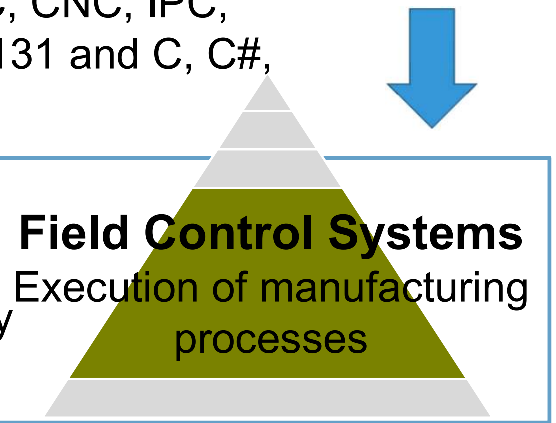
Field of application Field control systems

- Control of production processes like transport, forming, injection molding, sorting
- Control decisions
  open-loop control like interlocking - combination of sensor and actuator results in actuator activation as well as closed loop control
- Constraint
  due to cost and robustness (evolvability over decades) totally distributed control based on consumer market devices not accepted
- Integration of process-specific functions of plant safety and personal protection
- Time response: Strong requirements
  - Response times: 10ms < x < 1s
  - Synchronization: 100 μ s < x < 100ms
- Dependability: safety of humans and machines
- Operating system, hardware, programming environment: PLC, CNC, IPC, embedded systems; Programming languages: Mostly IEC 61131 and C, C#, rarely C++

- Need for real-time and dependable agents

⇒ Restricted negotiaton & communication time

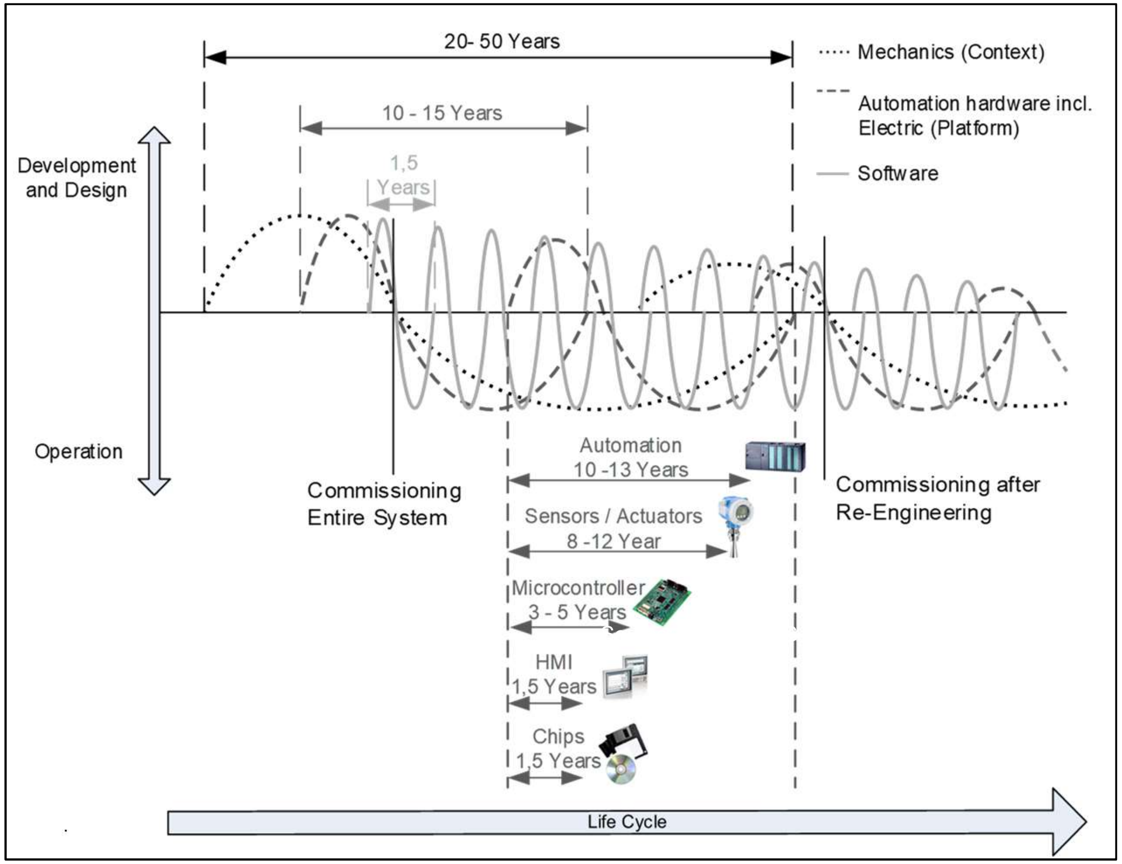
⇒ Restricted autonomy (action space) to ensure dependability

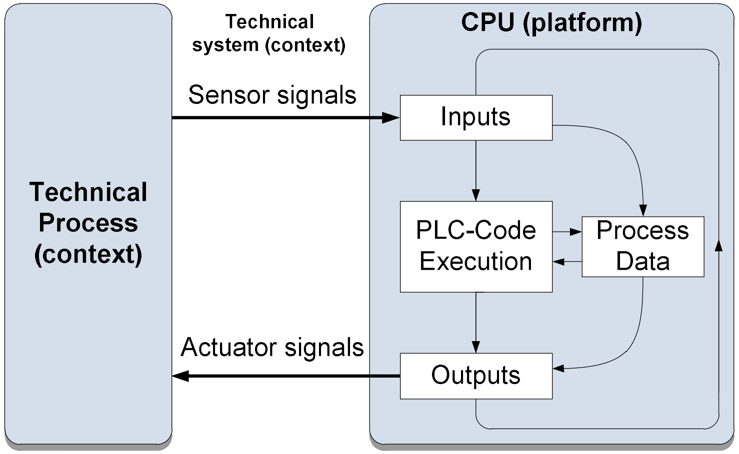


[**based on:** Lüder, A. Systems Engineering; 2018]

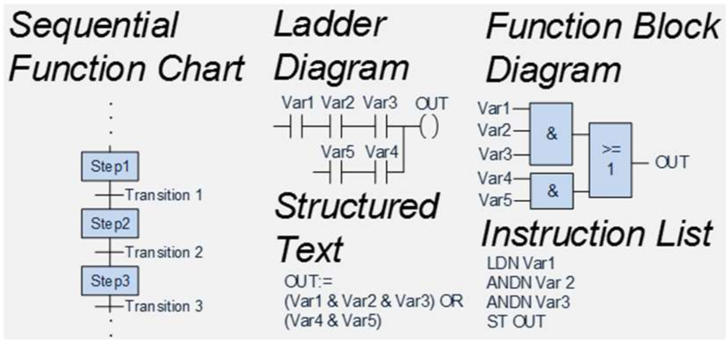
Automation
and Information Systems
Technical University of Munich

Software-evolution is the key – engineering as a basis for the software-evolution





IEC 61131-3 Languages



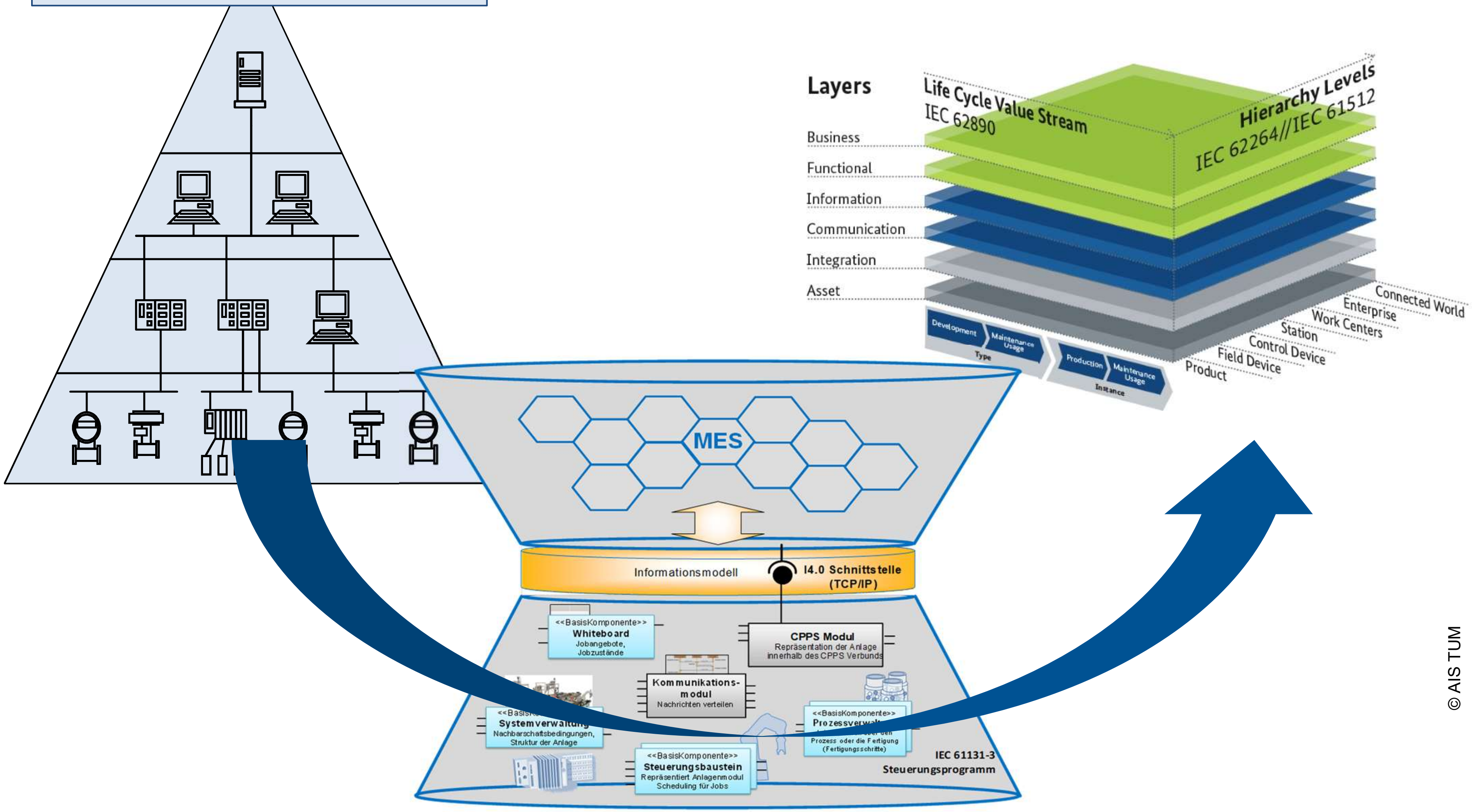
- **Importance of Software in production automation**
- 9% of a machine is software (cost) (VDMA 2008)
- 20% (average) up to 50% of engineering costs for aPS software (2012)
- 25% (average) engineering costs (forecast 2015)

Source: B. Vogel-Heuser; J. Folmer; C. Legat: *Anforderungen an die Softwareevolution in der Automatisierung des Maschinen- und Anlagenbaus*. In: at – Automatisierungstechnik, 62(3), 3/2014

Information architecture: AT pyramid, Diabolo and RAMI 4.0

Automation pyramid
1980- 2000

Reference Architectural Model Industrie 4.0 (RAMI 4.0)



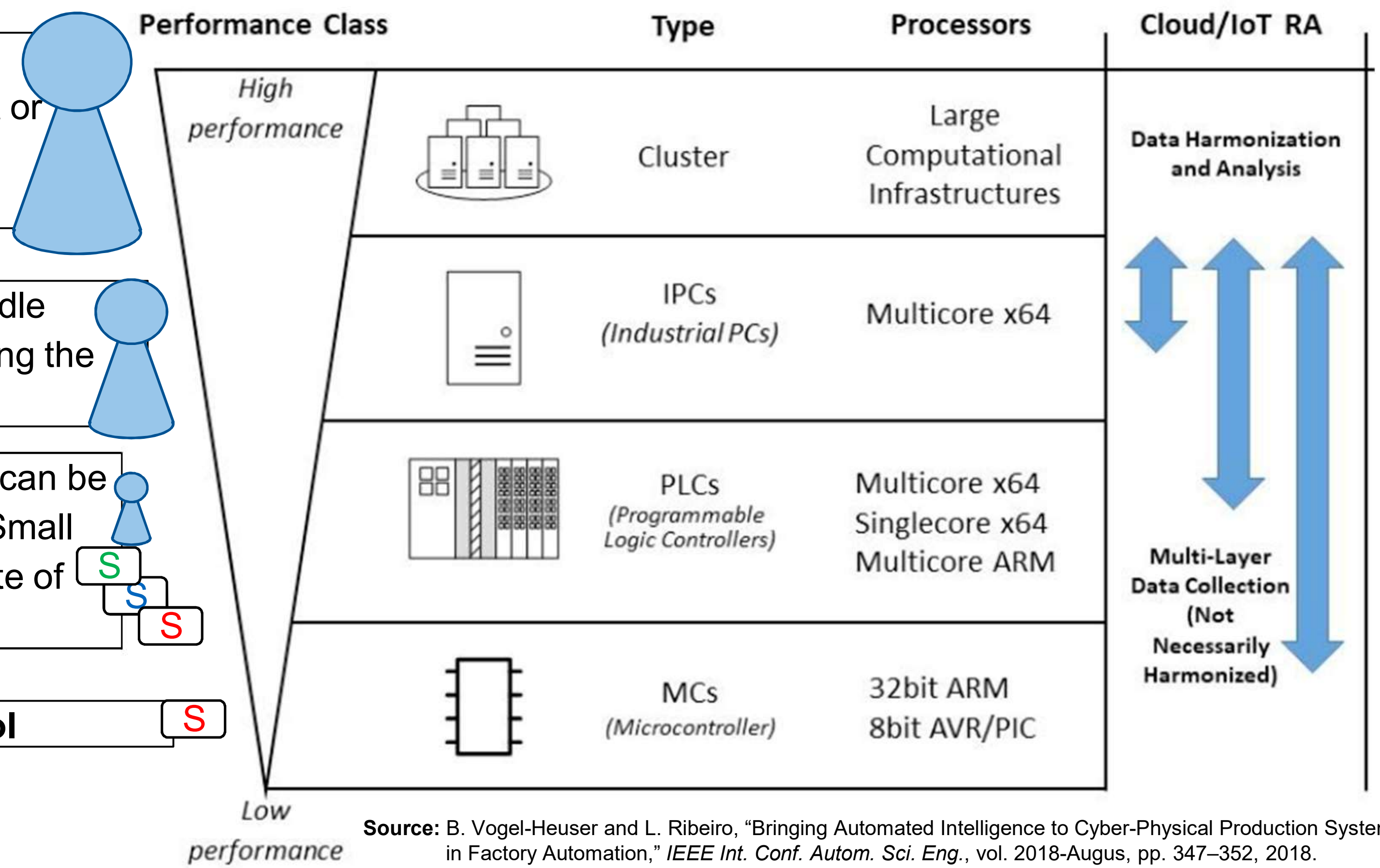
Why Agents in the Automation?

Agents using AI to coordinate deployment or other **high level organizational** tasks

Agents using AI to handle **specific tasks** regarding the process

Skills (Functions) that can be handled by an Agent. Small Agents with AI (not state of the Art)

Purely **reactive control**



Source: B. Vogel-Heuser and L. Ribeiro, “Bringing Automated Intelligence to Cyber-Physical Production Systems in Factory Automation,” *IEEE Int. Conf. Autom. Sci. Eng.*, vol. 2018-Augus, pp. 347–352, 2018.

- The increasing computational power allows smart algorithms and AI in aPS technology
- The hierarchy between Field level and higher levels becomes blurred more and more
- **Digital twin** from the industry's point of view:
- “A digital twin refers to a **virtual representation** of a physical asset or system **throughout its lifecycle**” and a “**Dynamic digital representations** that enable companies to understand, predict, and optimize the **performance** of their machines and their business”

Automation
and Information Systems
Technical University of Munich

Agent based application for Industry 4.0

Agent-based AAS (Passive, Re-active and Proactive types)

Agent-based I4.0 language (addressing VDI/VDE – 2193 standard, in collaboration with OvGU)

Resource agent pattern by Model-Driven Engineering:

(Knowledgebase, communication interface, coordination, control and process modules)

*Comparison of the IoT and DT standards

	AAS	DTDL	NGSI-LD	OData	STA	WoT
Resource Description						
Resource Term	Asset	Interface	Entity	Entity	Thing	Thing
Model Type(s)	Meta	Meta	Meta Cross-Domain	Meta	Cross-Domain	Meta
Resource Identification	IRI IRDI custom	DTMI	URI	URL custom	URL custom	URI
Type System (based on)	XSD	custom	JSON GeoJSON JSON-LD	custom	JSON SWE-standards	JSON JSON Schema
Resource Interlinking	X	X	X	X	- [a]	X
Semantic Annotation	X	O [b]	X	-	O [c]	X
Resource Elements						
Properties	X	X	X	X	X	X
Services	X	X	-	O [d]	O	X
Events	X	X	O [e]	-	O [e]	X
Serialization Format	JSON RDF XML OPC UA AutomationML	JSON RDF Avro Protobuf	JSON RDF	JSON XML	JSON	JSON RDF
Supported Kind of Data						
geo-spatial	-	-	X	X	X	-
temporal	-	-	X	X	X	-
historical	-	-	X	-	O [f]	-
Resource Discovery						
Protocols	- [a]	-	HTTP	HTTP	HTTP	HTTP [g] CoAP [g] DNS-SD [g]
Querying supported?	- [a]	-	X	X	X	O [g]
Query Language						
Query Language based on	- [a]	-	custom	custom	OData	SPARQL [a,g]
geo-spatial queries	-	-	X	X	X	-
historical queries	-	-	X	-	O [f]	-
Resource Access						
API: Define vs. Describe	define	-	define	define	define	describe
Protocols	HTTP MQTT OPC UA	-	HTTP	HTTP	HTTP MQTT	HTTP MQTT CoAP
Protocols extendible?	-	-	X	-	-	X

[a] extension under discussion; [b] only predefined definitions and only for telemetries, properties, and units; [c] only explicitly for observed properties and units, possible for everything else via custom properties; [d] only on service-level; [e] only property changes; [f] only for observations; [g] not part of standard, only in implementation(s); [x] y; Abbreviations: CoAP: Constrained Application Protocol; DNS-SD: Domain Name System - Service Discovery; HTTP: Hypertext Transfer Protocol; IRDI: International Registration Data Identifier; IRI: Internationalized Resource Identifier; JSON: JavaScript Object Notation; JSON-LD: JavaScript Object Notation - Linked Data; MQTT: Message Queuing Telemetry Transport; OPC UA: Open Platform Communications Unified Architecture; RDF: Resource Description Format; SPARQL: SPARQL and RDF Query Language; SWE: Sensor Web Enablement; URI: Uniform Resource Identifier; URL: Uniform Resource Locator; XML: Extensible Markup Language; XSD: XML Schema Definition.

***Source:** M. Jacoby and T. Usländer, "Digital Twin and Internet of Things—Current Standards Landscape," *Appl. Sci.*, vol. 10, no. 18, p. 6519, 2020

Agent based decision-making process (MAS for dynamic scheduling)

Agent based Supply Chain (addressed by Supply Chain Operations Reference- SCOR)

Agent based framework, semantic web technologies (JaCaMo project)

Automation
and Information Systems
Technical University of Munich

Extended comparison of field level control and IoT approaches

Danke @ Prof. Olivier, IMT

Classification schema*	Description*	Asset Administration Shell: AIS-TUM approach	Web of Things: IMT approach
Resource Description	Resource term	Asset	Thing
	Model type(s)	Meta	Meta
	Resource identification	IRI, IRDI, custom	URI
	Type system (based on)	XSD: XML Schema Definition	JSON, JSON schema
	Resource interlinking and semantic annotation	Yes	Yes
	Resource elements (properties, services, events)	Yes	Yes
	Serialization format	JSON, RDF, XML, OPC UA, AutomationML	JSON, RDF
Resource Discovery	Protocols	Under discussion: HTTP	No standardized: HTTP, CoAP, DNS-SD, OData
	Query Language based on	Under discussion and no standardized: SPARQL	Under discussion and no standardized: SPARQL
Resource Access	API: Define vs. Describe	API definition	API description
	Protocols	HTTP, MQTT, OPC UA, AMQP	HTTP, MQTT, CoAP
	Protocols extendible?	Yes	Yes
Categorization	Types, classes, typos, etc.	Types of interaction of AAS: Passive, Re-active, Proactive	Types of interaction affordances: Properties, Actions, and Events
Representative MAS implementation	Selected agent-based application and its components	Resource agent (RA) modules: Coordination process + Communication interface + Resource access + Knowledge Base	JaCaMo dimensions : Agent + Organization + Environment

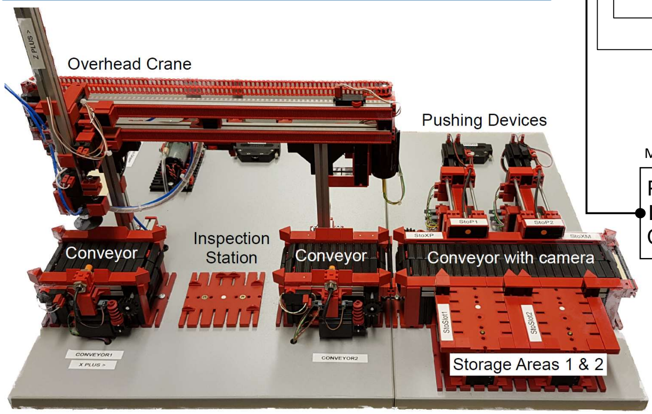
***Source:** M. Jacoby and T. Usländer, “Digital Twin and Internet of Things—Current Standards Landscape,” *Appl. Sci.*, vol. 10, no. 18, p. 6519, 2020

Network Architectures in CPPS

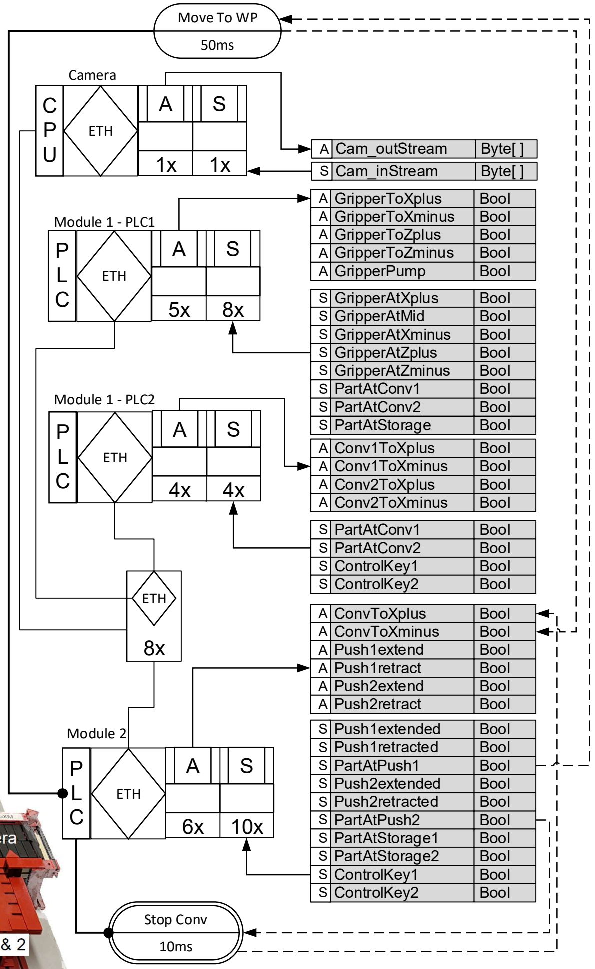
Central:
control functions concentrated in one central control unit

Decentral:
control functions divided onto several control units

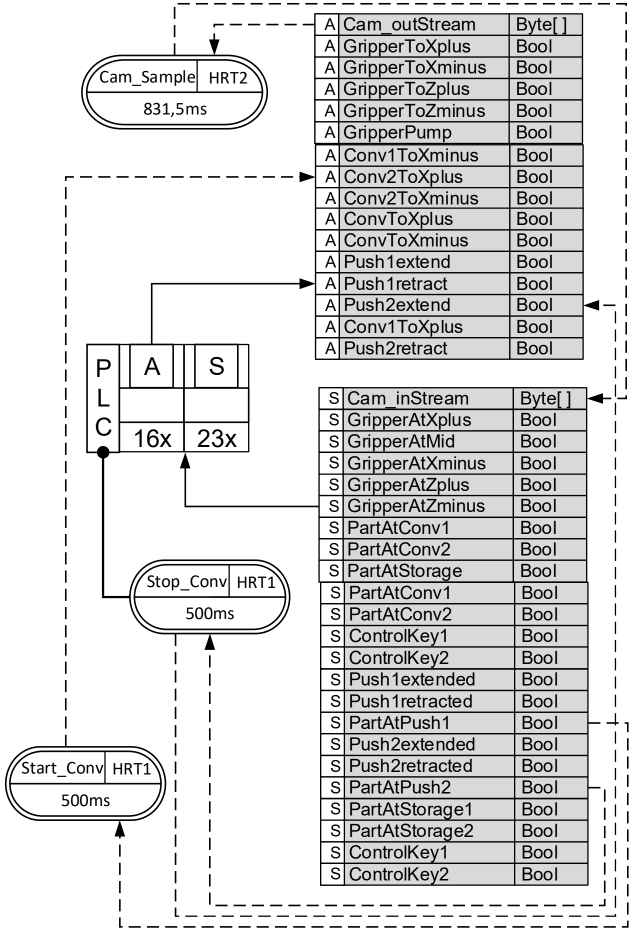
Hierarchical:
central coordinator, controlling distributed control nodes



Distributed Control



Centralized Control



Source: Hujo, D.; Vogel-Heuser, B.; Ribeiro, L. *Towards a Graphical Modelling Tool for Response-Time Requirements based on Soft and Hard Real-time Capabilities in Industrial Cyber-Physical Systems*. In: Journal of Emerging and Selected Topics in Industrial Electronics (JESTIE).

Modeling Timing behavior in agent-based CPPS

Many thanks to Prof. Luis Ribeiro

Goals

- Supporting decision for developing an agent-based CPPS
- Supporting evaluation of brown field agent-based CPPS

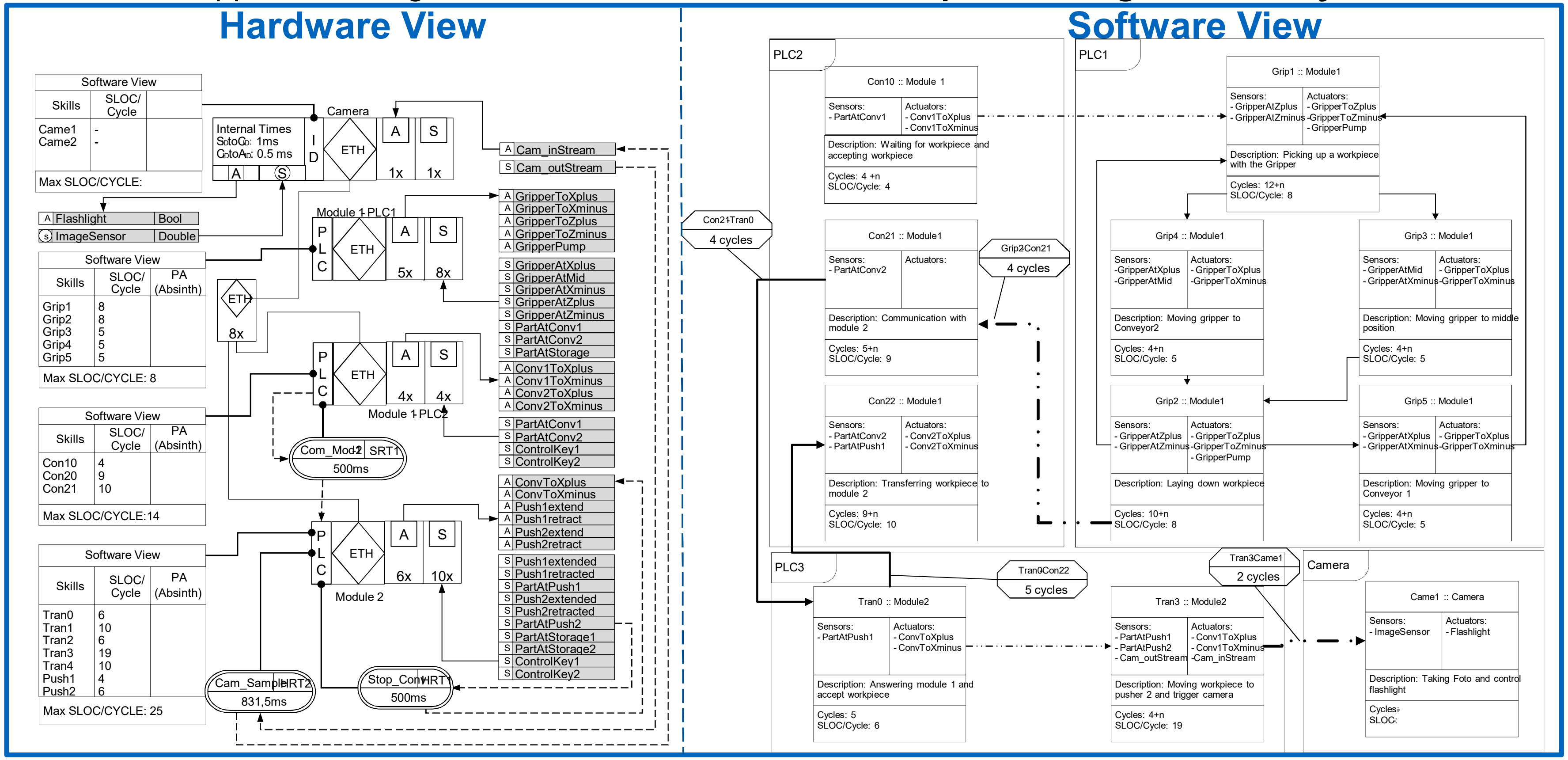
Notation for time related behavior of hard- and software:

- Heterogeneous Architecture
  →Micro Controller + PLC & Hard- and Software

Modeling timing behavior of:

- Communication
- Cycle Times (Hardware)
- Load (Hardware / Software)
- Physical Behavior
- Location of Software

"**Best Effort**" applications together with **hard real-time → Prerequisite for agent-based systems**



Source: Hujo, D.; Vogel-Heuser, B.; Ribeiro, L. *Towards a Graphical Modelling Tool for Response-Time Requirements based on Soft and Hard Real-time Capabilities in Industrial Cyber-Physical Systems*. In: Journal of Emerging and Selected Topics in Industrial Electronics (JESTIE).

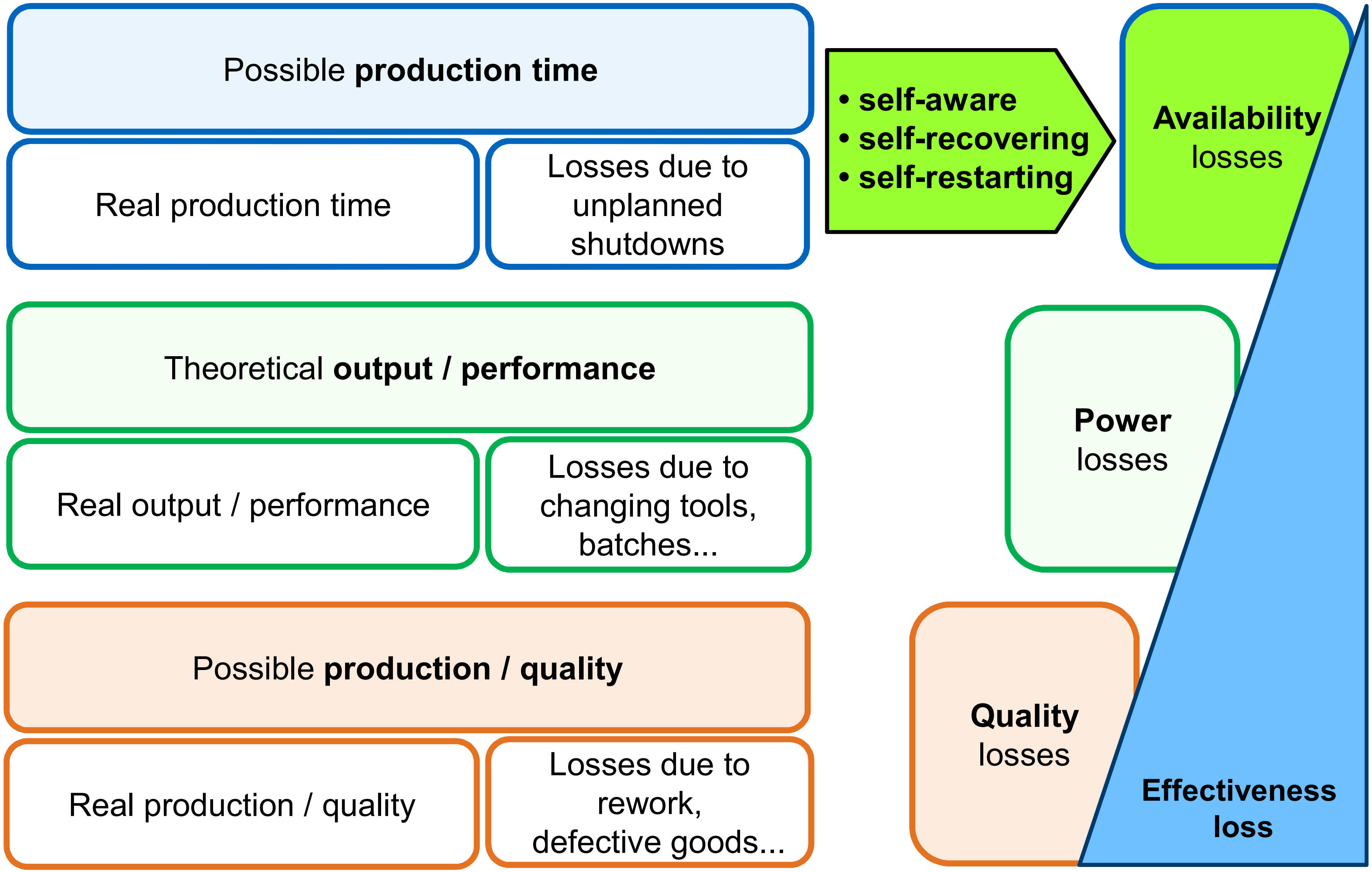
How to use
Knowledge and Learning
in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. Fundamentals of field level control
3. AI for field level control
    - **Knowledge from MDE** or data
    - Evolution of knowledge
4. Enabling Adaptive Factory and Order Controlled Production using MAS & DT
5. Conclusion and future work

2nd Summer School on AI for Industry 4.0,
France, Germany and Switzerland, 2021

Automation
and Information Systems
Technical University of Munich

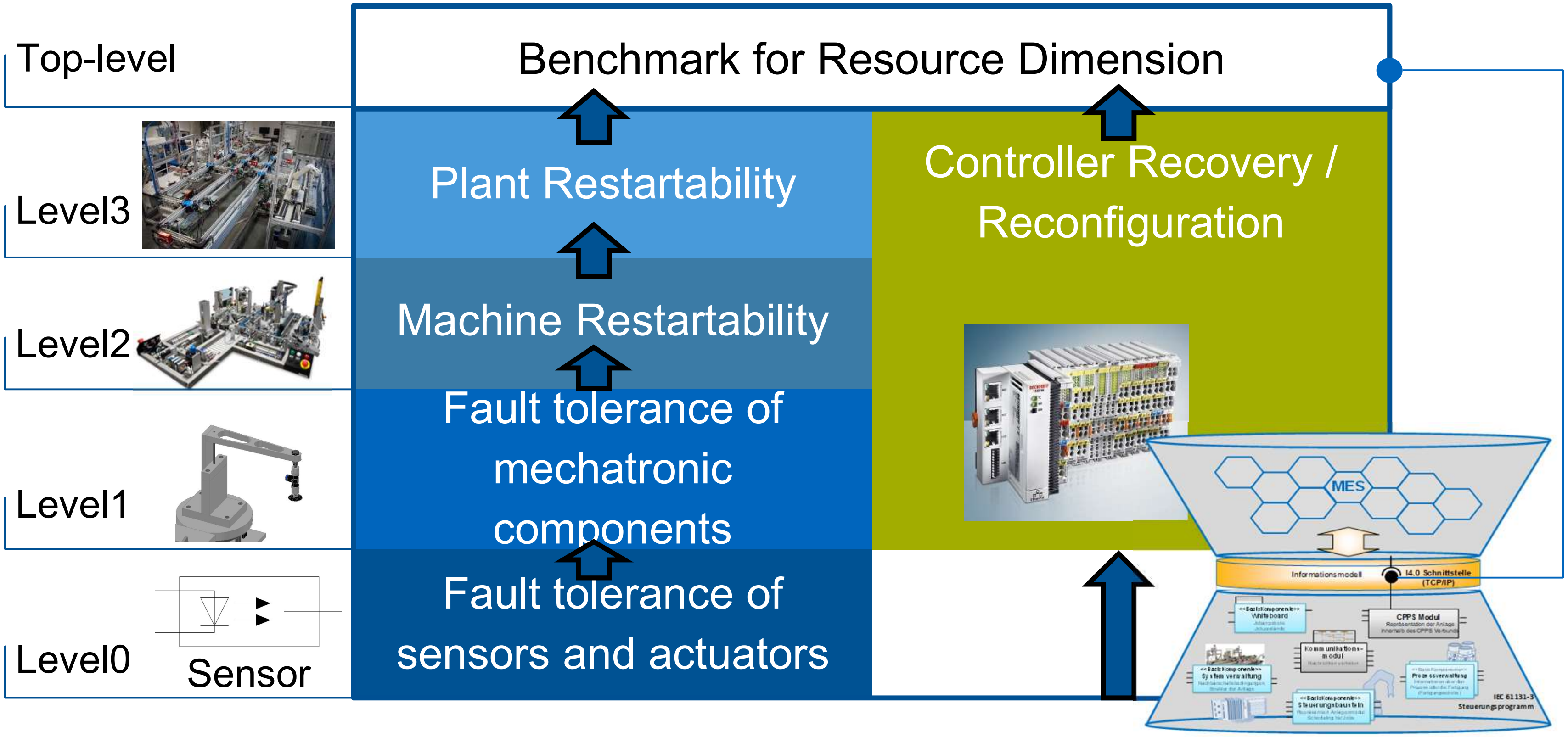
Increased Overall Equipment Effectiveness (OEE)



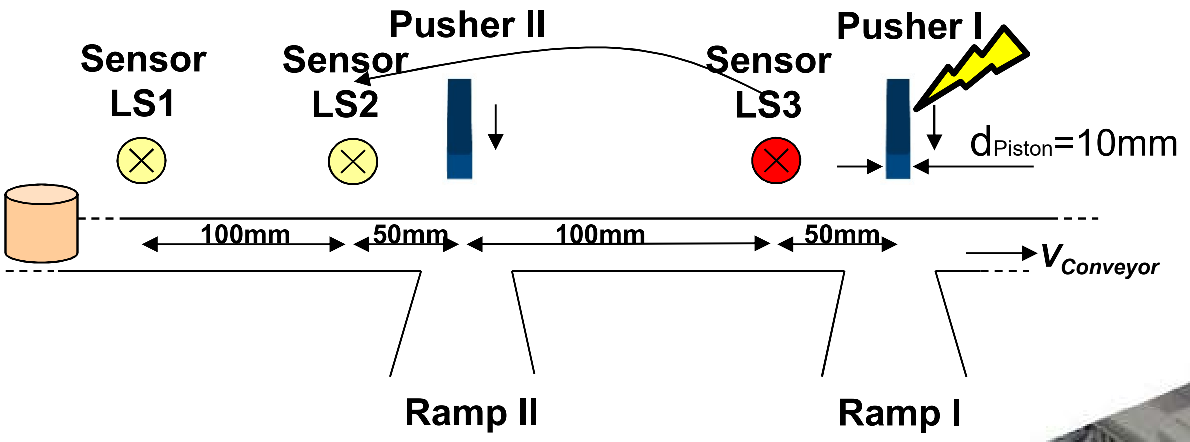
Restartability and Recovery to increase OEE

Restart(ability): to **resume execution after a failure**, using status and results recorded at a checkpoint. (*source: IEEE glossary*)

Recovery: includes **error correction and restart**:
- **correction** is the process of removing the original problem (the fault) and correct its manifestation (the error),
- **restart** is the process of moving the system to a normal state. (*source: Andersson et al. 2010*)



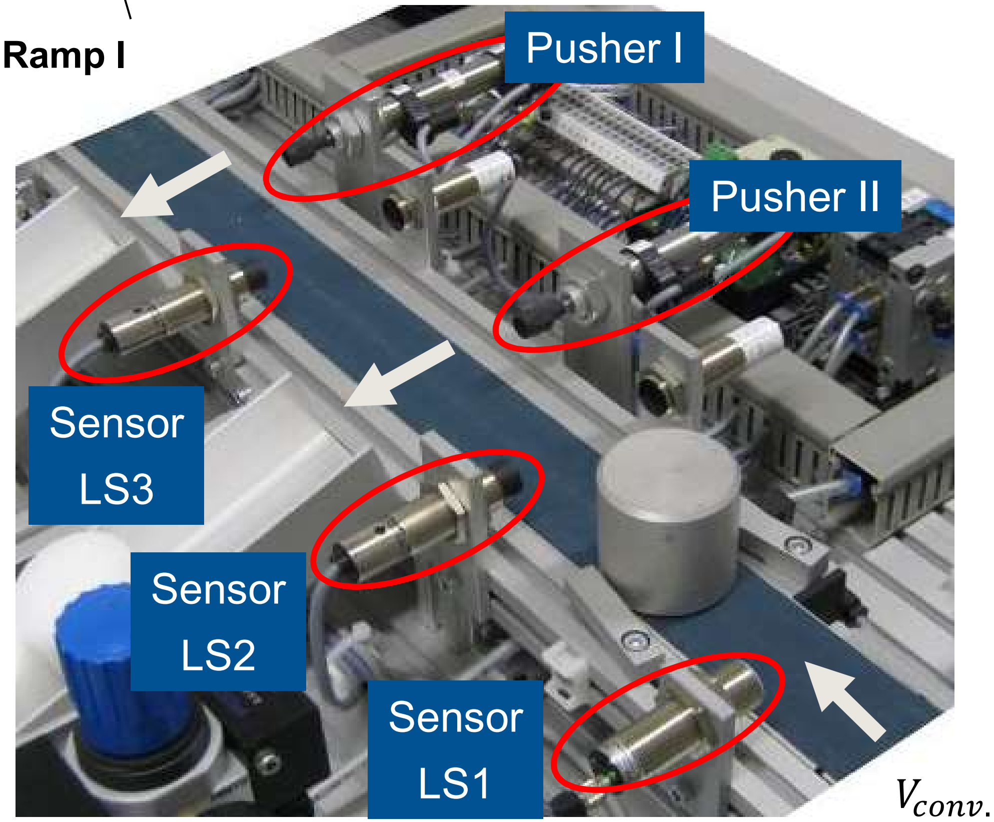
Adaptable Factory (AF) in case of fault



- Sensor LS3 is faulty → no value
- Pusher I: when to push?

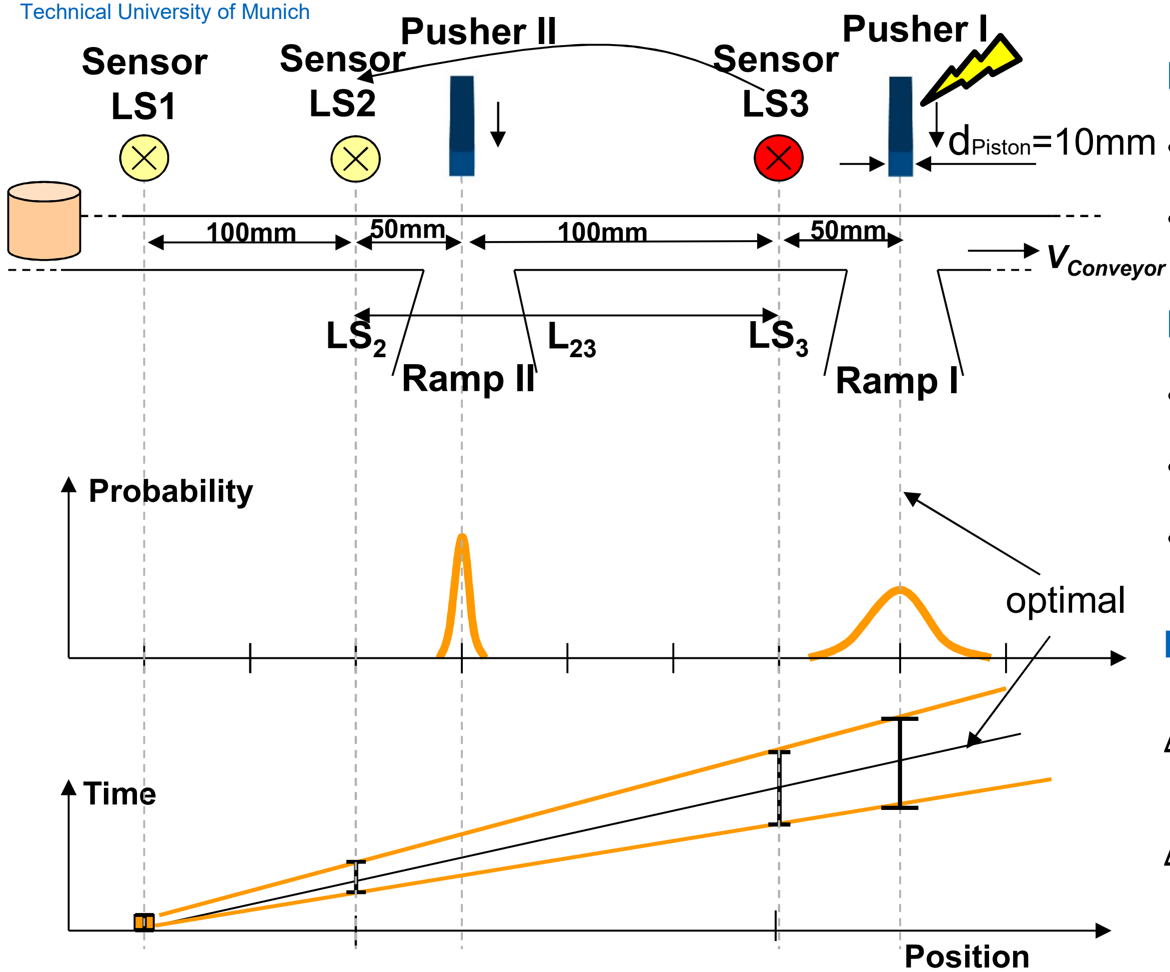
Fault Handling

- Keep processing even without sensor LS3
- Calculating the point of time, Pusher I has to push
- Calculation via time and speed



Automation
and Information Systems
Technical University of Munich

Problem: Tolerance range for virtual sensor



Online operating

Can ejection be guaranteed safely?

Current fluctuation influences speed

Solution: Move conveyor slowly in the piston area

Failure:

- LS3 → a weak or impermissible signal
- Pusher I → no signal to active push

Error handling:

- Error detection and agent remediation
- Virtual substitute value calculation
- Replacement value compensates

Error or deviation

Fluctuations in speed

$$\Delta L_{Route} = L_{23} \cdot \frac{\Delta v_{conv.}}{v_{conv.}}$$

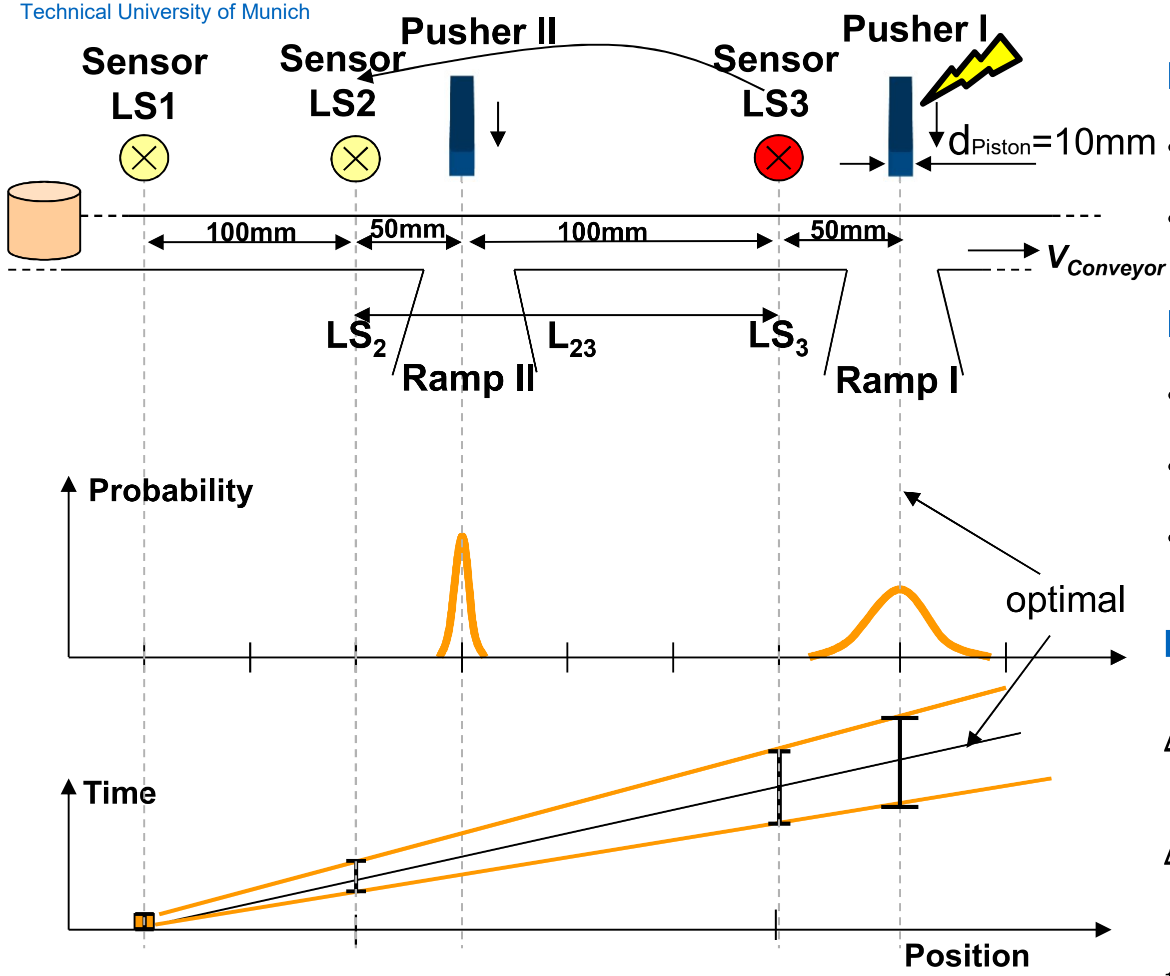
$$\Delta L_{all} = \Delta L_{Sens.} + \Delta L_{React} + \Delta L_{Route}$$

$$t - t_0 = \frac{L_{23}}{v_{conv.}} \rightarrow ready\ to\ Push$$

$$t_0 = time \ \rightarrow LS\ detects\ workpiece$$

$$t = timer\ starting\ with\ conveyor$$

Problem: Tolerance range for virtual sensor



$\Delta L_{Sens.} = Error\ created\ by\ sensing$

$\Delta L_{React} = Error\ created\ by\ actuating$

$\Delta L_{Route} = Error\ created\ by\ fluctuation\ in\ speed$

Failure:

- LS3 →a weak or impermissible signal
- Pusher I →no signal to active push

Error handling:

- Error detection and agent remediation
- Virtual substitute value calculation
- Replacement value compensates

Error or deviation

Fluctuations in speed

$$\Delta L_{Route} = L_{23.} \cdot \frac{\Delta v_{conv.}}{v_{conv.}}$$

$$\Delta L_{all} = \Delta L_{Sens.} + \Delta L_{React} + \Delta L_{Route}$$

$$t - t_0 = \frac{L_{23}}{v_{conv.}} \rightarrow ready\ to\ Push$$

$t_0 = time\ \rightarrow LS\ detects\ workpiece$

$t = timer\ starting\ with\ conveyor$

Additional Constraints with OCL

Define the additional constraints to avoid system malfunction:

- Due to the uncertainties of the calculation → only the previous sensor is considered
- What if the previous sensor is faulty / can only provide a weak or impermissible signal

  →No correct and reliable calculation for the current sensor value
- **To annotate this additional Constraint, the following term is declared**

Sorting all sensors by position

```
context Conveyor Module
inv: x-> self.LS -> sortedBy(Position)
        x[n-1].Signal < 2 implies x[n].Signal > 2
```

Define Array “x”

If the value of previous sensor (x[n-1]) is defect (signal <2)
→ current sensor (x[n]) has to be intact
→ otherwise the system is not functioning anymore

	LS1	LS2	LS3	$V_{conv.}$
LS1	LS1	vLS2	vLS3	
LS2		LS2	vLS3	
LS3			LS3	
$V_{conv.}$		vLS2	vLS3	$V_{conv.}$

ensor

ensor

Supporting information

Agent's knowledge base modeled as Parametric Diagram

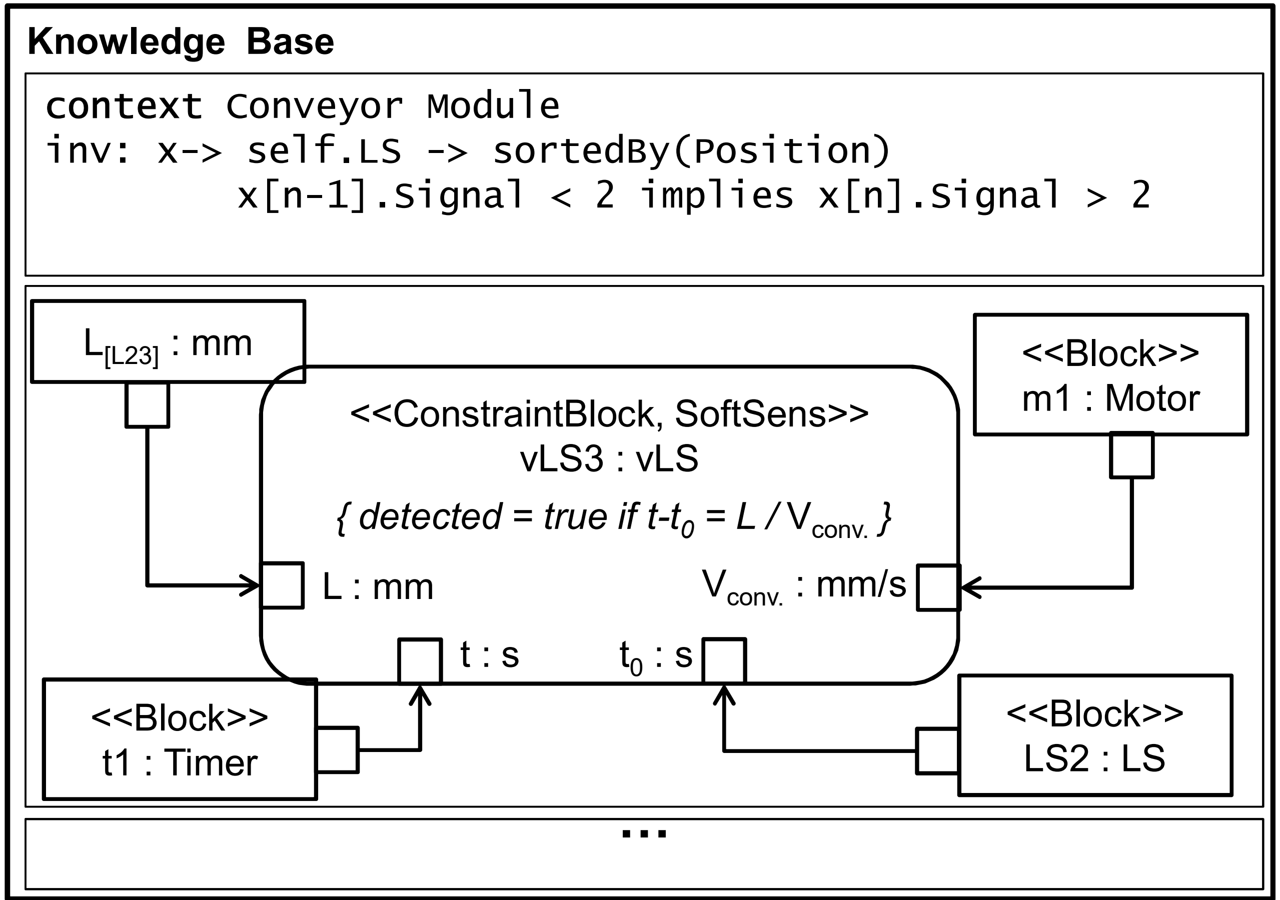
Knowledge base:

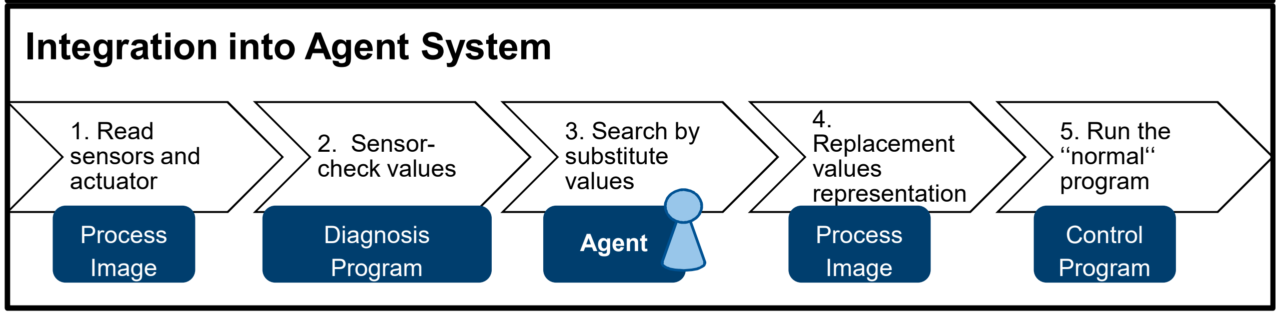
Contains knowledge of the system, and the behavior of the system and the environment, e.g. physical behavior, restrictions, layout, speeds, ...

- **t** is the time starting from the beginning of the conveying process
- $\mathbf{t_0}$ is the time when the Sensor LS2 detects a workpiece
- **L** is the difference between the positions of two sensors (**here Sensor LS2 - Sensor LS1**)
- $\mathbf{V_{conv.}}$ is the velocity of the coveyor

Agent:

- Agent makes decisions based on knowledge base and process image
- Agent controls **Diagnosis Program**





Code generation from Parametric diagram to implement agents' knowledge on PLC for runtime

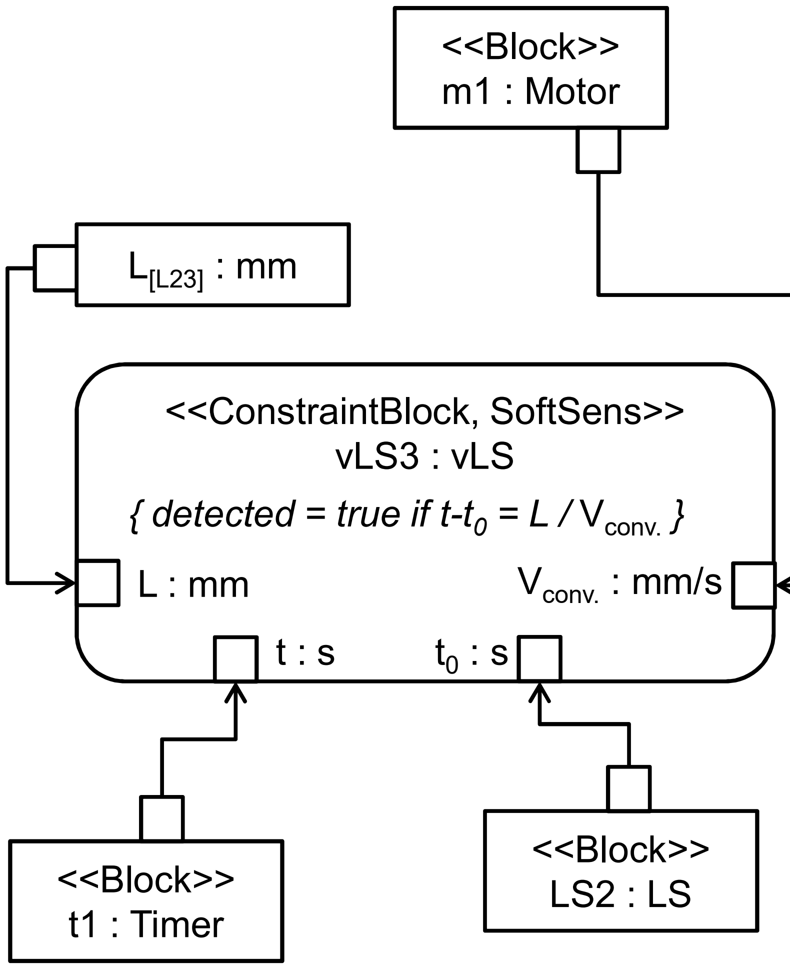


Table of soft sensors

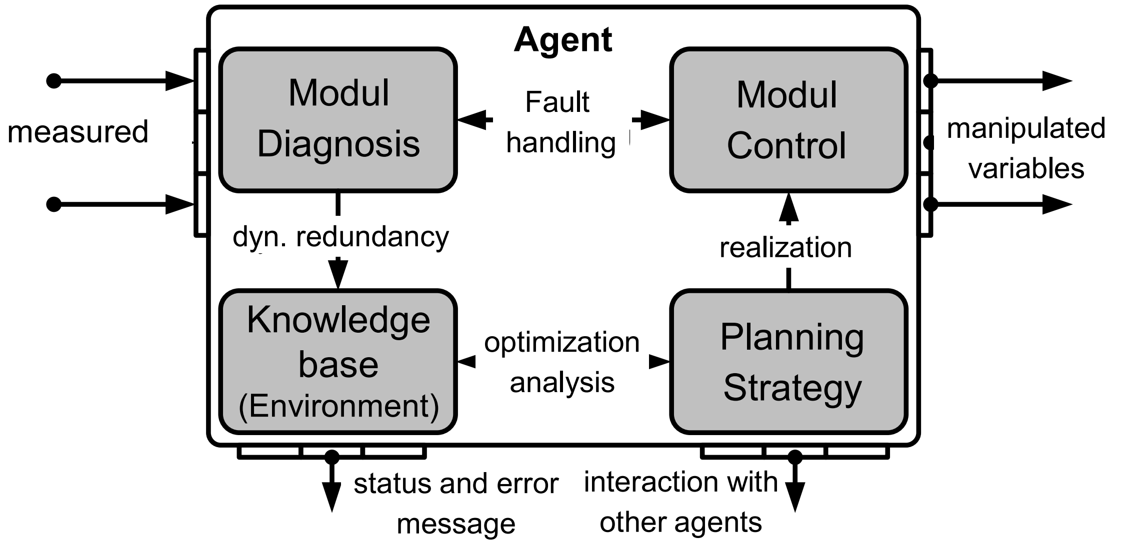
	LS1	LS2	LS3	$V_{conv.}$
LS1	LS1	vLS2	vLS3	
LS2		LS2	vLS3	
LS3			LS3	
$V_{conv.}$		vLS2	vLS3	$V_{conv.}$

Real sensor

Virtual sensor

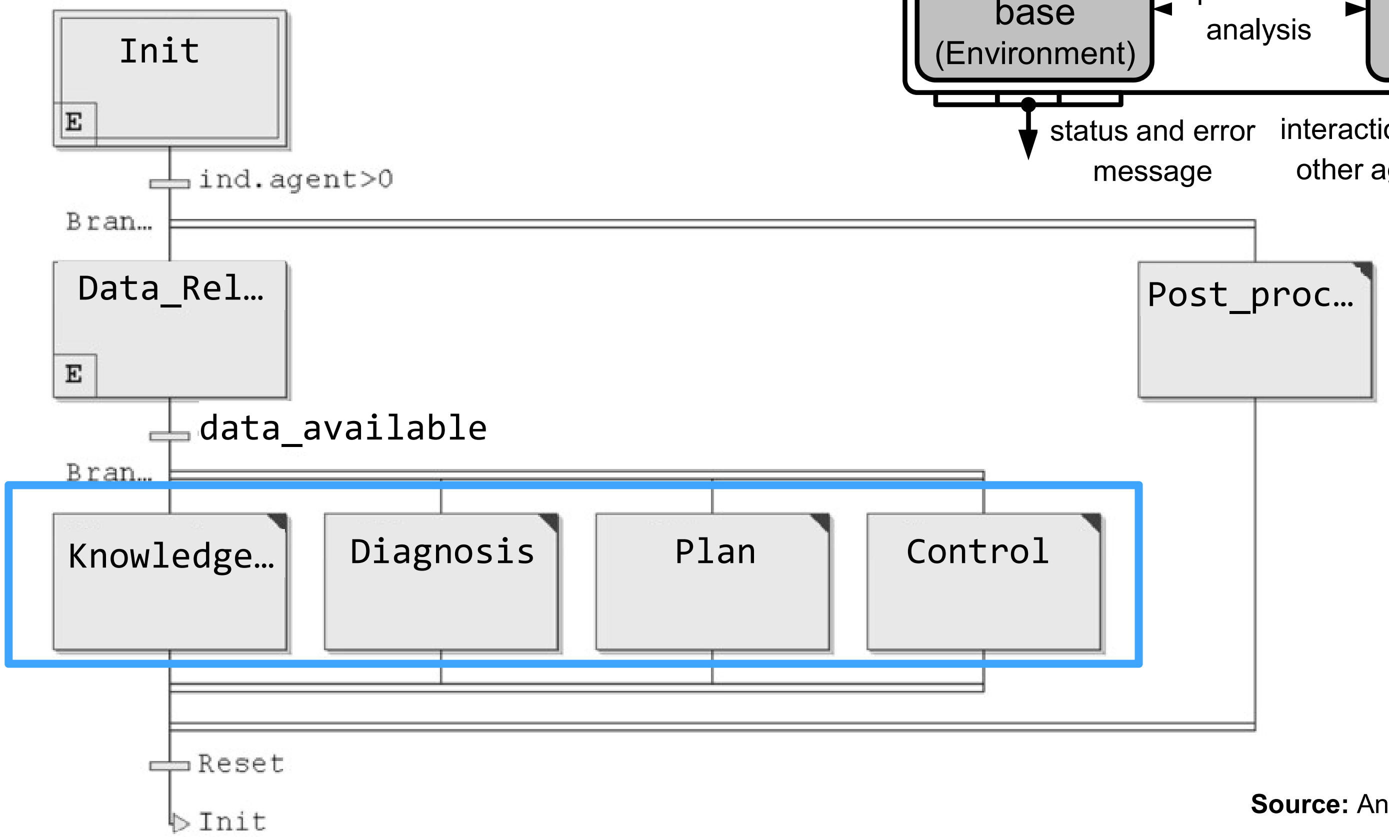
Supporting information

Control Agent in Sequential Function Chart (SFC)



Resource agent

SFC of the Resource agent

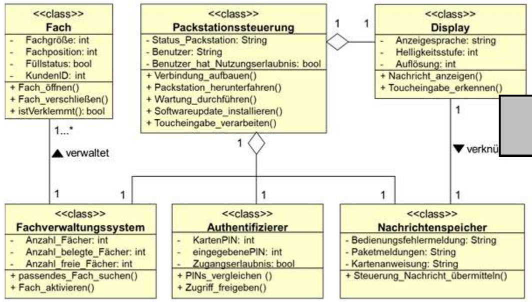


Source: Andreas Wannagat. Dissertation 2010

Model = Code

Structural Model

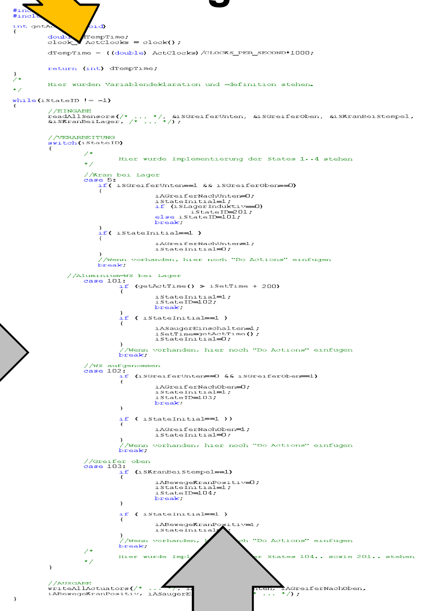
Class-Diagram



Declaration Section

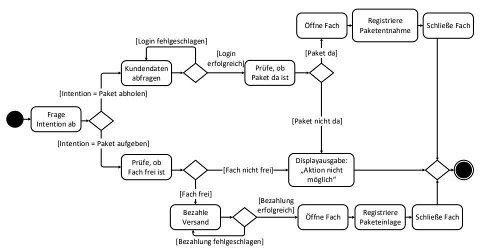
```c
typedef struct Lager_s {
      int WS_vorhanden;
      MATERIAL Material;
      HELLIGKEIT Helligkeit;
      Zylinder Schiebezylinder;
      Sensor Sensor_opt;
      Sensor Sensor_kap;
      Sensor Sensor_ind;
} Lager;

typedef enum {Aluminium, Kunststoff} MATERIAL;
typedef enum {hell, dunkel} HELLIGKEIT;

int Lager_WS_vereinzeln( void ) { /*...*/ }
MATERIAL Lager_WS_analysieren_Material( void ) { /*...*/ }
HELLIGKEIT Lager_WS_analysieren_Helligkeit( void ) { /*...*/ }
void Lager_Notaus( void ) {   /*...*/ }
int Lager_Init( void ) { /*...*/ }
```

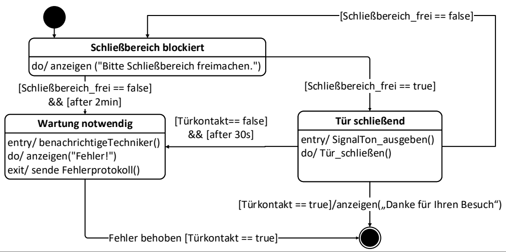
Program



Behavioral Model

Activity- and State Diagramm

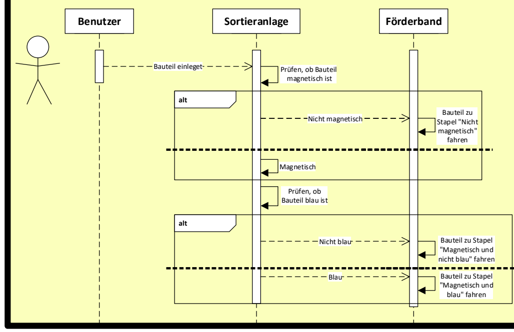




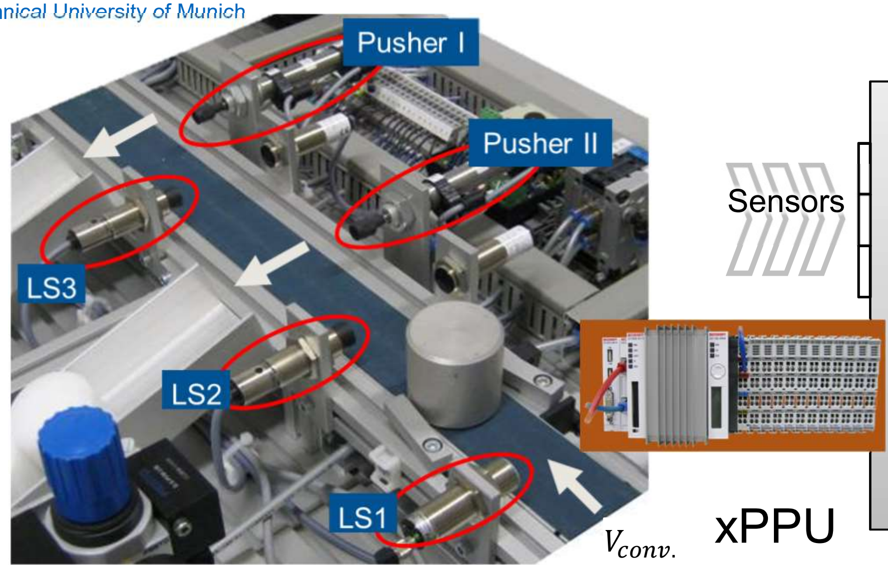
Implementation Section

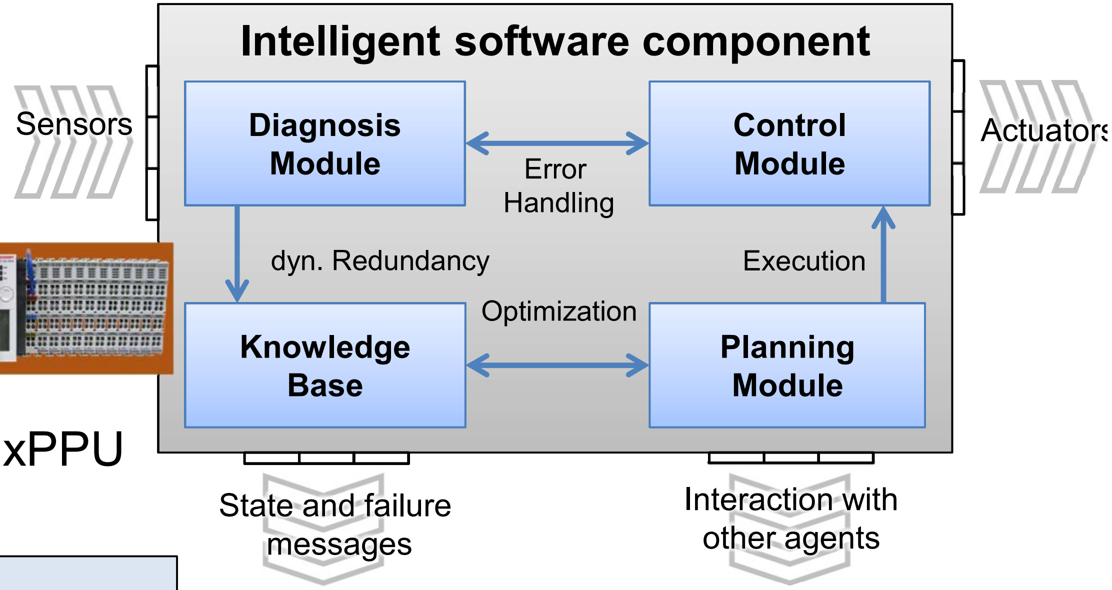
```c
//VERARBEITUNG
switch(iStateID)
{
  /*Hier würde Implementierung der ersten States
   stehen*/
  case 5: //Kran bei Lager
                             //ausgehende Transition
     if( iSGreiferUnten == 1 && iSGreiferOben == 0)
     {
       iAGreiferNachUnten = 0;   //Exit Action
       iStateInitial = 1;        //für nächsten State
       if (iSLagerInduktiv == 0) //State wechseln
          iStateID = 201;
       else iStateID = 101;
       break;                    //Verlassen des States
     }
     if( iStateInitial == 1 )    //bei Betreten
     {
       iAGreiferNachUnten = 1;   //Entry Action
       iStateInitial = 0;
     }
     //Wenn vorhanden, hier noch "Do Actions" einfügen
     break;
  /*Hier würde Implementierung der nächsten States
   stehen*/
}
```

Testing via SD



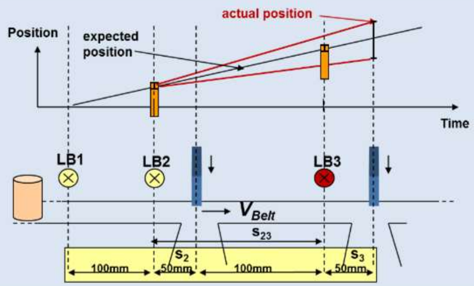
AF- Model-based fault recovery with agent – Level 0





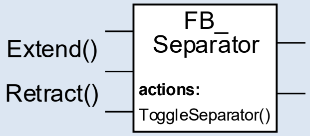
Diagnosis Module

- Evaluation of sensors values
- Execution of failure diagnosis



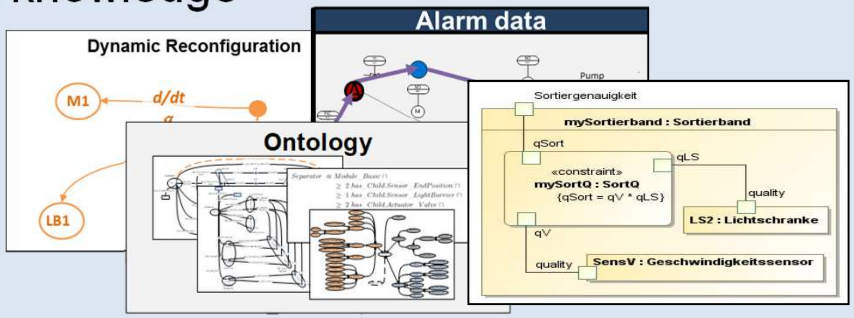
Control Module

Control of the plant module or other sub-agents



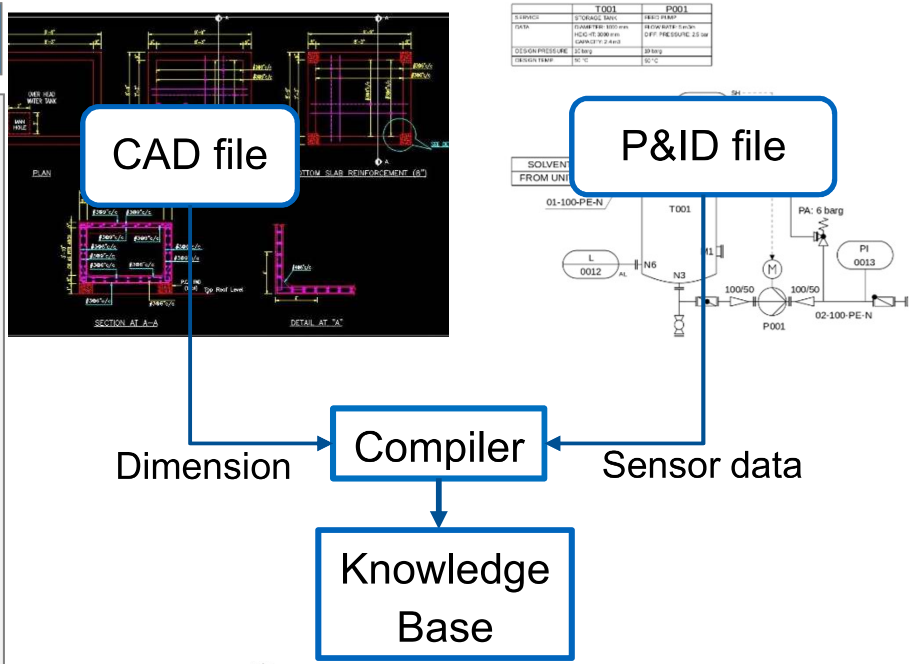
Knowledge Base

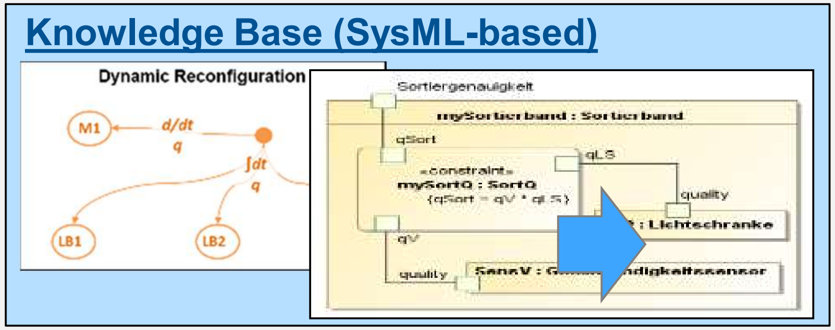
Models of the agents' local knowledge



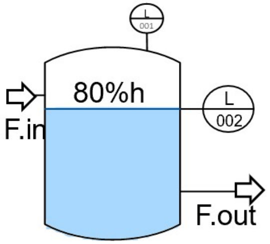
AF- Model-based fault recovery with agent – Level 0

Application in process control





Templates referred to the MOFM2T-standard



$\Delta F = F.in - F.out$

$\Delta V = (1-80\%) \bullet l \bullet w \bullet h$

$\Delta t = \Delta V / \Delta F$

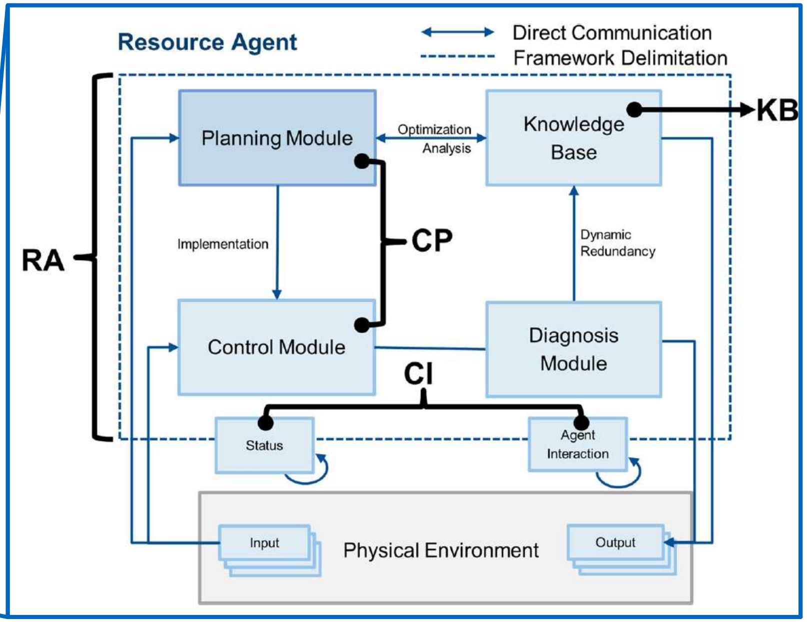
ADR(A1.out)	ADR(A2.out)	ADR(B1.out)	ADR(B2.out)
	ADR(VA2.out)		
		ADR(VB1.out)	
			ADR(VB2.out)

Implementation of redundancy matrix

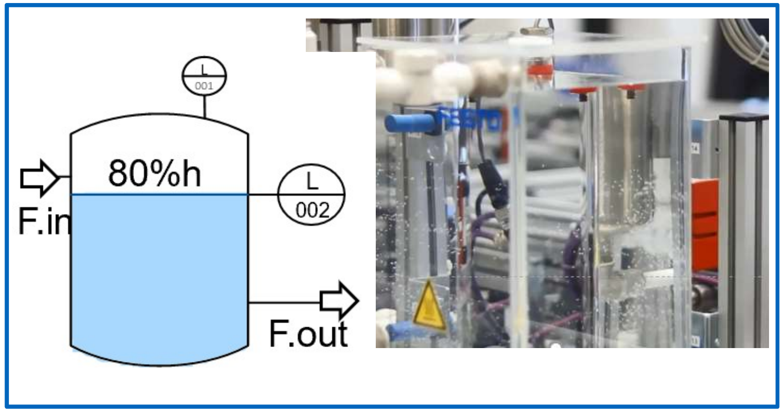
Automation
and Information Systems
Technical University of Munich

Agent design patterns for AF- fault recovery – Level 0
VDI/VDE 2653 part 4 (Draft)

Criteria	Descriptions	Wannagat's Resource agent
Pattern category	Favorable function patterns:, e.g., increased flexibility	Reliability (fault-tolerance)
Pattern type	Name of the pattern type	MAS for faulty sensor or actuator identification in automation systems
Pattern name	Name of the MAS pattern	Agent@PLC
Pattern description	Logic structure (which components does the pattern contain?)	Four (4) main modules are in the Resource agent, addressed by patterns: CI, CP, RA and KB
Context/area of application	Application context of the pattern	aPS
MAS-architecture	Approach for realization of the agents' behavior	Hybrid-pattern replaces faulty valve value with virtual one, MDE calculation
Solution	Graphical depiction of the MAS-Architecture	See Figure
Knowledge base and processing	How is the knowledge stored? Models, rules, etc.	OOP and SysML
Learning /knowledge acquisition	Methods and techniques	Possible, filtering wrong valve' values
Implementation	MAS technologies for realization, e.g. languages	IEC 61131-3
Real-time properties	Timeliness and concurrency requirements	x
Dependability	Requirements towards availability, security	↗
MAS-autonomy	Autonomy/independence in decision making	Half-half dependable–individual agents represent and control plant units



CI: Communication interface; **CP:** Coordination Process; **KB:** Knowledge Base; **RA:** Resource access



Multi-agent systems in industrial automation (Gründruck): Selected patterns for field level control and energy systems, VDI/VDE 2653 Part 4, VDI/VDE, Berlin, Mar. 2021.

How to use
Knowledge and Learning
in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. Fundamentals of field level control
3. AI for field level control
   - **Knowledge** from MDE or **data**
   - Evolution of knowledge
4. Enabling Adaptive Factory and Order Controlled Production using MAS & DT
5. Conclusion and future work

2nd Summer School on AI for Industry 4.0, France, Germany and Switzerland, 2021

Increase Overall Equipment Effectiveness (OEE)

Possible production time

Real production time | Losses due to unplanned shutdowns

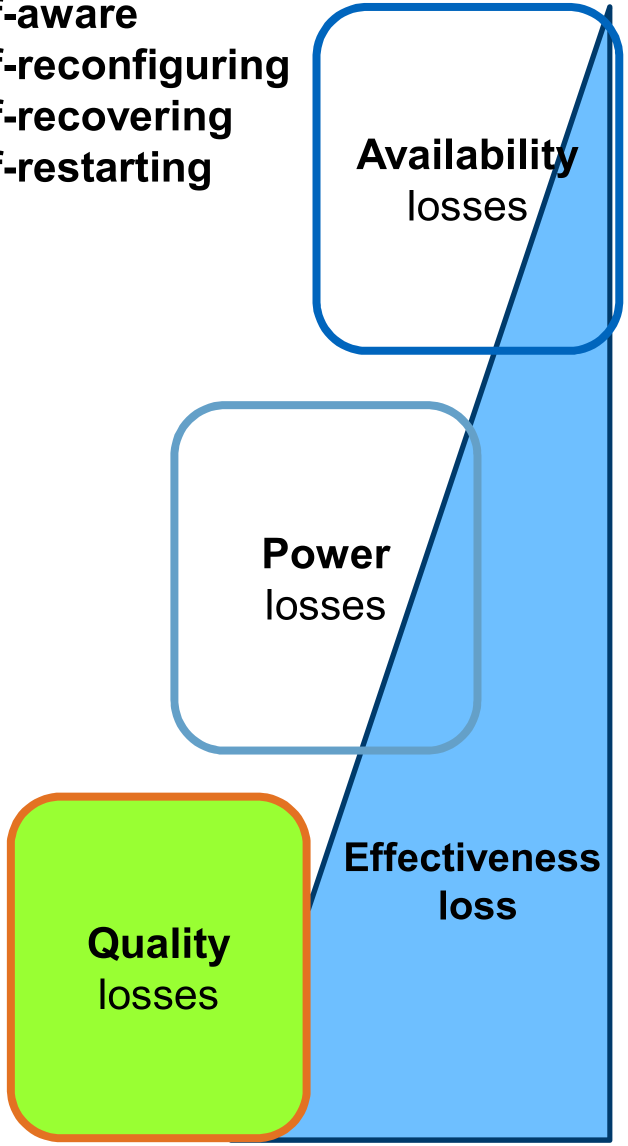
Theoretical output / performance

Real output / performance | Losses due to changing tools, batches...

Possible production / quality

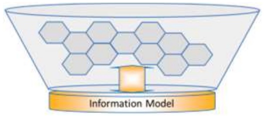
Real production / quality | Losses due to rework, defective goods...

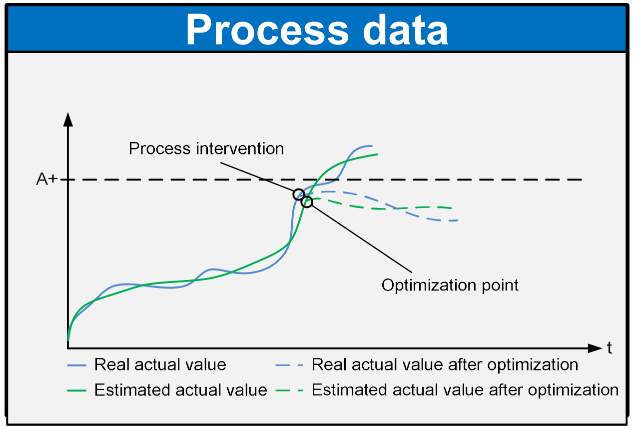
- **self-aware**
- **self-reconfiguring**
- **self-recovering**
- **self-restarting**

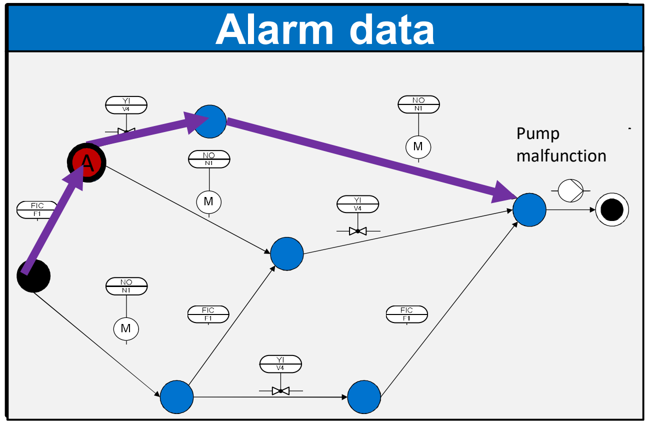


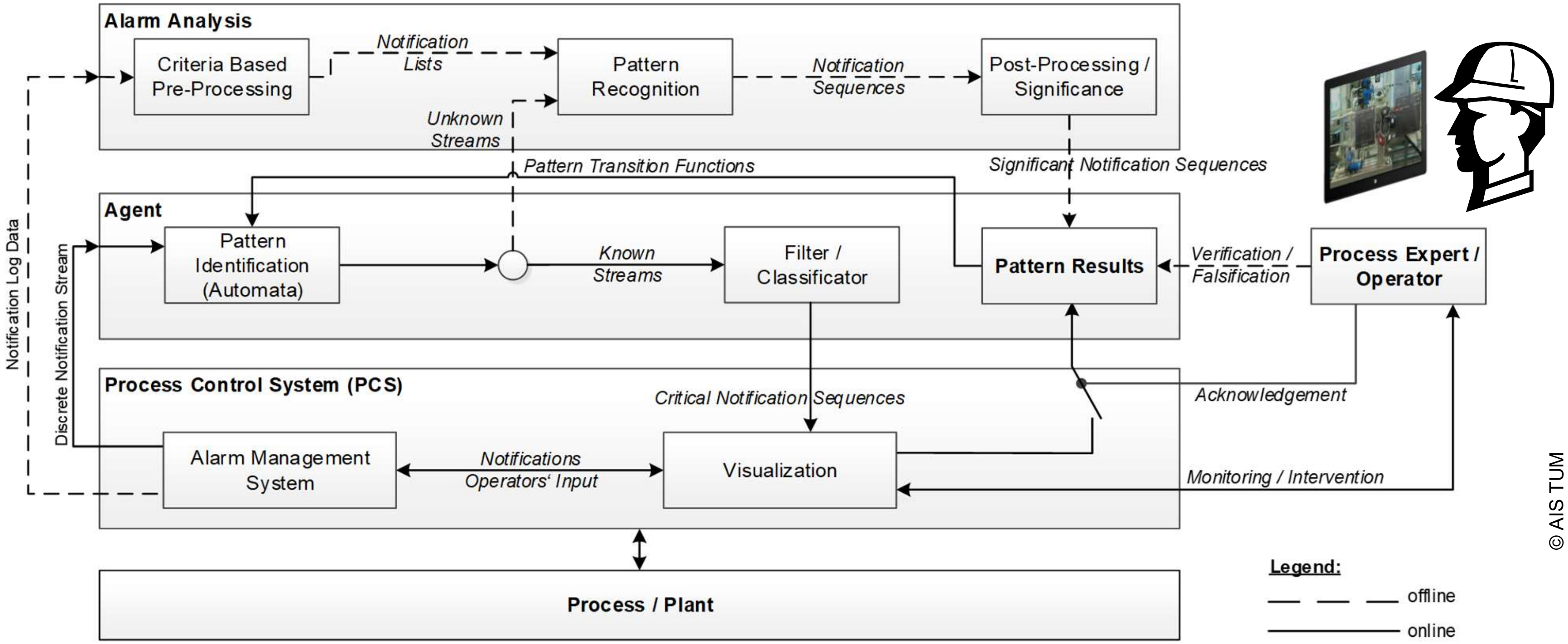
Automation
and Information Systems
Technical University of Munich

Operator Assistance using Agents for Alarm Patterns



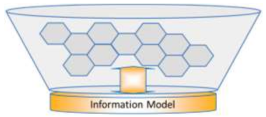


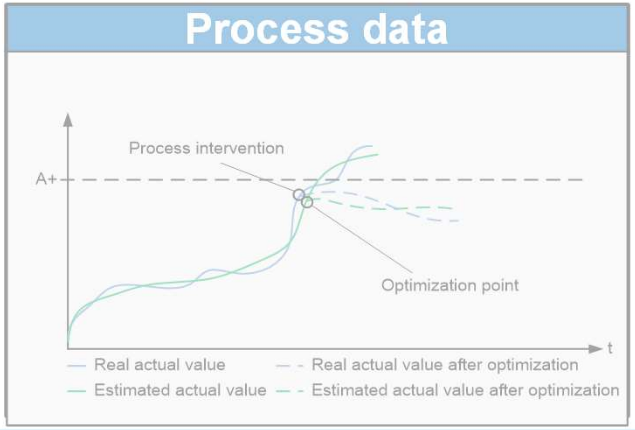


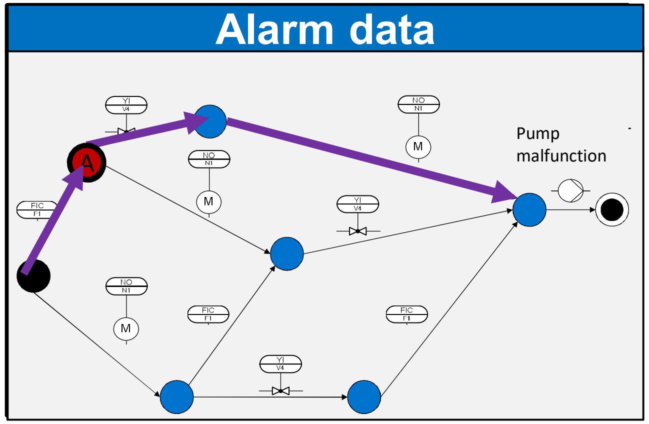


Source: Vogel-Heuser, B. et al.: *Criteria-based Alarm Flood Pattern Recognition using Historical Data from Automated Production Systems (aPS)*. In: Journal Mechatronics, 1-12, 2015. – in press

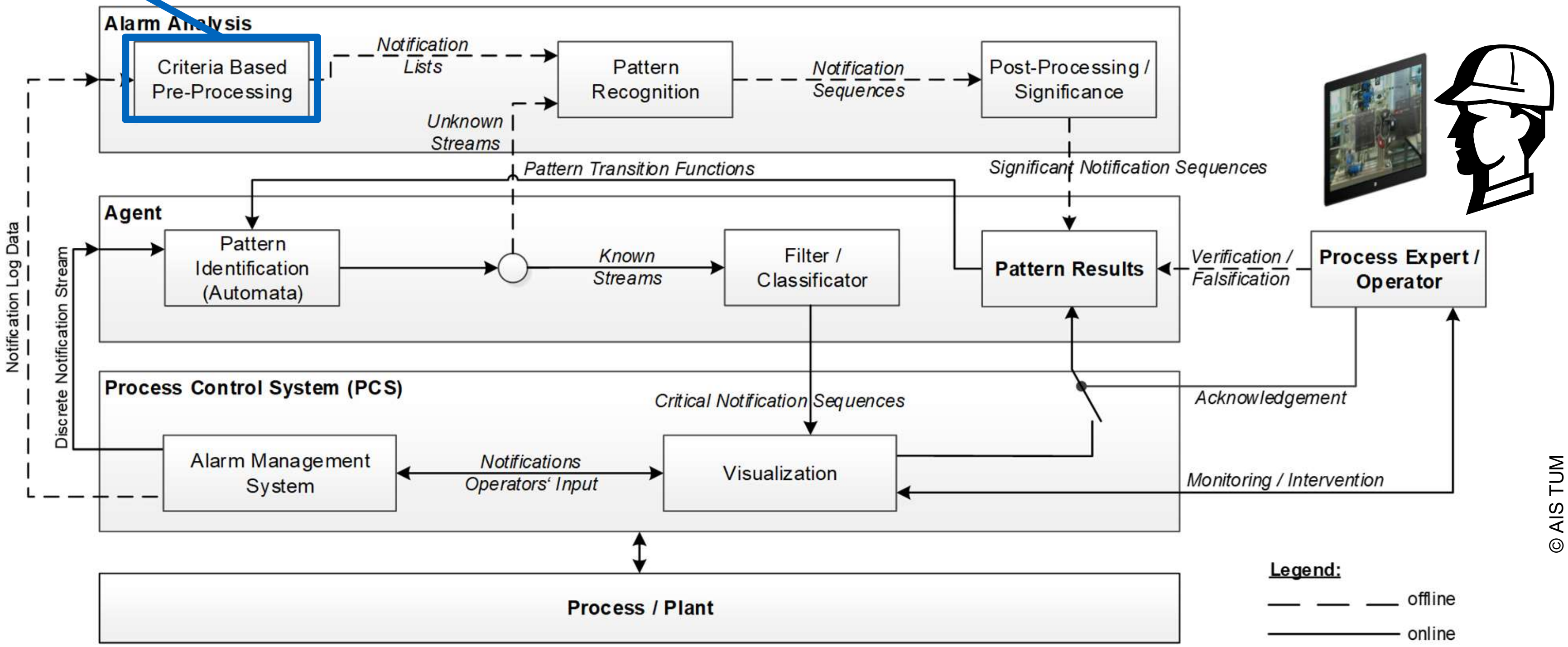
Operator Assistance using Agents for Alarm Patterns







E.g., Similarity analysis with clustering



source: Vogel-Heuser, B. et al.: *Criteria-based Alarm Flood Pattern Recognition using Historical Data from Automated Production Systems (aPS)*. In: Journal Mechatronics, 1-12, 2015. – in press

How to use Knowledge and Learning in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. Fundamentals of field level control
3. AI for field level control
   - Knowledge from MDE or data
   - **Evolution of knowledge**
4. Enabling Adaptive Factory and Order Controlled Production using MAS & DT
5. Conclusion and future work

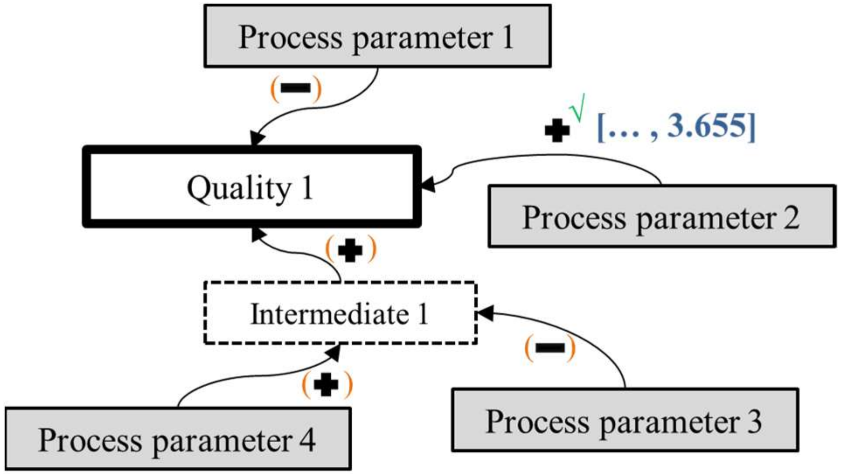
2nd Summer School on AI for Industry 4.0, France, Germany and Switzerland, 2021

Technologists' and Operators' expert knowledge

Expert Knowledge through Interviews

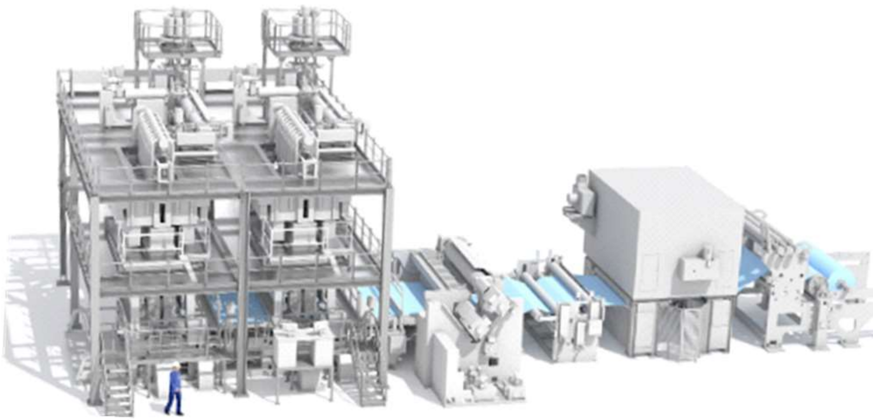
=> Mental Models representing the cause and effect relation of process and quality parameters of aPS

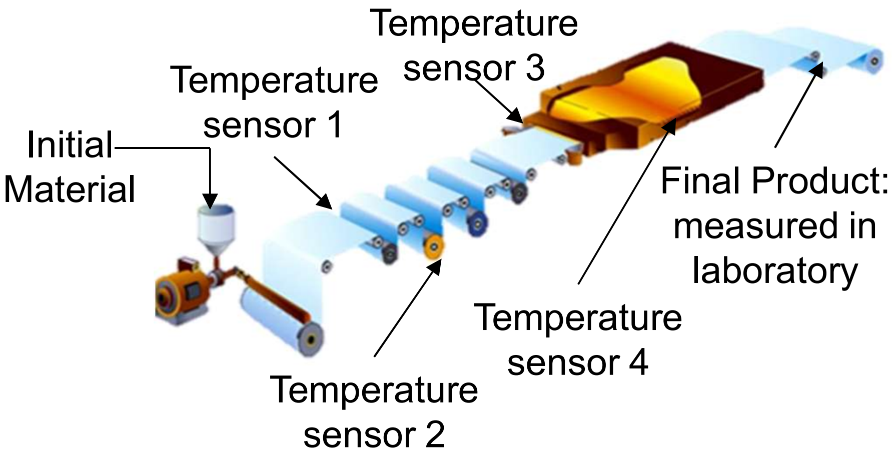
Cause-Effect Graph (Excerpt)



✚ positive relation √ Confirmed by data analysis

▬ negative relation ( ) Not (yet) confirmed by data analysis





Source: B. Vogel-Heuser, V. Karaseva, J. Folmer, I. Kirchen. "Operator Knowledge Inclusion in Data-Mining Approaches for Product Quality Assurance using Cause-Effect Graphs," in 20th IFAC World Congress, July 2017.

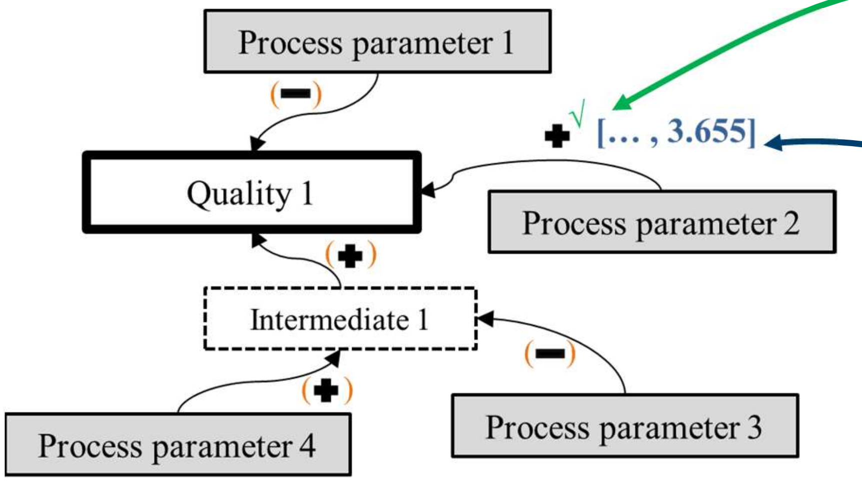
Expert knowledge – mismatch with process data

Agent with embedded expert knowledge ⇔ Agent analysing data (online), using offline design models to predict quality parameter

Expert Knowledge through Interviews

=> Mental Models representing the cause and effect relation of process and quality parameters of aPS

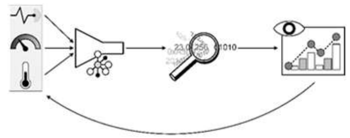
Cause-Effect Graph (Excerpt)



✚ positive relation   √ Confirmed by data analysis

━ negative relation   ( ) Not (yet) confirmed by data analysis

Data Mining



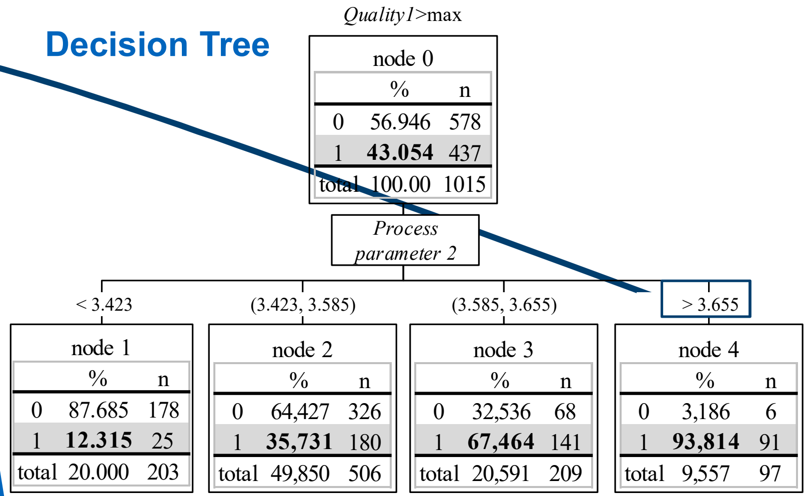
Logistic Regression

Estimator for Quality1>max

	β_j	S.E.	Sig.
absolute term	-44.379	3.225	0.000
Process parameter 1	0.041	0.405	0.920
Process parameter 2	12.285	0.987	0.000
Intermediate 1	0.256	0.350	0.507

S.E. = Standard Error, Sig. = Significance

Decision Tree



Source: B. Vogel-Heuser, V. Karaseva, J. Folmer, I. Kirchen. "Operator Knowledge Inclusion in Data-Mining Approaches for Product Quality Assurance using Cause-Effect Graphs," in 20th IFAC World Congress, July 2017.

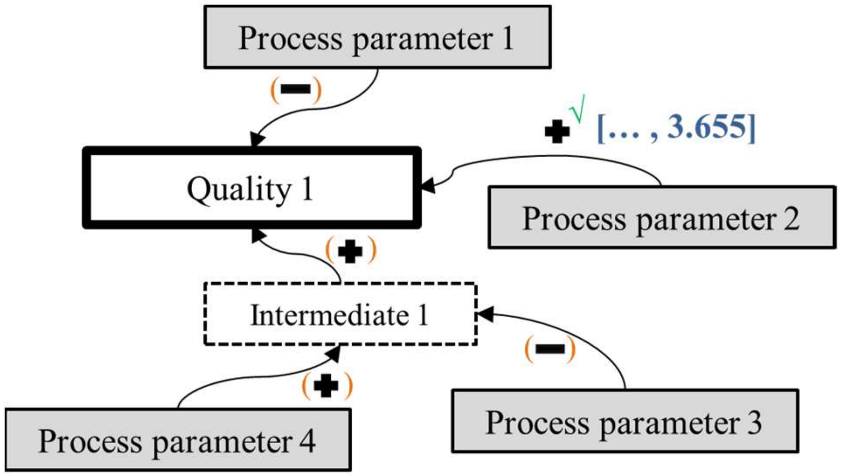
Automation
and Information Systems
Technical University of Munich

Expert knowledge – mismatch with process data

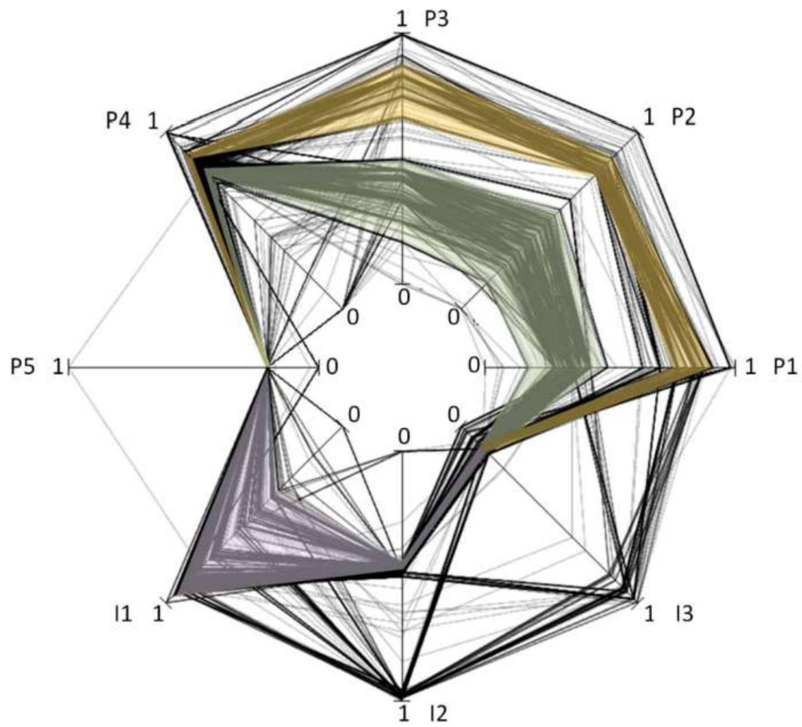
Expert Knowledge Acquisition through Interviews

=> Mental Models representing the cause and effect relation of process and quality parameters of aPS

Cause-Effect Graph (Excerpt)



+	positive relation	√	Confirmed by data analysis
−	negative relation	( )	Not (yet) confirmed by data analysis

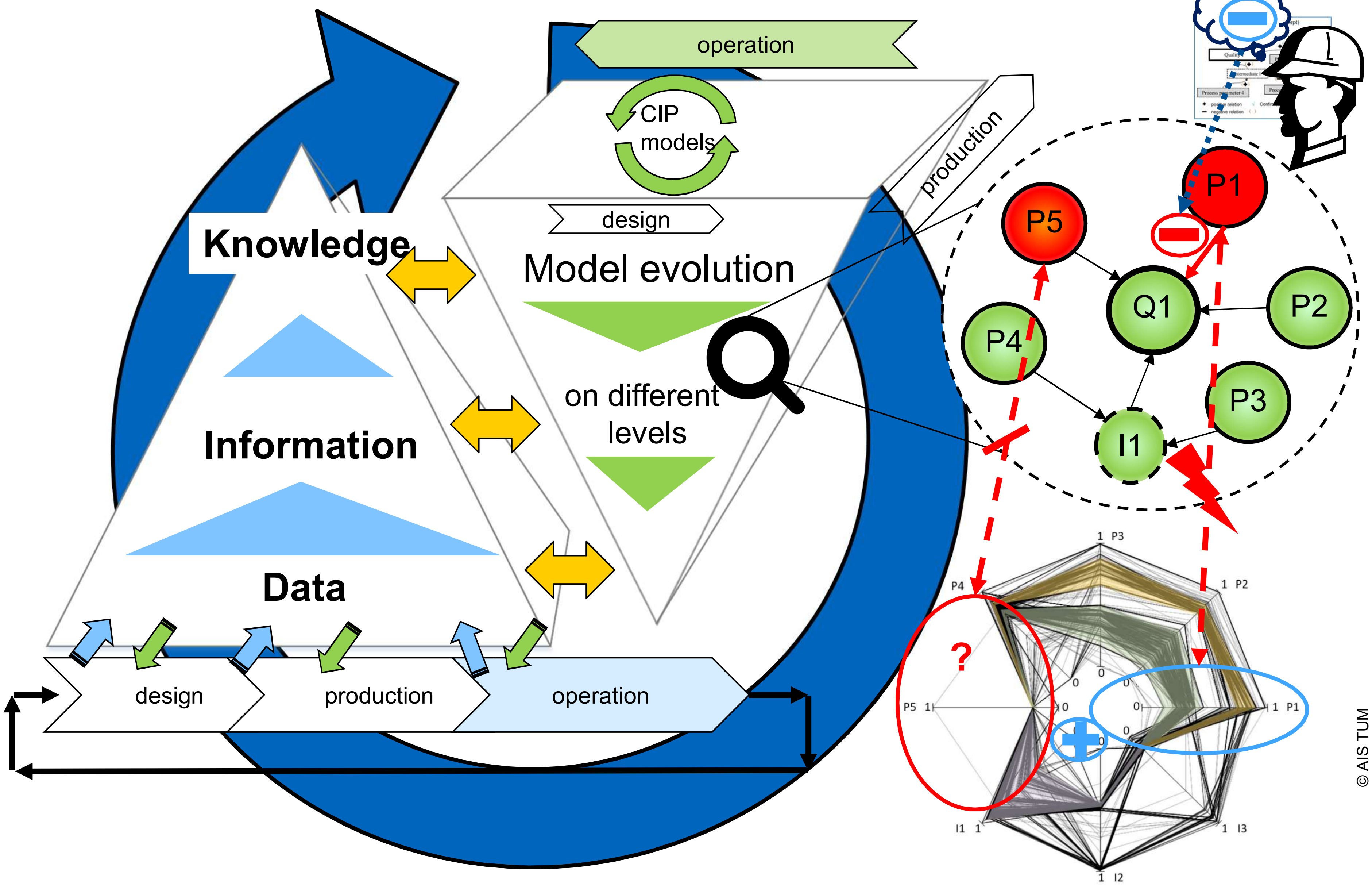


- ➢ Data only show a specific selection of parameter combinations
- ➢ Data analysis cannot show true correlation of parameters

Source: B. Vogel-Heuser, V. Karaseva, J. Folmer, I. Kirchen. "Operator Knowledge Inclusion in Data-Mining Approaches for Product Quality Assurance using Cause-Effect Graphs," in 20th IFAC World Congress, July 2017.

Source: I. Weiß and B. Vogel-Heuser. "Assessment of Variance & Distribution in Data for Effective Use of Statistical Methods for Product Quality Prediction," Automatisierungstechnik (at), vol. 66, no. 4, pp. 344-355, Apr. 2018.

Process data to improve models



How to use Knowledge and Learning in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. Fundamentals of field level control
3. AI for field level control
4. **Enabling Adaptive Factory and Order Controlled Production using MAS & DT**
5. Conclusion and future work

2nd Summer School on AI for Industry 4.0, France, Germany and Switzerland, 2021

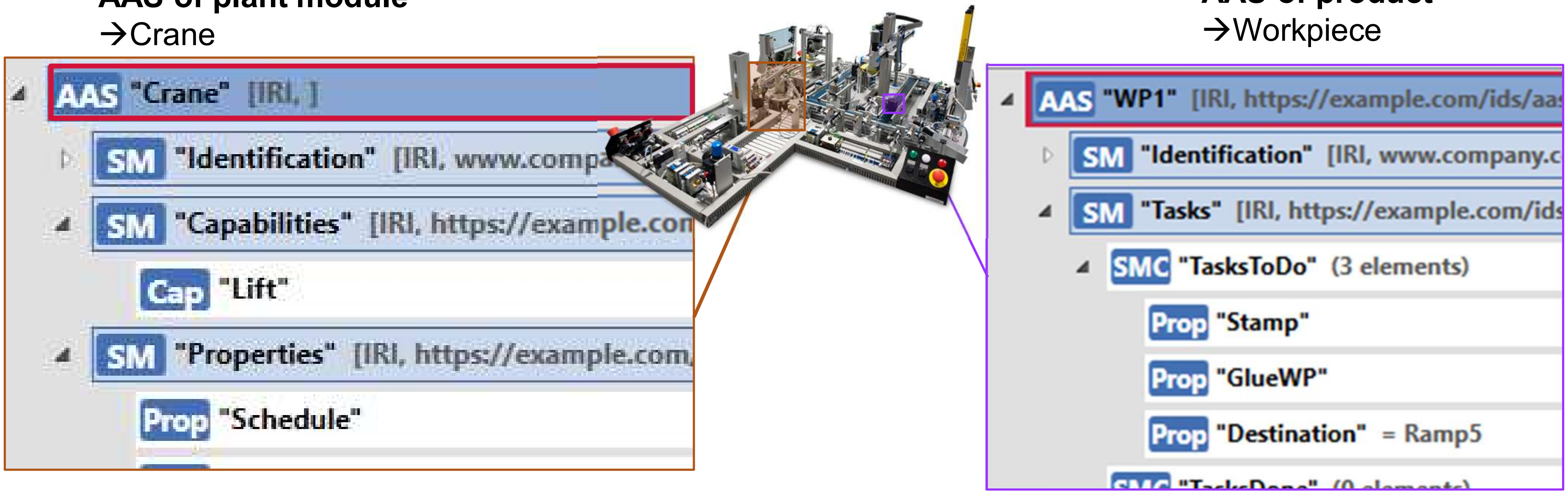
AAS of the xPPU

Agent type	Information	AAS concept	Definition (Plattform I4.0, 2020)
RA	Resource capabilities	Capability	“Implementation-independent potential of an Industrie 4.0 component to achieve an effect within a domain”
	Description of costs, objective function	Submodel, properties	“Models which are technically separated from each other and which are included in the asset administration shell”
	RA parameters, variables, status information	Submodel template	“Specification of the common features of an object in sufficient detail that such object can be instantiated using it”
PA	Product features	Submodel, properties	*cp. above*
	Mapping product features onto production processes	Capability	*cp. above*
	Static parameters; variables; production status	Submodel template	*cp. above*

AAS of plant module
→Crane

Reference to a module

AAS of product
→Workpiece



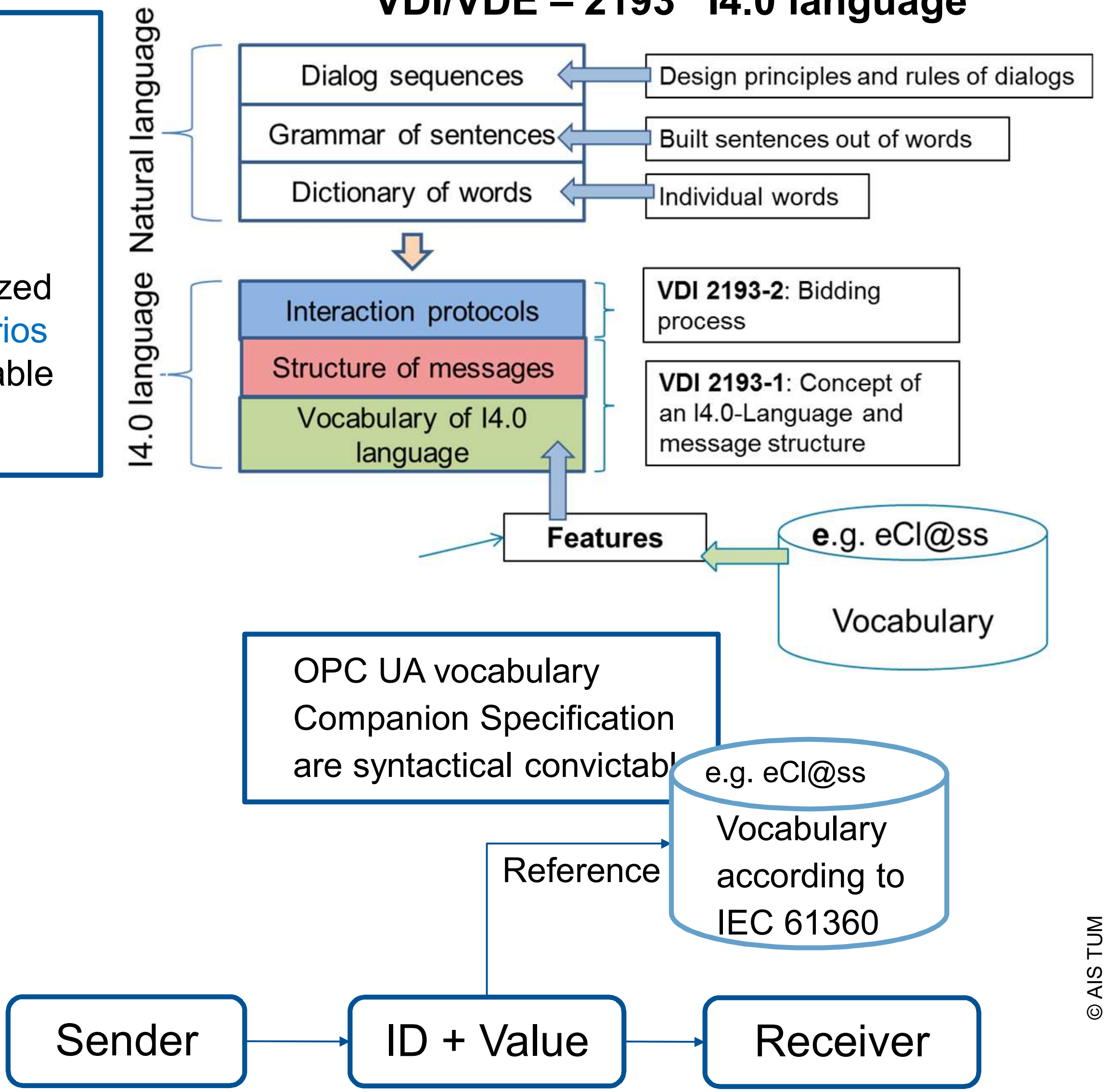
Source: F. Ocker, C. Urban, B. Vogel-Heuser and C. Diedrich. "Leveraging the Asset Administration Shell for Agent-Based Production Systems," in *17th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, Elsevier, Jun. 2021.

I4.0 scenarios' gap
VDI/VDE 2193-1: Vocabulary of the I4.0 language

Gap
Machine executable and semantic unambiguous abstraction of the real world

Solution
Extending the interoperability by standardized I4.0 language among multiples I4.0 scenarios (not only focusing on the traditional Adaptable factories concept)

VDI/VDE – 2193 "I4.0 language"



Nominal Voltage 240V

Attribute	Value
ID	0173-1#02-BAB576#005
Version	V9.1
Name	Spannung
Description	-
Symbol	U
SI unit	V
Data type	real
Value	240
Value range	0..240

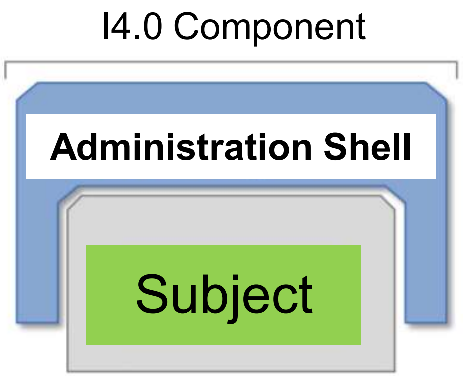
ID: ISO 29002-5
URL
IEC 61360

LNI, Competence Center 4.0 Hannover, Hannover Messe, ifak, AAS networked 06.04.2020

Basic System and Digital Twin for Industry 4.0

***Management shell (Asset Administration Shell or AAS)**

- Administration shell + physical object → I4.0 component
- provides interface for I4.0 communication
- Addresses: access protection, visibility, identity and rights management, confidentiality and integrity

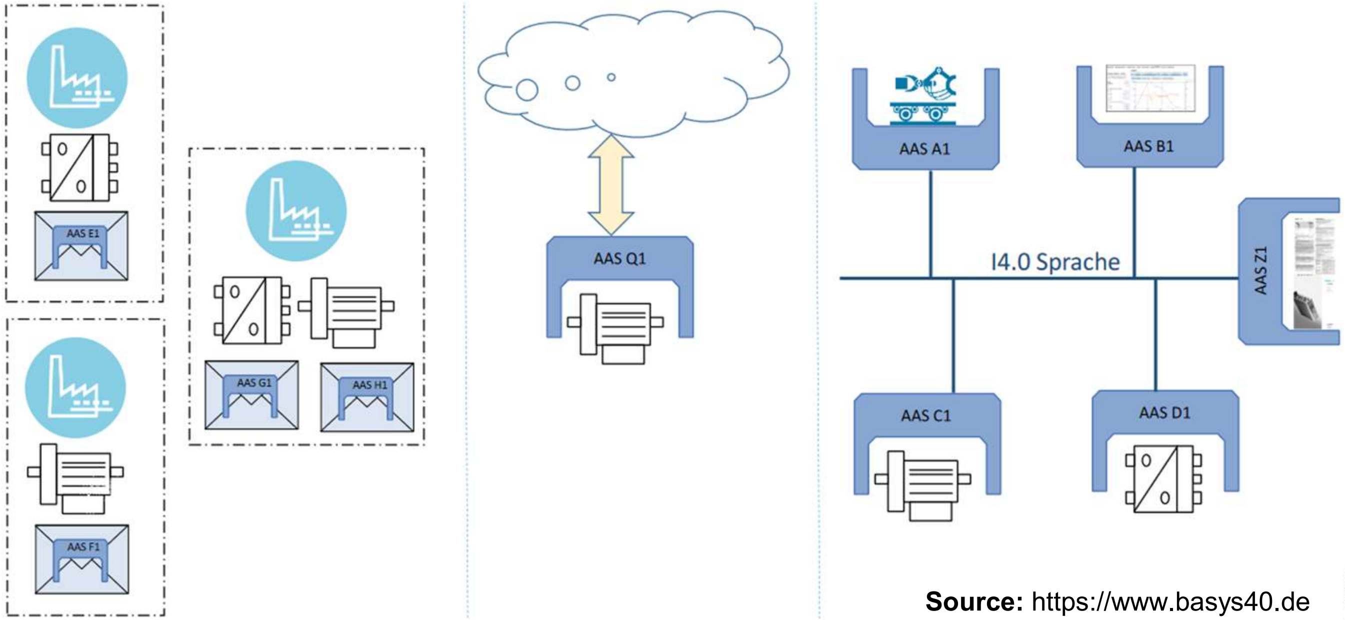


Source: Industrie 4.0 Plattform

Passive AAS (standardized format)

Re-active AAS (accessible via interface)

Pro-active AAS (I4.0 language)



Source: https://www.basys40.de

DotAAS-Details of the Asset Administration Shell - Part 1 The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01)

***Source:** Platform Industrie 4.0 (2019). Glossary. https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/Glossary/glossary.html

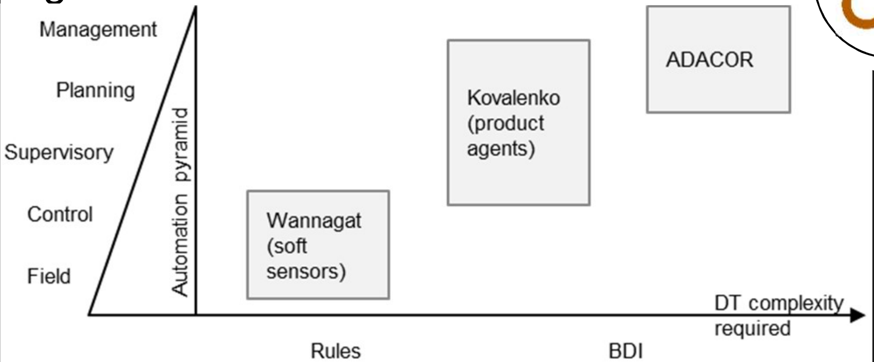
Combining Agents and Digital Twins

Multi-Agent Systems

Agent types (Cruz Salazar et al., 2019)

- **Resource Agent**
- **Product Agent**
- Communication Agent
- Agent Management System

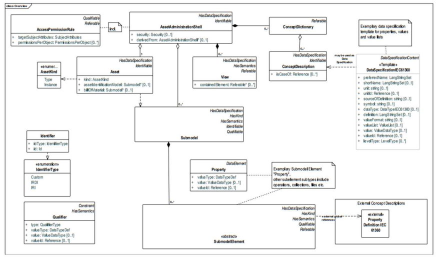
Agent architectures



Digital Twins (DTs)

Asset Administration Shell (Plattform I4.0, 2021)

- **Passive**: exchange format
- **Reactive**: provide a client/ server interface in terms of an Application Programming Interface
- **Proactive**: decision-making skills that ensure the goal-oriented behavior of I4.0 components in the sense of autonomous systems

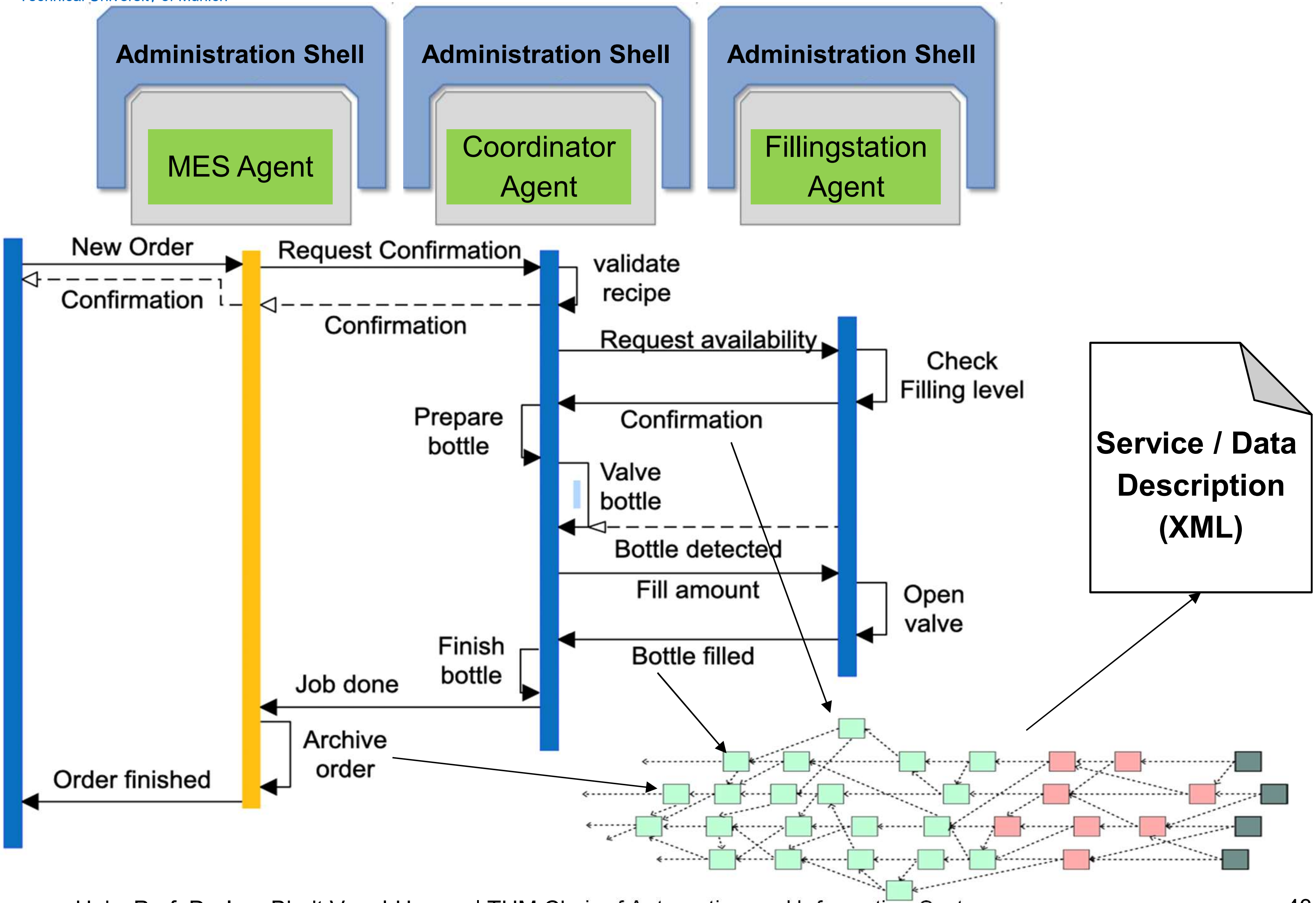


Communication:
I4.0 language (Plattform I4.0, 2018) defines message structure, interaction protocols, and content (VDI 2193) – similar to FIPA

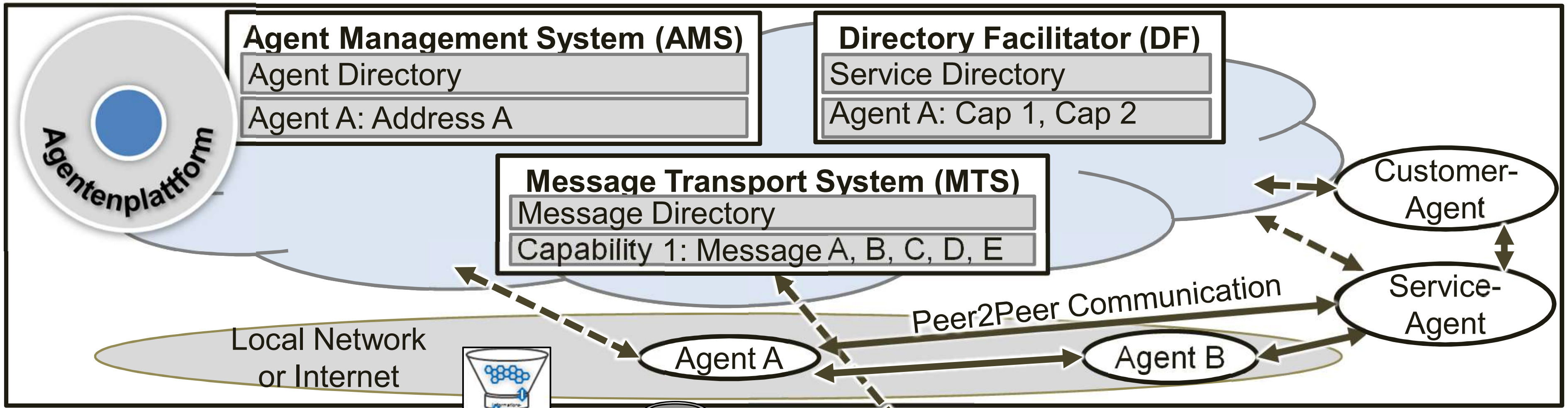
1 – Engineering	2 - Deployment	3 – Operation	4 - Evolution & Maintenance
Include information relevant for agents in the DTs	Create links to support communication between • Agents and DTs • Agents and control level • Agents and agents	Autonomous operation of the production system	Feed changes from the shop floor back into the DT

Source: F. Ocker, C. Urban, B. Vogel-Heuser and C. Diedrich. "Leveraging the Asset Administration Shell for Agent-Based Production Systems," *in 17th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, Elsevier, Jun. 2021.

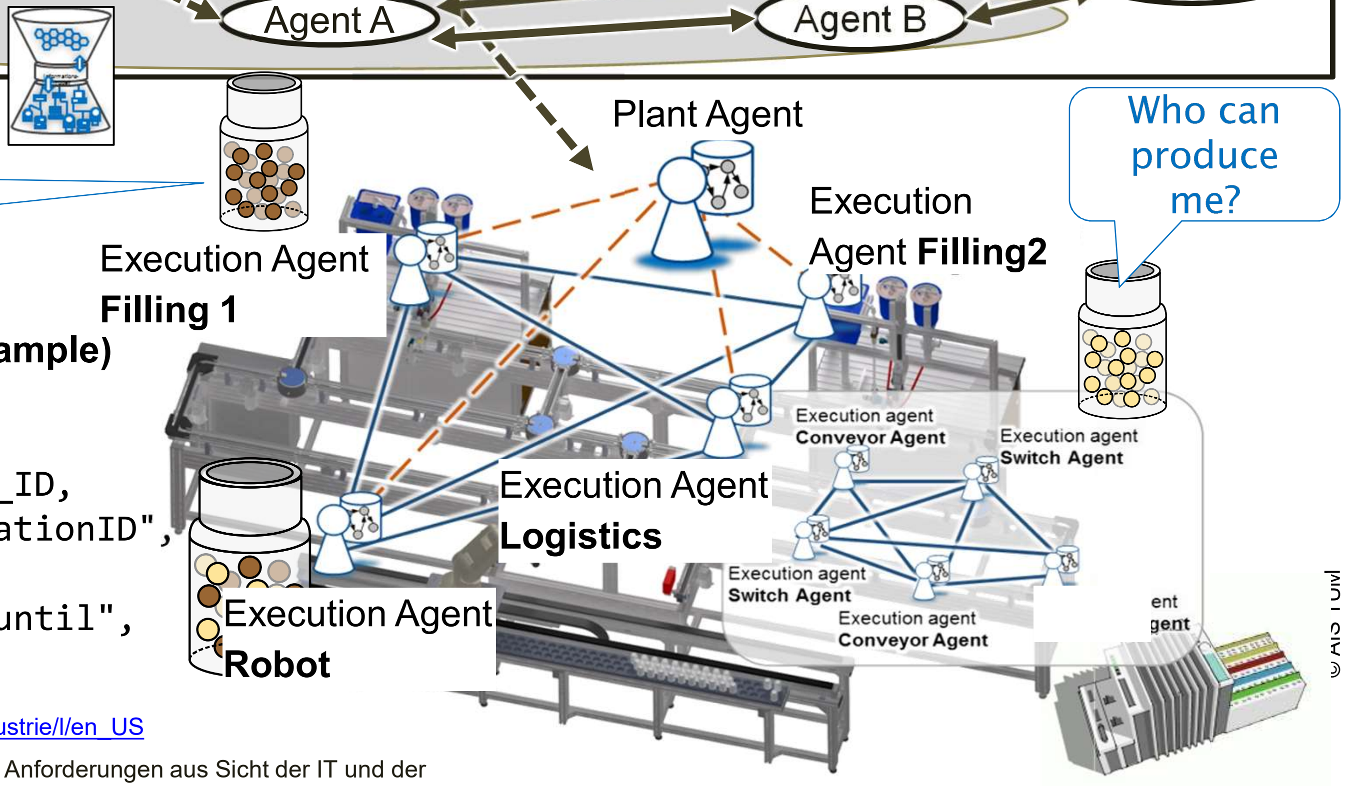
I4.0 language messages as IOTA transactions



I4.0 architecture for OCP



Who can transport me?



Who can produce me?

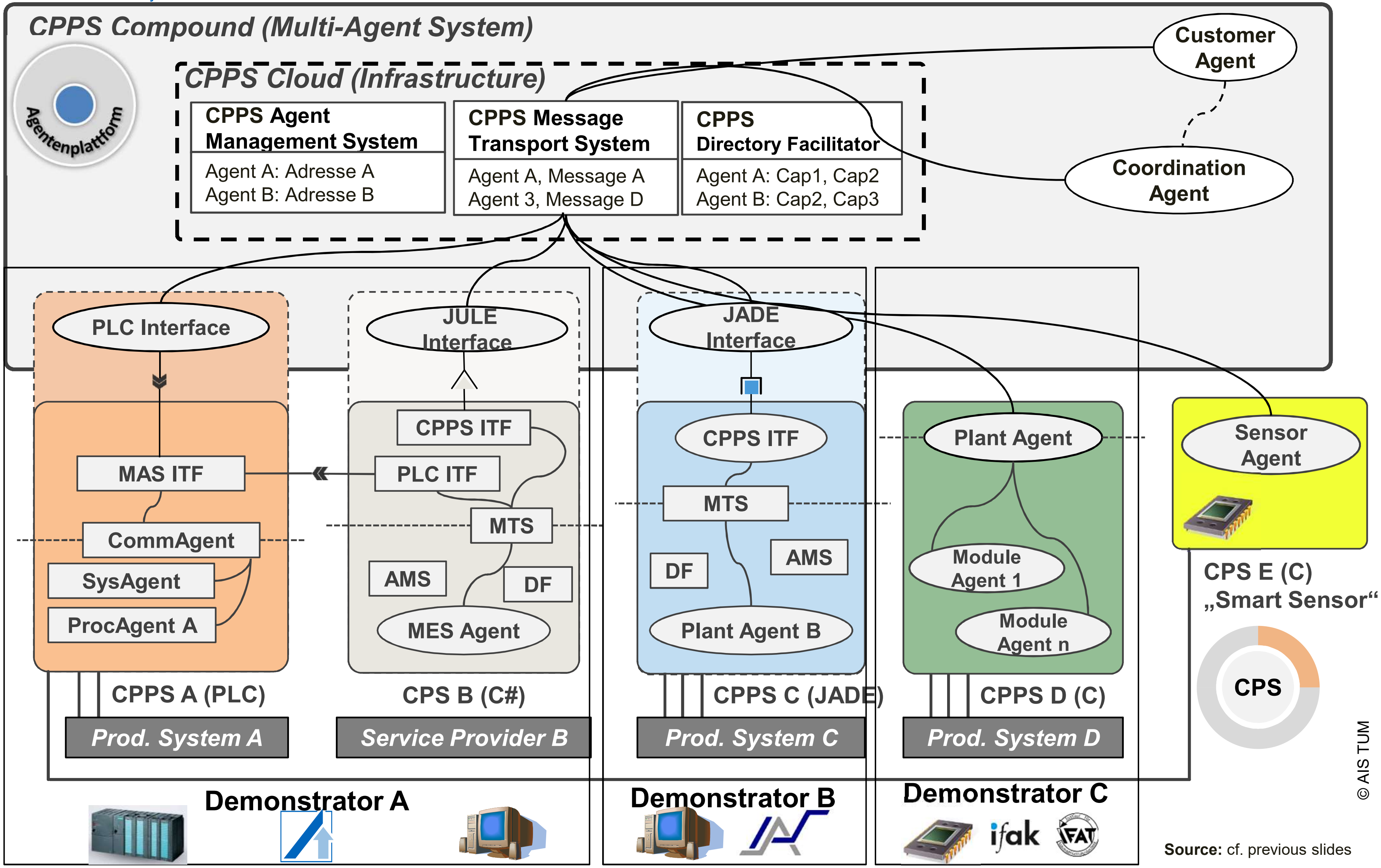
Message in JSON-Format (example)

"{"Sender":"Sender_ID",
"Destination":Destination_ID,
„ConversationNo":„ConversationID",
"MessageNo":"Message_ID",
"ResponseDueTo":"Answers-until",
"Purpose":"**CfP**„
...

Starterkit I4.0: http://i40d.ais.mw.tum.de/index/industrie/l/en_US

Source: B. Vogel-Heuser: Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik. In: Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer, 2014.

Agent-based CPPS architecture



Source: cf. previous slides

How to use Knowledge and Learning in Multi Agent Systems for dependable Field Level Control to realize Industry 4.0

1. Introduction & motivation
2. Fundamentals of field level control
3. AI for field level control
4. Enabling Adaptive Factory and Order Controlled Production using MAS & DT
5. **Conclusion and future work**

2nd Summer School on AI for Industry 4.0, France, Germany and Switzerland, 2021

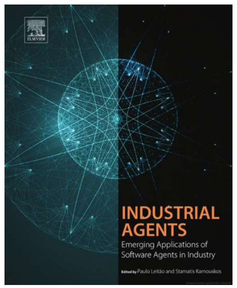
Automation
and Information Systems
Technical University of Munich

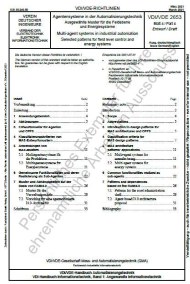
VDI/VDE-GMA Technical Committee 5.15 “Multi-Agent-Systems”

Goal: development of **agent-based applications** in automation technology & development of **novel approaches** and **methods** of agent systems

Current Research Foci:

Agent systems and learning	*Agents in the Supply Chain/ Agents in Decision Network*
• Learning agents, AI and (distributed) learning methods • (Selective) information exchange between agents (transfer of intermediate knowledge) & semantic service discovery techniques • Autonomy & AI	• Agents in **production, intralogistics & Supply Chain** (JIT and JIS) • Architecture to combine the concepts of agents, service-oriented architectures and semantics • Autonomy for the decision maker • Other application areas





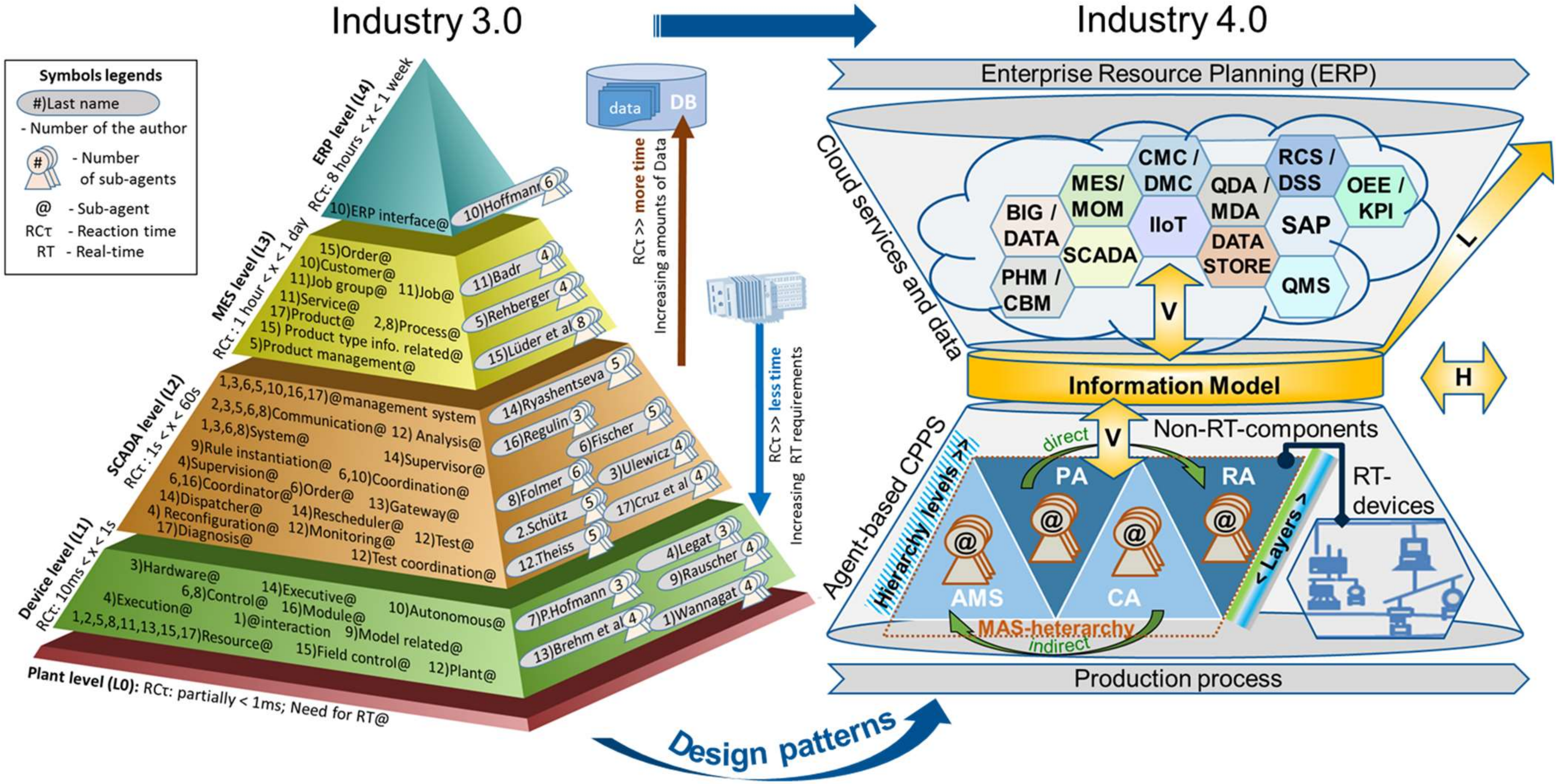
TUM
Helmut Schmidt Universität
Hochschule Augsburg University of Applied Sciences
Universität Stuttgart Germany
Otto von Guericke Universität Magdeburg
RWTH Aachen University

German Universities

Infineon
Honeywell
FESTO
GEFASOFT
PILZ THE SPIRIT OF SAFETY
SIEMENS

Industry partners

Using Design Patterns to migrate to Industry 4.0



Migration from Industry 3.0 to the Industry 4.0. The left side is the automation levels of action for all identified sub-agents. The right side is a proposed “diabolo” architecture, adapted from AIS-TUM. Abbreviations in alphabetic order, are **@**: Sub-agent pattern; **AMS:** Agent Management System; **CA**: Coordination Agent; **CBM**: Condition Based Monitoring; **CMC**: Collaborative Manufacturing Community; **CPPS**: Cyber Physical Production System; **DMC**: Decentralized Manufacturing Community; **DSS**: Decision Support System; **H**: Horizontal integration; **IIoT**: Industrial Internet of Things; **KPI**: Key Performance Indicator; **L**: Life-cycle integration; **MAS**: Multi-Agent System; **MES**: Manufacturing Execution Systems; **MOM:** Manufacturing Operations Management; **OEE**: Overall Equipment Effectiveness; **PA**: Process Agent; **PHM**: Prognostics and Health Management; **QMS:** Quality Management System; **RA**: Resource Agent; **RCS**: Resilient Control System; **RT**: Real-time; **SAP**: Systems Applications Products; **SCADA**: Supervisory Control and Data Acquisition; and **V**: Vertical integration.

Source: L. A. Cruz S., D. Ryashentseva, A. Lüder, and B. Vogel-Heuser, “Cyber-physical production systems architecture based on multi-agent’s design pattern—comparison of selected approaches mapping four agent patterns,” *Int. J. Adv. Manuf. Technol.*, vol. 105, no. 9, pp. 4005–4034, Jul. 2019.

Thank you for your attention!
We welcome international exchange students!

Birgit Vogel-Heuser, Ordinaria
- Lehrstuhl Automatisierung und Informationssysteme (AIS)
  Fakultät Maschinenwesen
- Work@ Munich School of Robotic and Machine Learning
- Munich Data Science Institute (MDSI)

Technische Universität München (TUM)
www.ais.mw.tum.de; vogel-heuser@tum.de