Advanced Aspects on Model-driven Engineering and implementation of field level agents

1. **Introduction of TUM and AIS**
2. Field level agents in automation
3. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
4. Use case: Sensor reconfiguration modeled with SysML and code generation from SysML Model to PLC‘s
5. Agents‘ knowledge of valid combinations or sequences modeled with OCL
6. Metrics for adaptivity
7. Conclusion and future work

Univ.-Prof. Dr.-Ing. Birgit Vogel-Heuser

Head of Chair and Director of Institute

Automation and Information Systems (AIS)

Technical University of Munich

www.ais.mw.tum.de/ vogel-heuser@tum.de

Technische Universität München

- **39,081 students**
- **13 faculties**
- **3 Integrative Research Centers**
- **6 Corporate Research Centers**
- **13,248 female students**
- **9,846 staff members**
- **411 buildings**
- **~ €1.1 billion invested in construction since 2001**

Students by Department	Total	No. of female students	No. of international students
Mechanical Engineering	5,216	762	1,247

Technical University of Munich

- The leading university in mechanical and electrical engineering in Germany

Rankings 2015

- Technical University of Munich:
  - 51st at the Academic Ranking of World Universities (Shanghai-Ranking)
  - 60th at the QS World University Ranking
- Faculty of Maschinenwesen:
  - 19th at the QS World University Ranking by Subject (1st in Germany)

Memberships Head of Chair

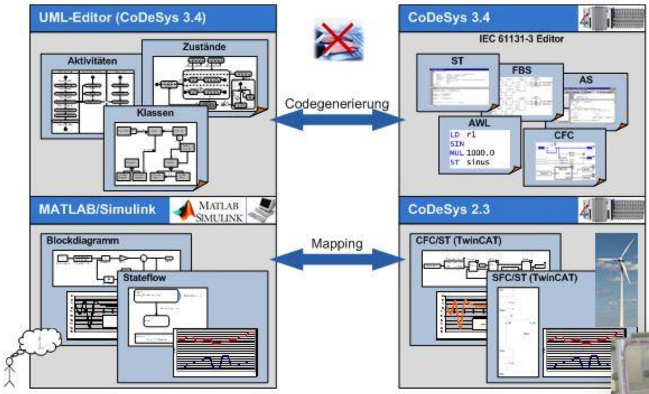
- Chair of VDI/VDE (Association of German Engineers) TC 5.15 “Multi-Agent Systems in Automation”
- Coordinator of CRC (Collaborative Research Center) 768 “Managing cycles in innovation processes”
- Co-Initiator of PP (Priority Programme) 1593 “Design for Future – Managed Software Evolution”

Scientific staff

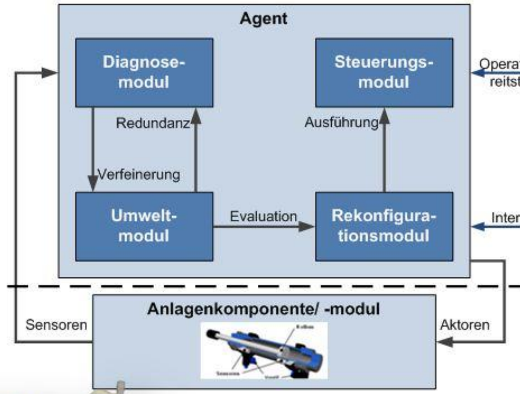
- ca. **20** PhD students
- **9** technicians, trainees (software engineering)

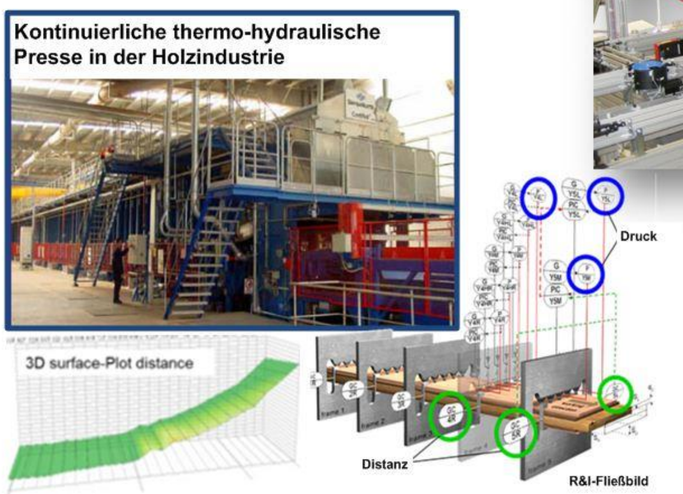
Research Topics

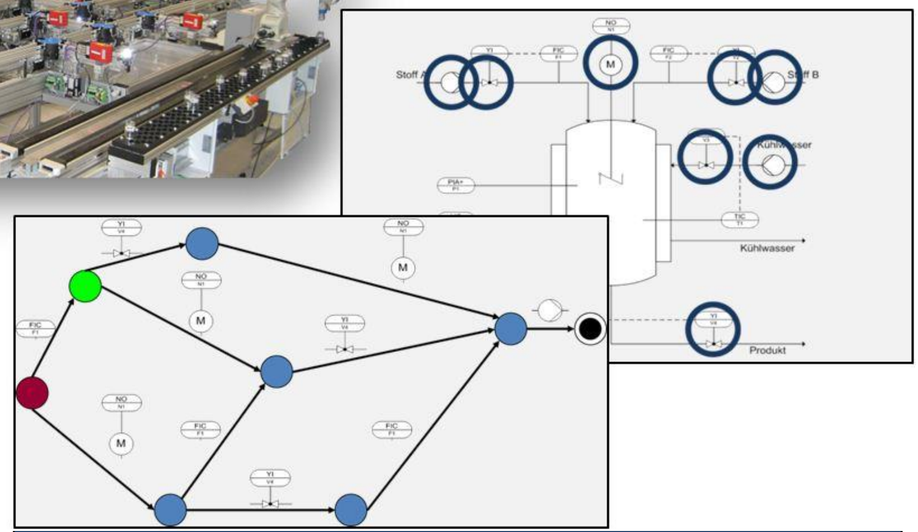
Model-Driven Development



Intelligent Distributed Systems



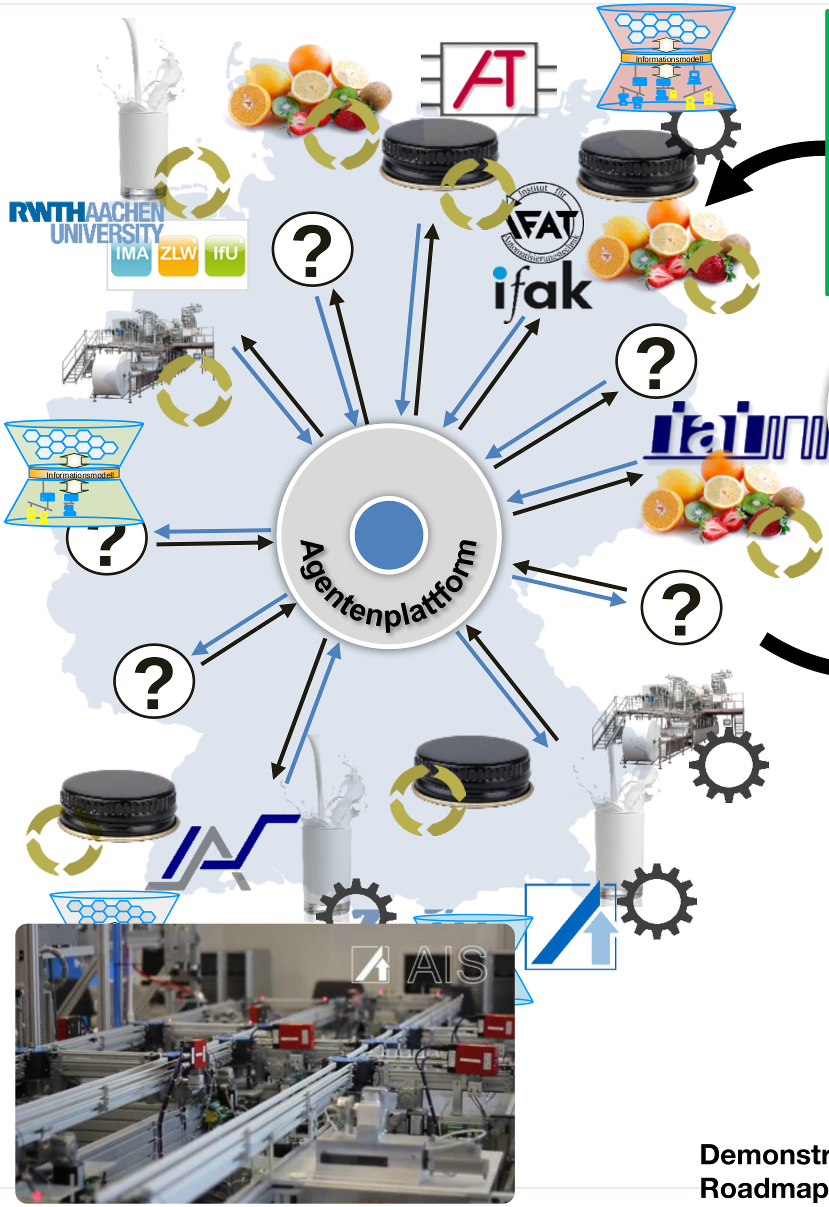


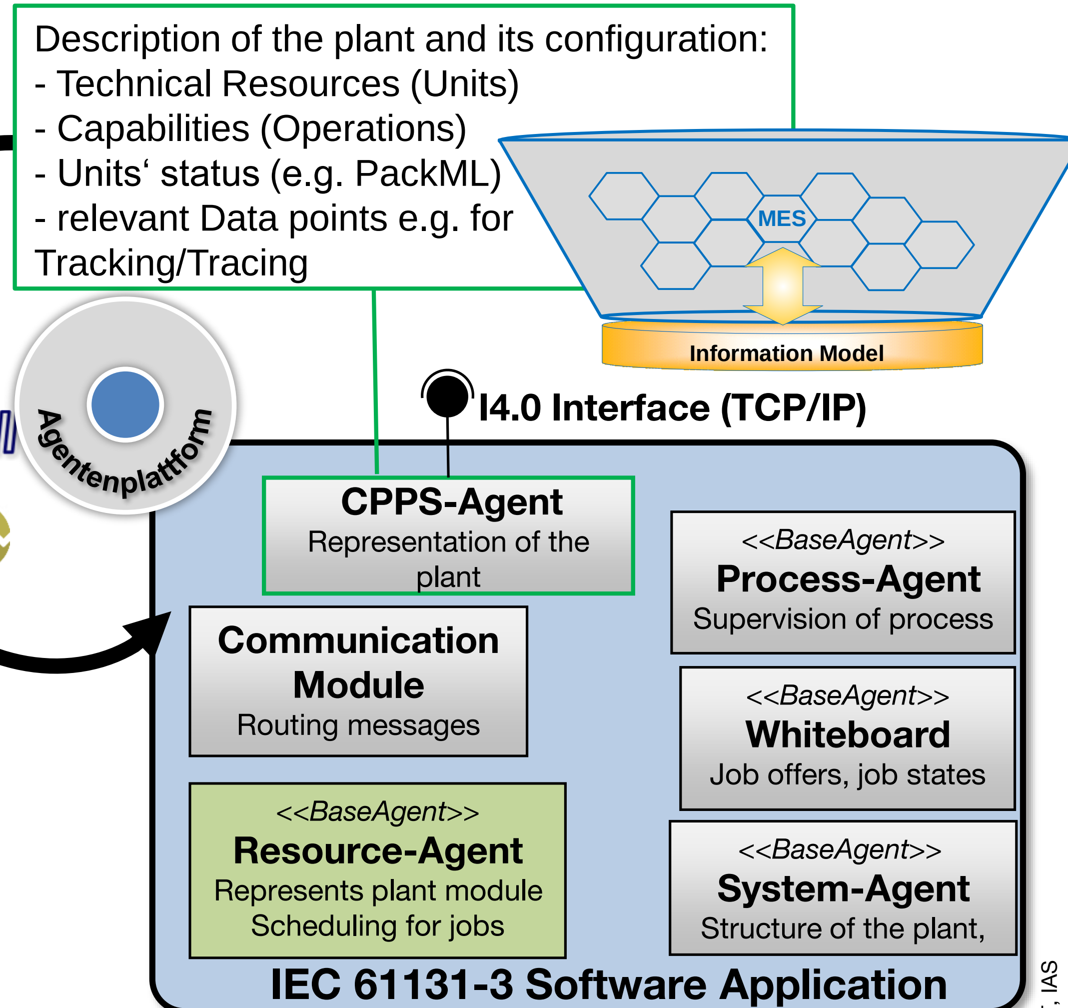


Smart Information

Big Data in aPS

My Joghurt – accepted Industrie 4.0 demonstrator





Now officially part of the PLATTFORM INDUSTRIE 4.0 roadmap

Demonstrator: http://i40d.ais.mw.tum.de
Roadmap: http://www.plattform-i40.de/I40/Navigation/DE/In-der-Praxis/Karte/karte.html

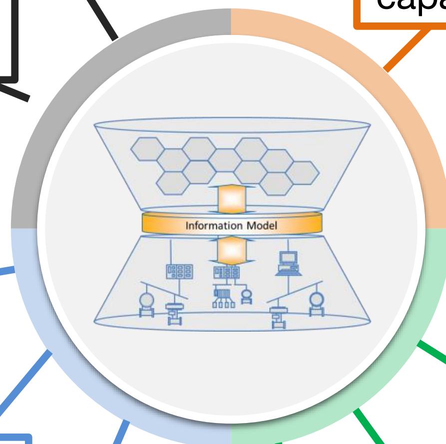
Characteristics of Cyber-Physical Production Systems (CPPS) – Industrie 4.0

Data processing for humans

Assistance systems for Engineering

Data processing and integration for humans

Architecture models (reference architecture) for a category of aggregation/modules related to properties, capabilities, interfaces…



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)

Intelligent products and production units

Production units with **inherent capabilities**

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

Flexible production units, **adaptable** to modified product requirements, allow also structural changes

Description of product and operating resources, e.g. ontology, for independent analysis, presentation, organisation and execution of a production process

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

Advanced Aspects on Model-driven Engineering and implementation of field level agents

1. Introduction of TUM and AIS
2. **Field level agents in automation**
3. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
4. Use case: Sensor reconfiguration modeled with SysML and code generation from SysML Model to PLC‘s
5. Agents‘ knowledge of valid combinations or sequences modeled with OCL
6. Metrics for adaptivity
7. Conclusion and future work

Trends in automation technology

- Global competition
- High customer requirements regarding individualisation of products
- Short life-cycles
- „Long-Living“ production platforms

Challenges in automation technology

- Process improvements, greater efficiency
- Stability
- Flexibility
- Responsiveness
- Greater reusability

Disadvantages of today’s way of thinking for software development

Today’s software development for automation systems

Function-oriented, process-oriented, state-oriented, object-oriented

System structure and behavior predetermined by design

Functionality of the system elements, relationships between elements

Statistic software model

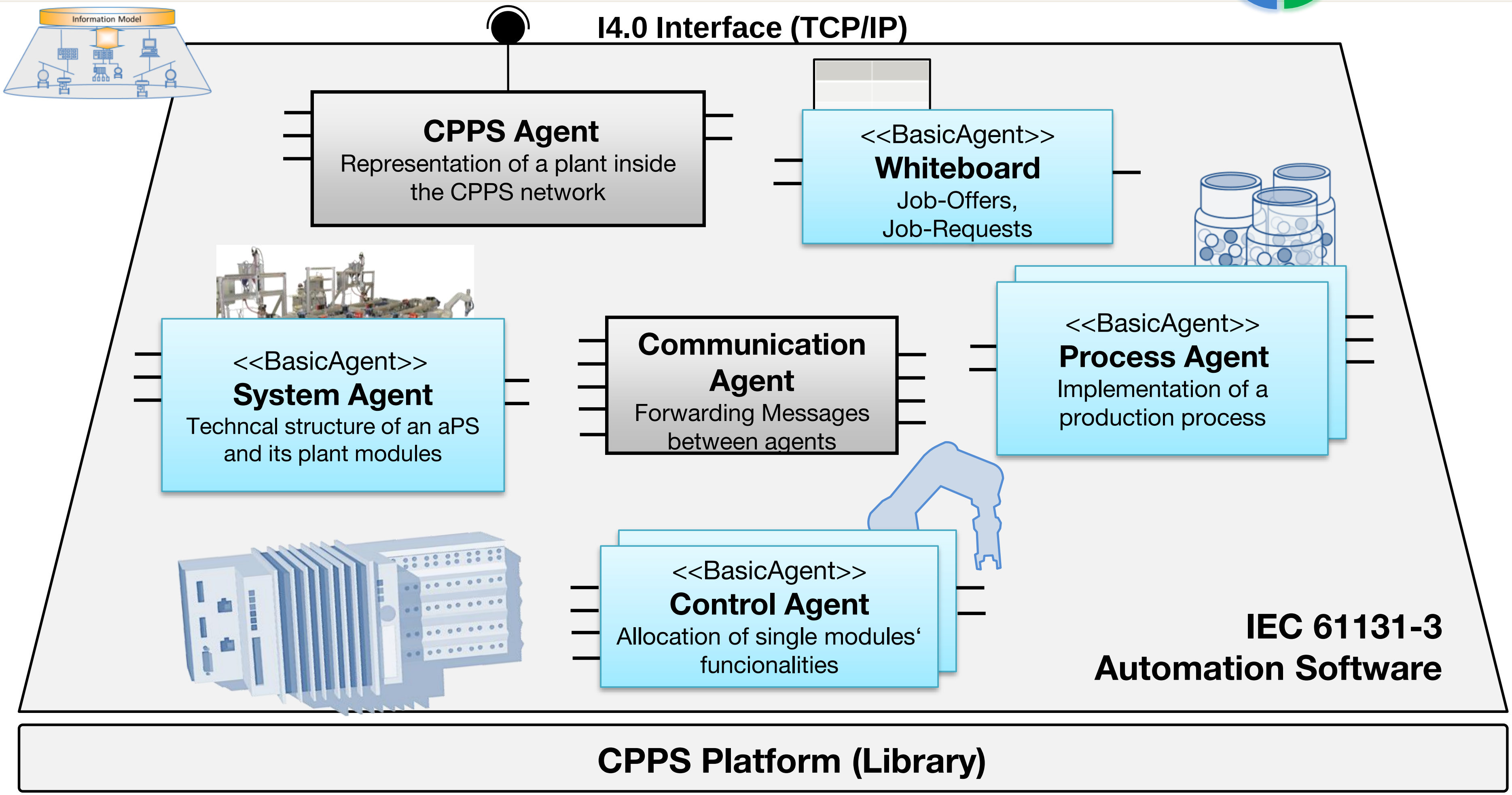
All relevant circumstances must be considered in the design

High dependency of elements

Failure of an element → Alternative defined in the design or failure

Aim: Software systems that can react to situations that are not specifically provided during design phase

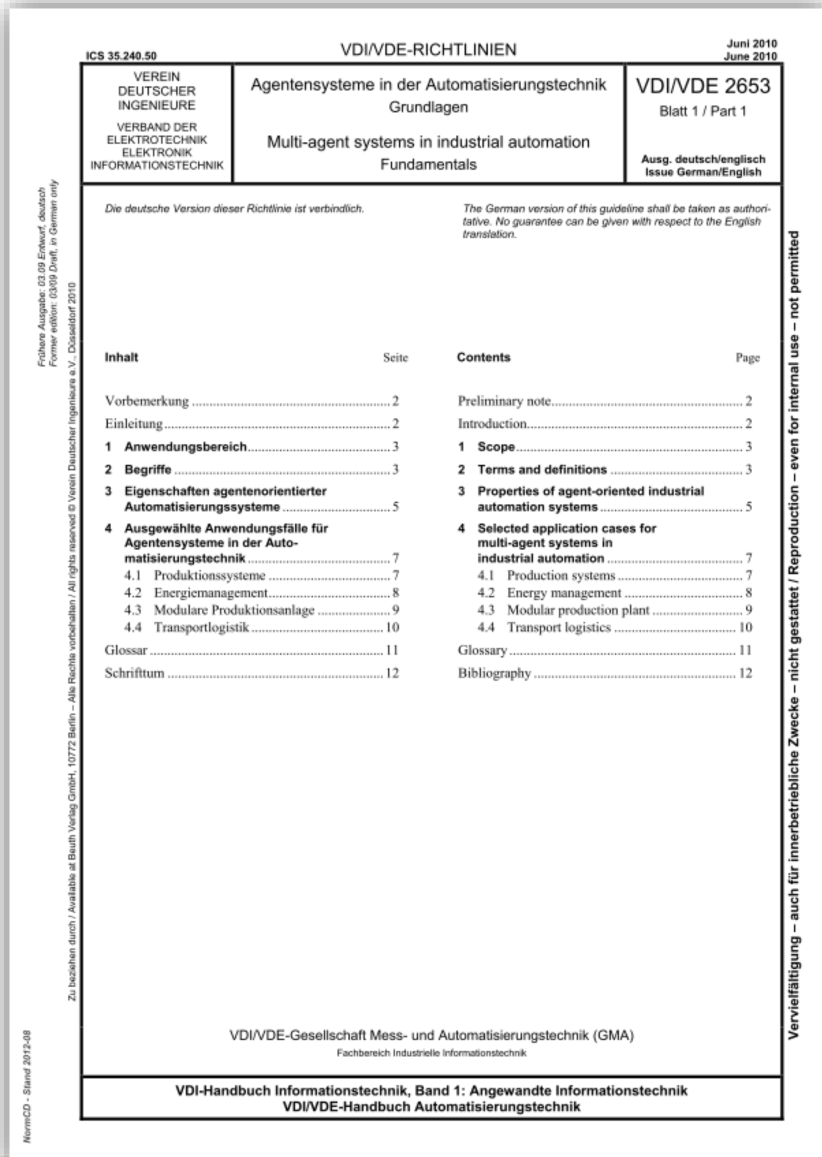
Definition of an agent



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**.

Source: VDI-Standard 2653 Sheet 1, 2010

Guideline VDI/VDE 2653 – Part 1


ICS 35.240.50

VDI/VDE-RICHTLINIEN

Juni 2010
June 2010

VEREIN DEUTSCHER INGENIEURE VERBAND DER ELEKTROTECHNIK ELEKTRONIK INFORMATIONSTECHNIK	Agentensysteme in der Automatisierungstechnik Grundlagen Multi-agent systems in industrial automation Fundamentals	VDI/VDE 2653 Blatt 1 / Part 1 Ausg. deutsch/englisch Issue German/English

Die deutsche Version dieser Richtlinie ist verbindlich.

The German version of this guideline shall be taken as authoritative. No guarantee can be given with respect to the English translation.

Inhalt Seite

Vorbemerkung 2
Einleitung 2
1 **Anwendungsbereich** 3
2 **Begriffe** 3
3 **Eigenschaften agentenorientierter Automatisierungssysteme** 5
4 **Ausgewählte Anwendungsfälle für Agentensysteme in der Automatisierungstechnik** 7
4.1 Produktionssysteme 7
4.2 Energiemanagement 8
4.3 Modulare Produktionsanlage 9
4.4 Transportlogistik 10
Glossar 11
Schrifttum 12

Contents Page

Preliminary note 2
Introduction 2
1 **Scope** 3
2 **Terms and definitions** 3
3 **Properties of agent-oriented industrial automation systems** 5
4 **Selected application cases for multi-agent systems in industrial automation** 7
4.1 Production systems 7
4.2 Energy management 8
4.3 Modular production plant 9
4.4 Transport logistics 10
Glossary 11
Bibliography 12

VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA)
Fachbereich Industrielle Informationstechnik

VDI-Handbuch Informationstechnik, Band 1: Angewandte Informationstechnik
VDI/VDE-Handbuch Automatisierungstechnik

Multi-agent systems in industrial automation

Part 1: Fundamentals

- Terminology for agents used in automation, basic concepts and properties of agent systems
- User groups: Operators and developers

Part 2: Development

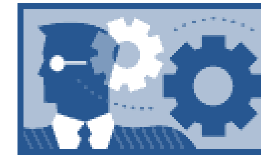
- Criteria for the selection and for the comparison of agent-oriented development methods and platforms
- Analysis of existing agent-oriented development methods
- User groups: Those who are concerned with the development of multi-agent systems

Part 3: Application

- Reports (industry and universities) using multi-agent systems in different application fields
- Definition of the problem and the problem solution with agents as well as the advantage of using agents
- User groups: Those who are concerned with the development of multi-agent systems

- New chapter „Learning“:
  - Integration of rule-based learning approaches, e.g. learning classifier systems, for critial but comprehensible control tasks
  - Classifier System reviews the applicated rules. → Good rules are kept in rule set. Unusable rules are deleted in rule set.
  - Initial learning process can be reduced through a priori rules
  - Using rules based on fuzzy logic improves the adaptation of the rules
- New chapter „Energy“:
  - New sub-chapters „Approach for controlling small distributed energy systems“, „Multimodal energy systems“, „Home automation“, „Micrgrids“, „Virtual power plants“, „Energy management in vehicles“
  - Switch to renewable energies requires small and distributed energy conversion systems
  - Realisation through multi-agent system
- New chapter „Mobile transport robots“
  - Using multi-agent systems for the control and communication of mobile transport robots
  - Collaboration between MAS to accomplish joint tasks
- New chapter „Data analysis “
  - Use the resulting data from different systems to generate more information and knowledge for the expansion of business areas and models from the collected data
  - Horizontal and vertical coupling of heterogeneous systems
  - If damage was recorded, compensation strategies are applicable
- New chapter „Smart Environment“
  - Complexes and hierarchical control structures are implemented by decentralized intelligent algorithms
  - Information about the building and the current situation are stored in a knowledge-based, semantic building model

What are the possible problems and benefits of the control of the production by software agents?



Issues:

- Violation of real-time requirements of the production process
- No determinism of the software by the autonomy of the agents
- For simple „linear“ processes and systems the overhead in the software may be too high

Advantages:

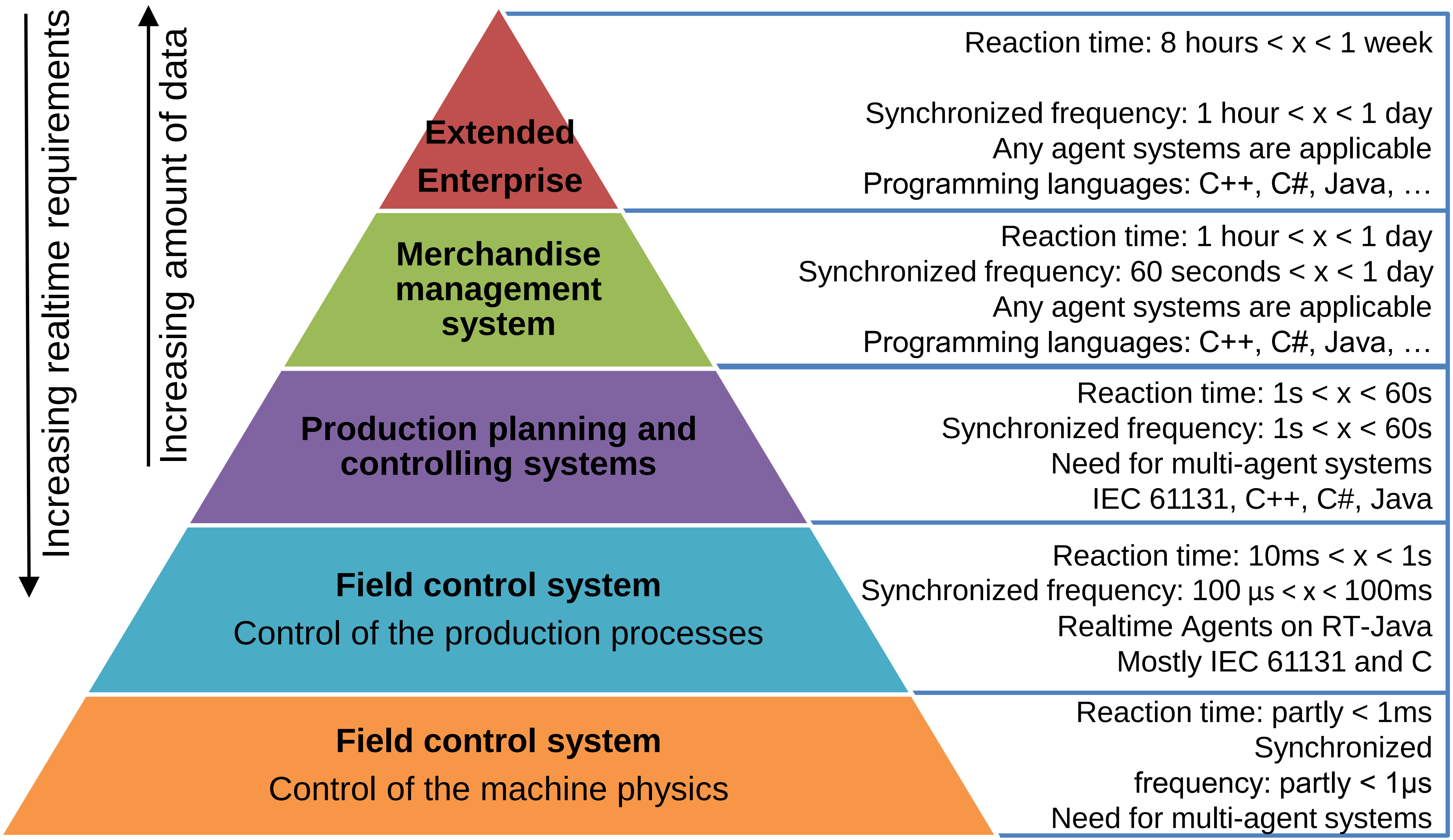
- Reconfiguration of processes in case of failure
- Compensation of failures of plant components
  - Adjustment of throughput / use of buffers in simple systems
- Optimization of plant processes under changing environmental conditions

Morphological box for classifying agents- Which agent types are most appropriate for field level agents?

<table>
<tr><th>Property</th><th colspan="8">characteristic</th></tr>
<tr><td>autonomy</td><td colspan="4">autonomous</td><td colspan="4">passive</td></tr>
<tr><td rowspan="2">architecture</td><td colspan="4">reactive</td><td colspan="4">proactive</td></tr>
<tr><td colspan="2">memoryless</td><td colspan="2">stored</td><td>FIPA</td><td>BDI</td><td>InteRRaP</td><td>…</td></tr>
<tr><td>communication</td><td colspan="4">synchronous</td><td colspan="4">asynchronous</td></tr>
<tr><td>language</td><td colspan="4">jointly defined</td><td colspan="4">interpretive</td></tr>
<tr><td>protocol definition</td><td colspan="3">Petri nets</td><td colspan="2">automata</td><td colspan="3">text</td></tr>
<tr><td>world model</td><td colspan="4">together</td><td colspan="4">Individually with intersection</td></tr>
<tr><td rowspan="2">adaptivity</td><td colspan="2" rowspan="2">Not adaptively</td><td colspan="6">learning mechanism</td></tr>
<tr><td colspan="2">imitative</td><td colspan="2">self-critically</td><td colspan="2">rule-based</td></tr>
<tr><td>mobility</td><td colspan="4">mobile</td><td colspan="4">stationary</td></tr>
<tr><td>competition</td><td colspan="4">cooperative</td><td colspan="4">competitive</td></tr>
<tr><td>method of cooperation</td><td colspan="4">negotiating</td><td colspan="4">dominant</td></tr>
<tr><td>Knowledge of their own abilities</td><td colspan="3">Non</td><td colspan="2">resources</td><td colspan="3">skills</td></tr>
<tr><td>Perception of the environment</td><td colspan="4">through communication</td><td colspan="4">through observation</td></tr>
</table>

Source: A. Wannagat und B. Vogel-Heuser: Kopplung von regelungstechnischer Analyse und Agentensystemen. *Holleczek, P.; Vogel-Heuser, B. (Hrsg.): Echtzeitsysteme im Alltag, Informatik aktuell, Springer Berlin Heidelberg*, 36-45.

Fields of application for agents

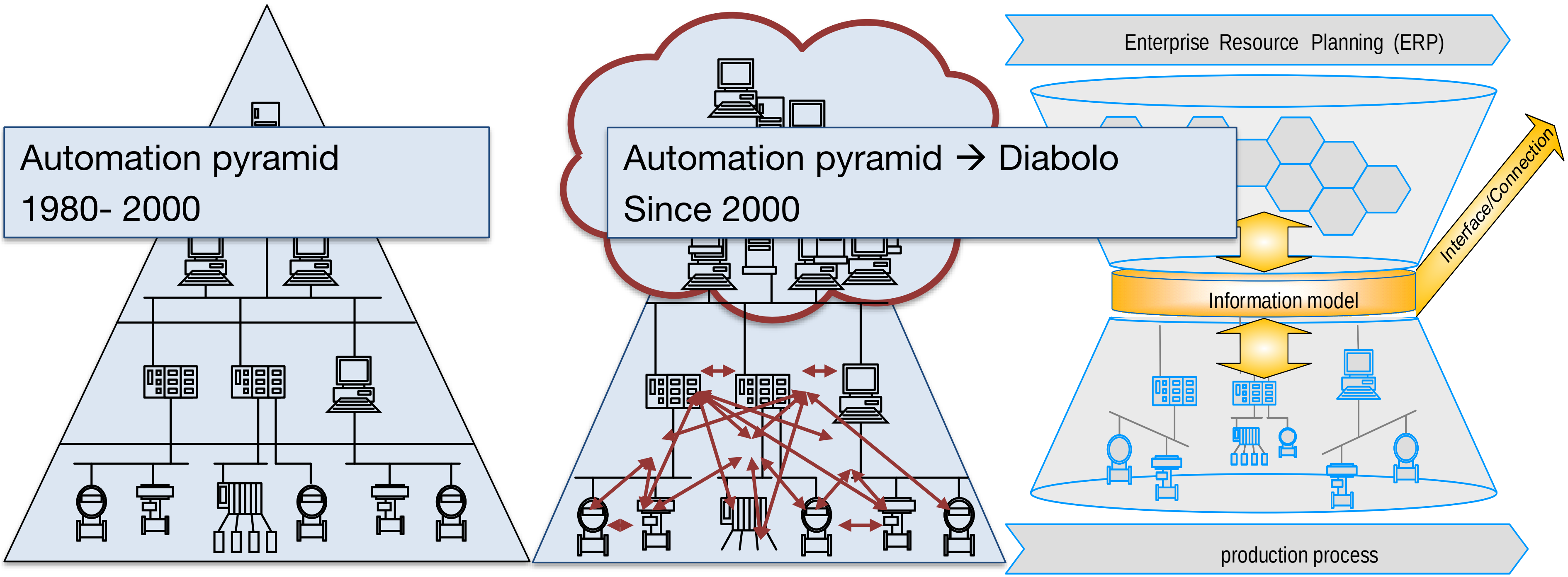


Source: See Lüder, A. Möglichkeiten und Grenzen Agentenbasierter Steuerungssysteme; 2006

Advanced Aspects on Model-driven Engineering and implementation of field level agents

1. Introduction of TUM and AIS
2. Field level agents in automation
3. **Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator**
4. Use case: Sensor reconfiguration modeled with SysML and code generation from SysML Model to PLC's
5. Agents' knowledge of valid combinations or sequences modeled with OCL
6. Metrics for adaptivity
7. Conclusion and future work

Changes in the requirements of automation architecture



Intelligent field devices: Integration of functions based on I/Os from the control level to the field level

Decentralized control system: Integration of automation functions in filed devices of production systems

Results

Increasing flexibility based on the interconnection of modularized field devices with production systems
→ Limitation of individual production systems

Source (right): B. Vogel-Heuser, G. Kegel, K. Bender und K. Wucherer: Global Information Architecture for Industrial Automation. In: Automatisierungstechnische Praxis (atp), Jahrgang 51 (2009), Heft 1, S. 108-115.

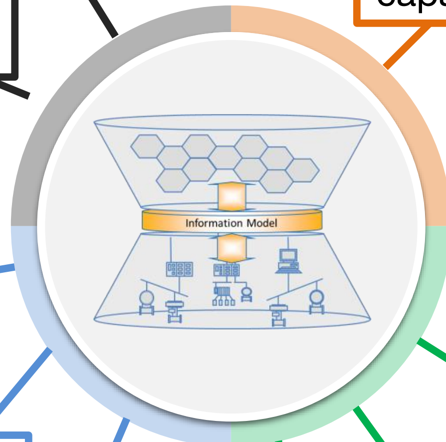
Characteristics of Cyber-Physical Production Systems (CPPS) – Industrie 4.0

Data processing for humans

Assistance systems for Engineering

Data processing and integration for humans

Architecture models (reference architecture) for a category of aggregation/modules related to properties, capabilities, interfaces…



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)

Intelligent products and production units

Production units with **inherent capabilities**

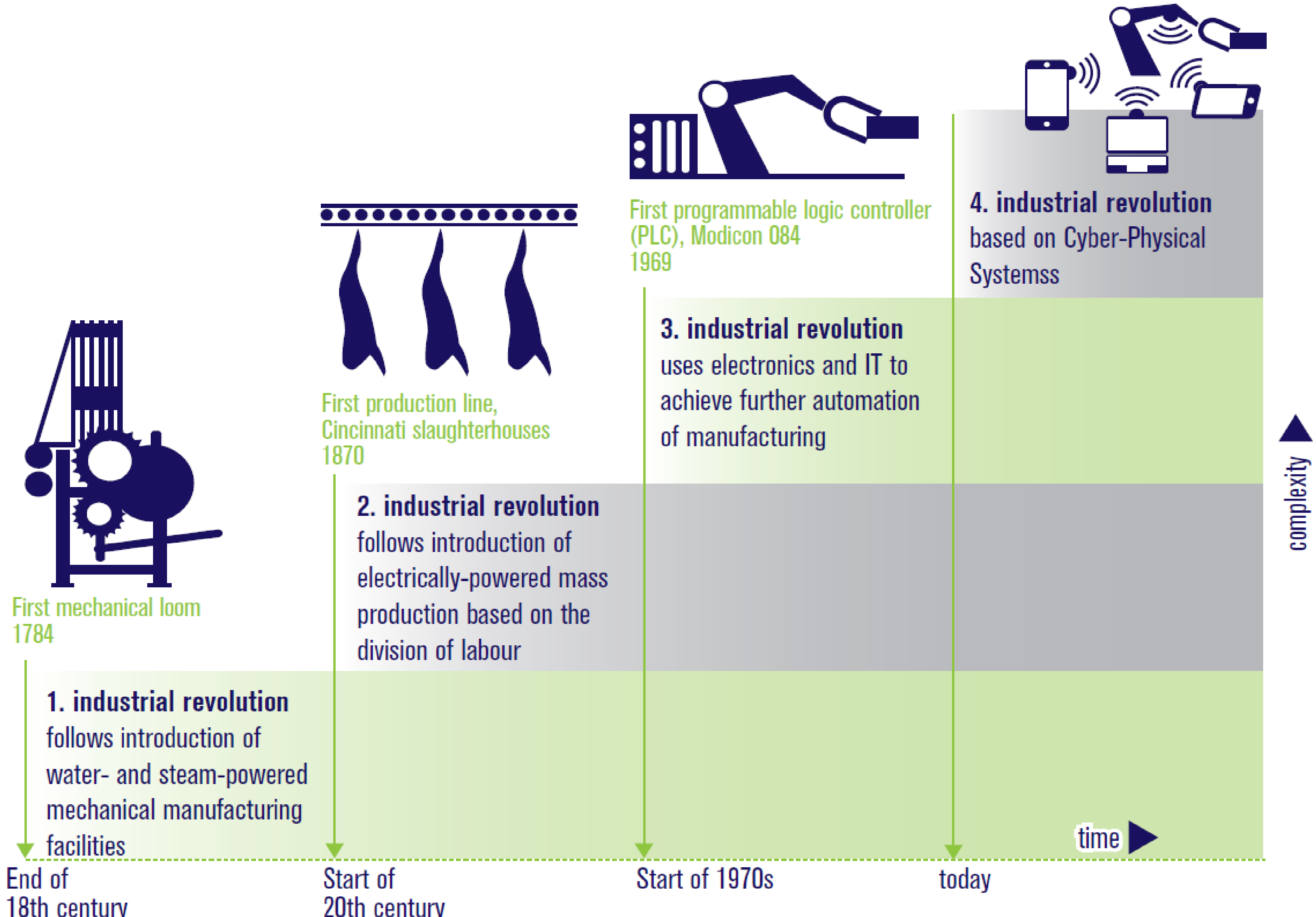
Data analysis of process and alarm data and connection with engineering data

Flexible production units, **adaptable** to modified product requirements, allow also structural changes

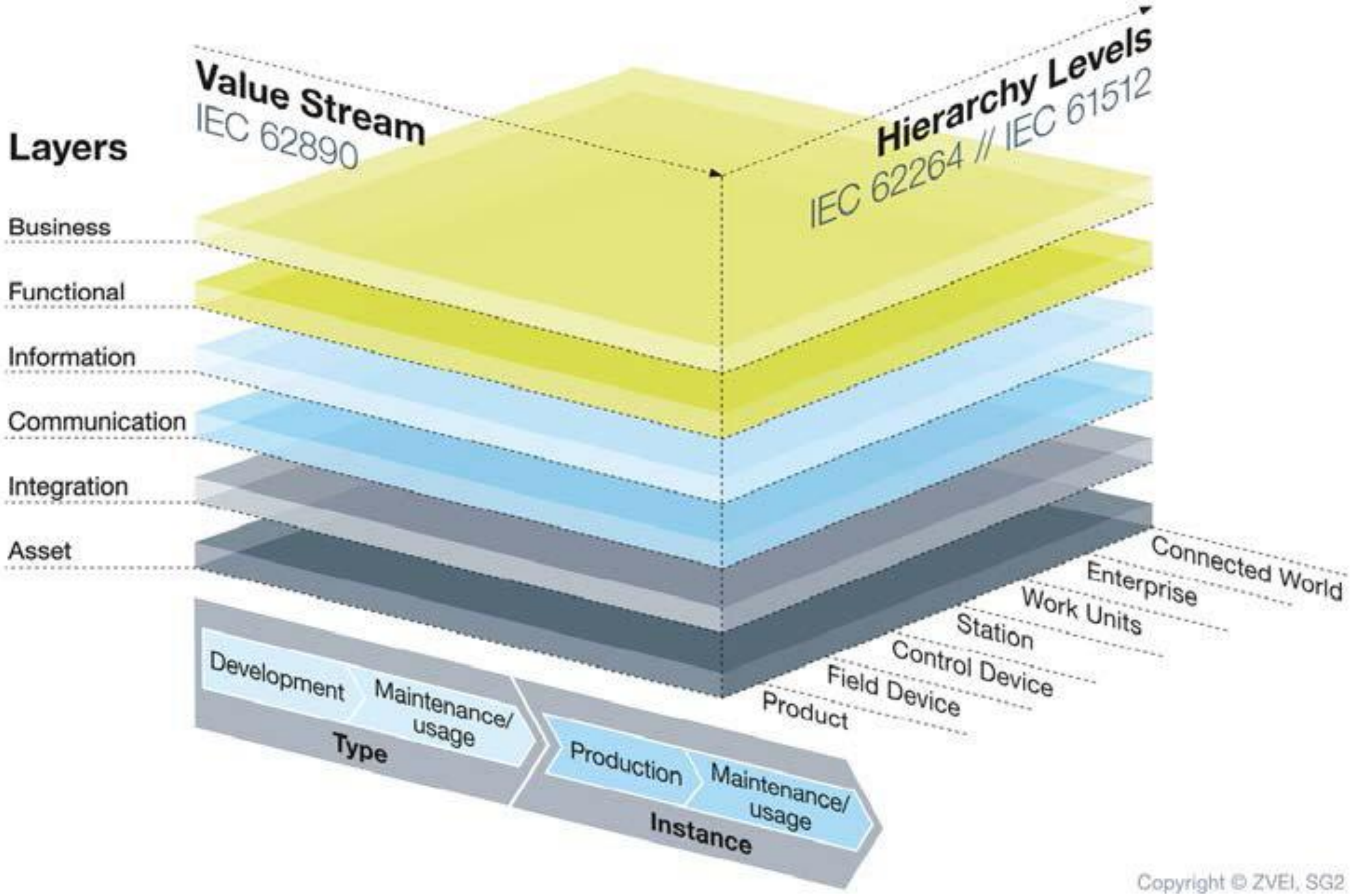
Description of product and operating resources, e.g. ontology, for independent analysis, presentation, organisation and execution of a production process

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

Four stages of industrial revolution



Source: DFKI 2011



Source: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik: Statusbericht; Industrie 4.0; Wertschöpfungsketten. Düsseldorf: VDI e.V., April 2014.

Characteristics of Industrie 4.0 component based on RAMI 4.0

Identifiability

- Unique identifier in network
- Physical objects are referenced by an ID
- Security
- **Timely Behavior**
- Different address types for I4.0 components and (application) objects

Virtual Description

Virtual representation (including dynamic behavior)

Security and Safety

- Protection for functionality and data (Security)
- Machine safety (Safety)
- Mindset-infrastructure security by Design (SbD)

I4.0-conform Semantics

Support semantics standardized for I4.0

State

State can be obtained at any time

Combinability

I4.0 components can be composed to form a bigger component

Quality of Service

Satisfaction of required characteristics as e.g. **real-time properties**, **dependability** etc.

I4.0-compliant services and states

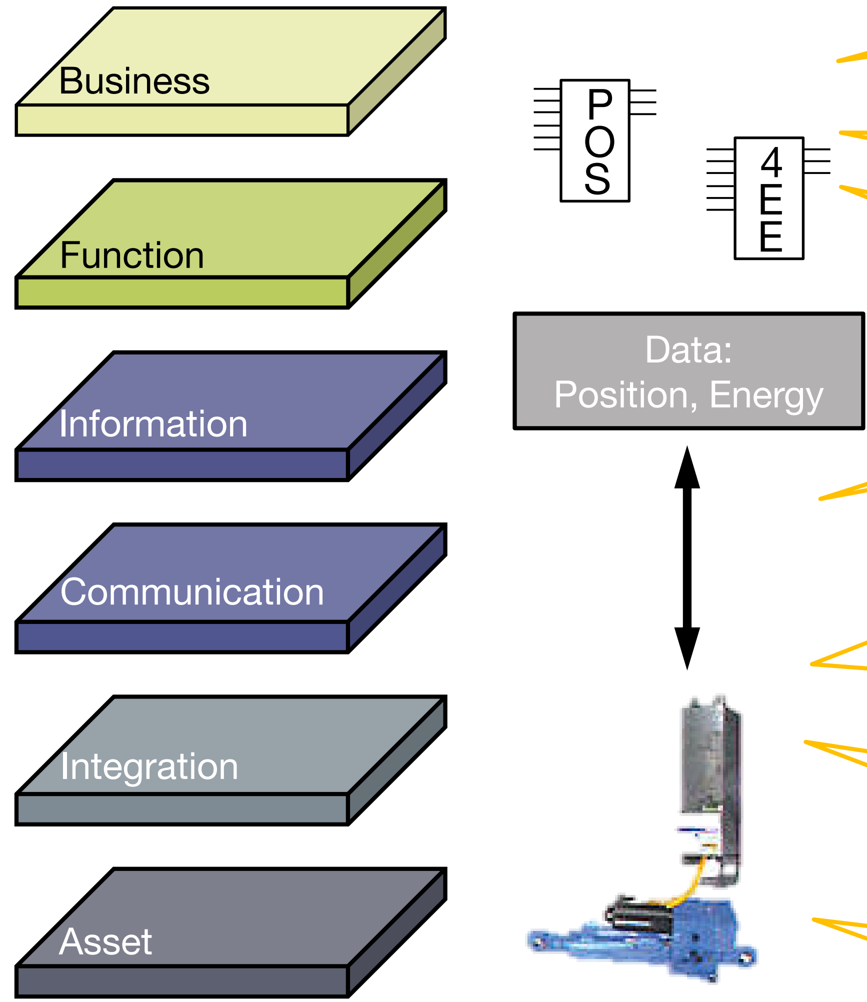
- Distinction between shop floor/office floor
- Protocols and application functions can be updated/ extended
- Application layers with different protocols

I4.0-conform communication

Self-identification (SOA-Service model)

Source: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik: Statusbericht; Industrie 4.0; Wertschöpfungsketten. Düsseldorf: VDI e.V., April 2014.

Example: servohydraulic axis RAMI 4.0



Energy management as a business process

Position control linked with machine-head control on functional layer

Position control with position assignment, collecting and evaluating of all energy data

Ethernet with OPCUA

Controls:
moment, velocity, …
additionally loadable functions:
safety, condition monitoring, control technology, energy save mode,…

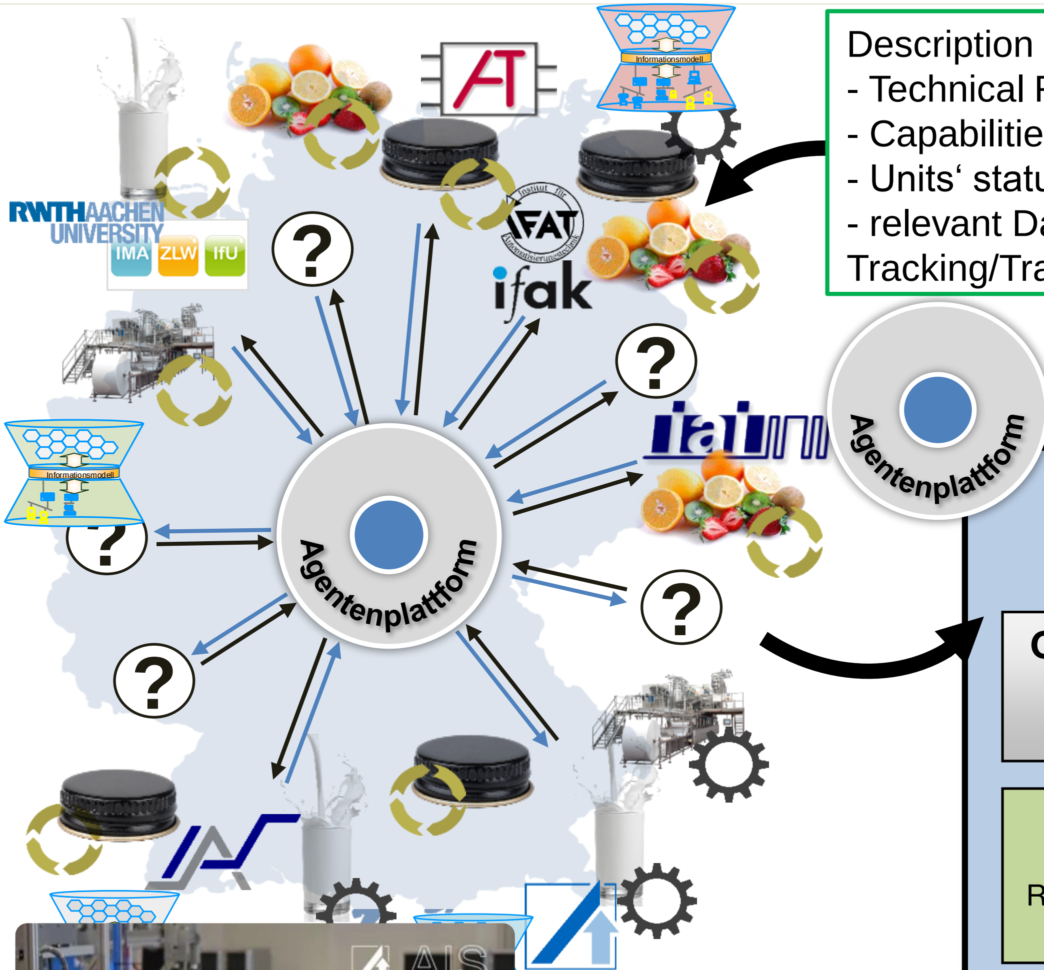
Booster:
webserver, compilation of all sensor information

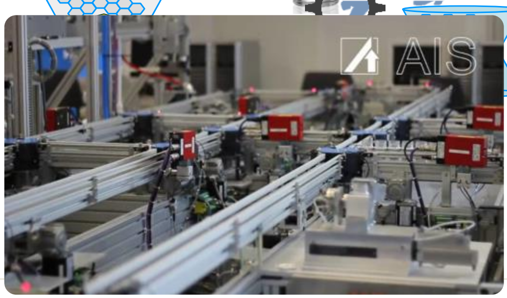
System:
cylinder, block with pump, servomotor, valves, …

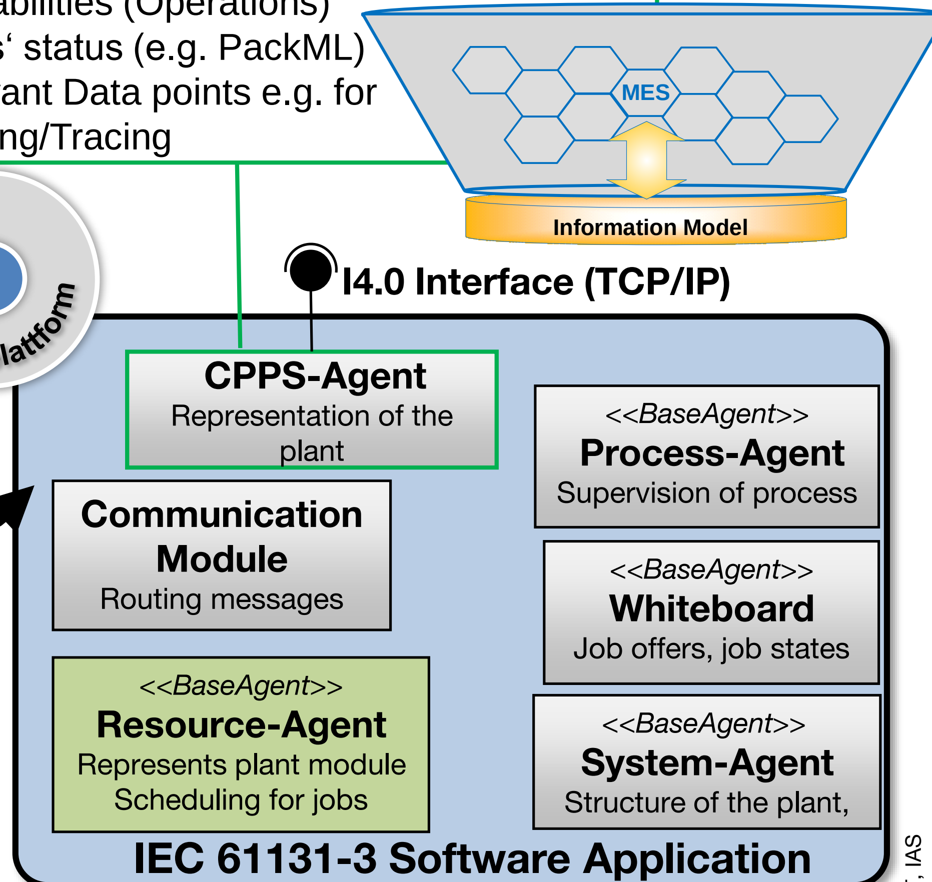
Source: open automation „ZVEI: RAMI 4.0 – next steps und das Referenzarchitekturmodell des IIC im Vergleich“, open automation, VDE Verlag, 15.09.2015, http://www.openautomation.de/detailseite/zvei-rami-40-next-steps-und-das-iira-im-vergleich.html.

My Joghurt – accepted Industrie 4.0 demonstrator

Description of the plant and its configuration:
- Technical Resources (Units)
- Capabilities (Operations)
- Units' status (e.g. PackML)
- relevant Data points e.g. for Tracking/Tracing







Now officially part of the PLATTFORM INDUSTRIE 4.0 roadmap

Demonstrator: http://i40d.ais.mw.tum.de

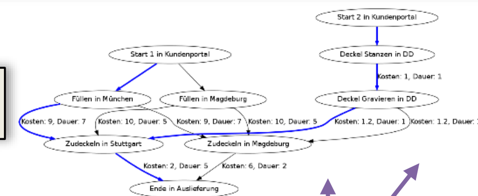
Roadmap: http://www.plattform-i40.de/I40/Navigation/DE/In-der-Praxis/Karte/karte.html

Customer places order

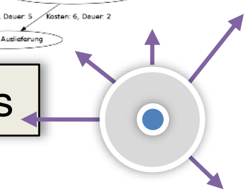
Splitting orders into sub-orders

Collecting charges and deadlines for sub-orders of system

Determine (new) schedule



Contracting (new) sub-orders

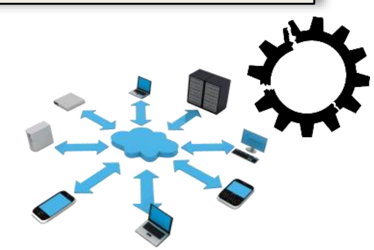


Production monitoring (operator and customer)

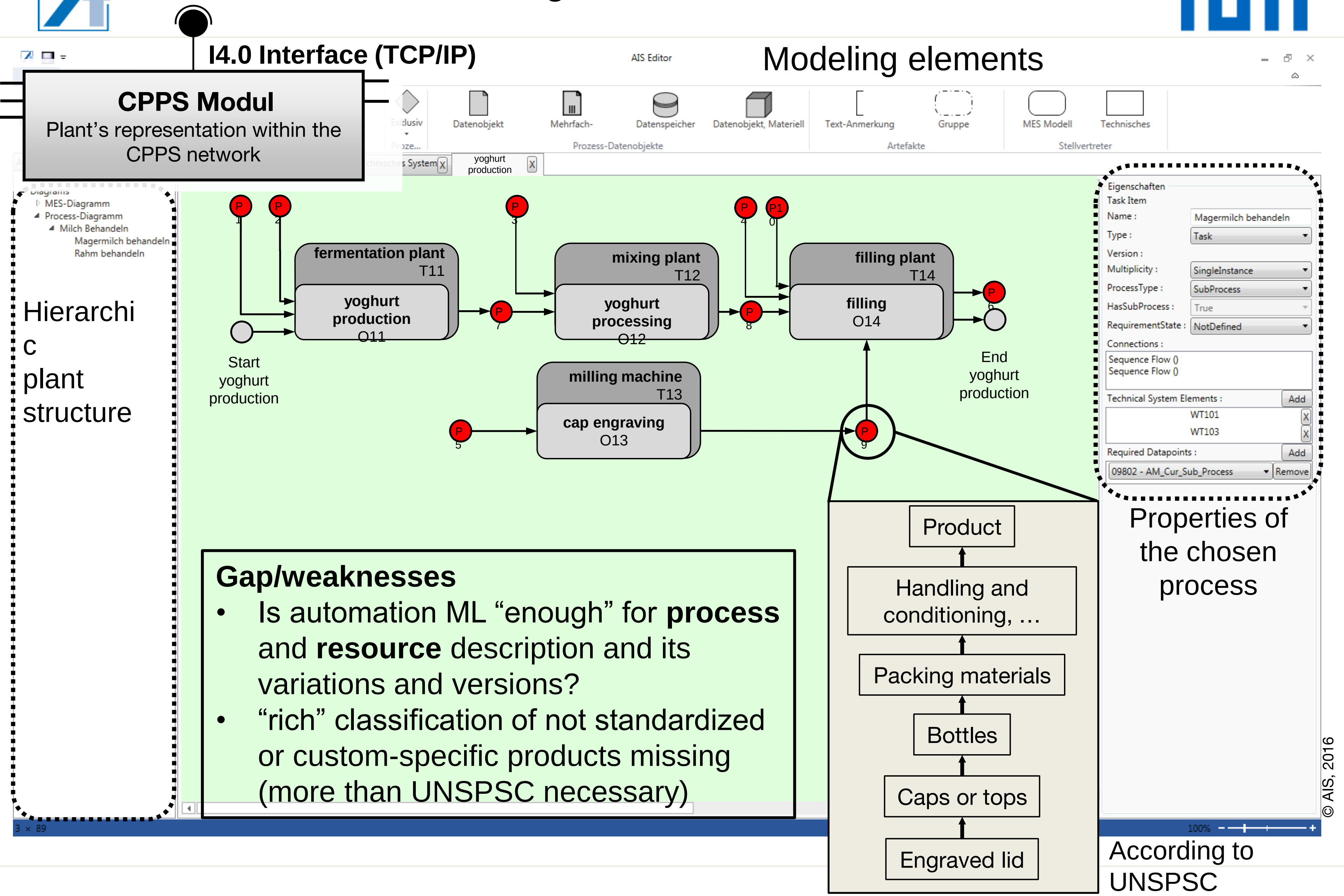
e.g.

Automatic troubleshooting

Send status report



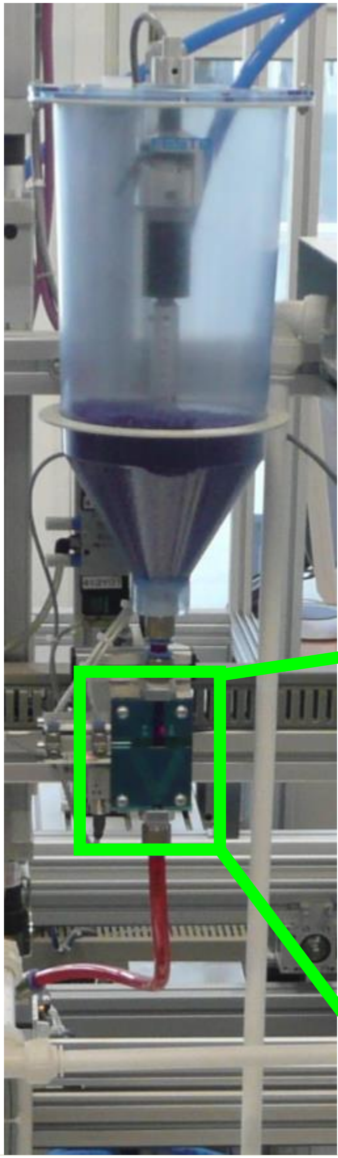
Modeling of CPPS with MES-ML



Gap/weaknesses

- Is automation ML "enough" for **process** and **resource** description and its variations and versions?
- "rich" classification of not standardized or custom-specific products missing (more than UNSPSC necessary)

Modeling of Ressource for pellet dispenser based on Automation ML



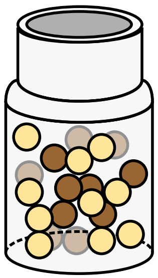
- SystemUnitClassLib
  - Filling station {**Class:**}
    - Tank {**Class: Role:**}
    - Pellet dispenser {**Class: Role:**}
      - Pneumatic cylinder {**Class: Role:**}
        - Interfaces
          - Input Air Extend {**Class:** PortConnector}
          - Input Air Retract {**Class:** PortConnector}
          - Behaviour {**Class:** PLCopenXMLInterface}
      - Case {**Class: Role:**}
        - Interfaces
          - Geometry {**Class:** COLLADAInterface}

<AutomationML/>

- CAEX for structural desciption
- PLCOpenXML for behavioral description
- Collada for geometric description

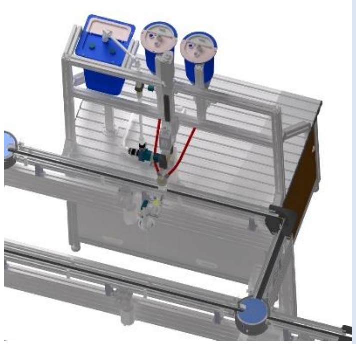
Input Diameter	
Name	Input Diameter
Description	
Value	8
Default Value	
Unit	mm
DataType	xs:string
Attribute1	
Output 1 Diameter	
Name	Output 1 Diameter
Description	
Value	8
Default Value	
Unit	mm
DataType	xs:string
Attribute1	
Attribute2	
Output 2 Diameter	
Name	Output 2 Diameter
Description	
Value	5
Default Value	
Unit	mm
DataType	xs:string

Checking attributes of ressource model and product model with ontologies



Product description

- **Name**: White chocolate balls
- **Viscosity**: 2.5 Pa*s
- **Yield strength**: 20 Pa
- **Diameter**: 0.5 cm
- **Aggregation state**: solid



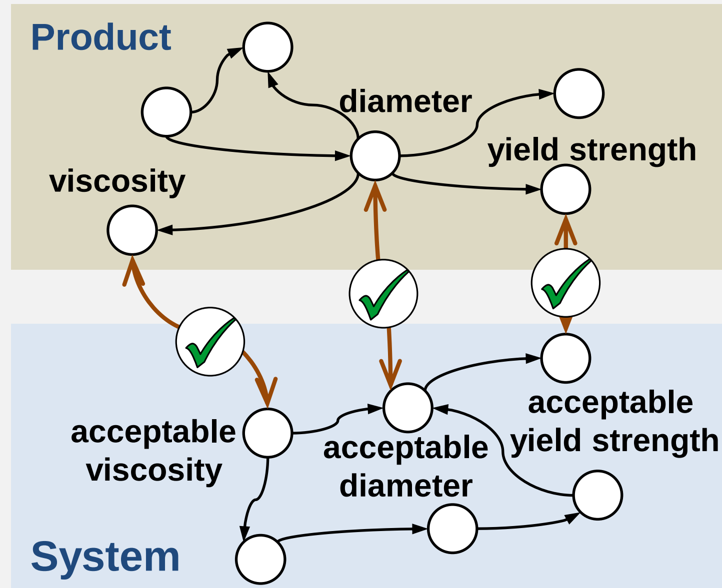
Ressource description

- **Name**: Filler
- **Acceptable viscosity**: 1..3 Pa*s
- **Acceptable yield strength**: 10..30 Pa
- **Acceptable diameter**: 0.2..1 cm
- **Functionality**: separate single solid

Ontology

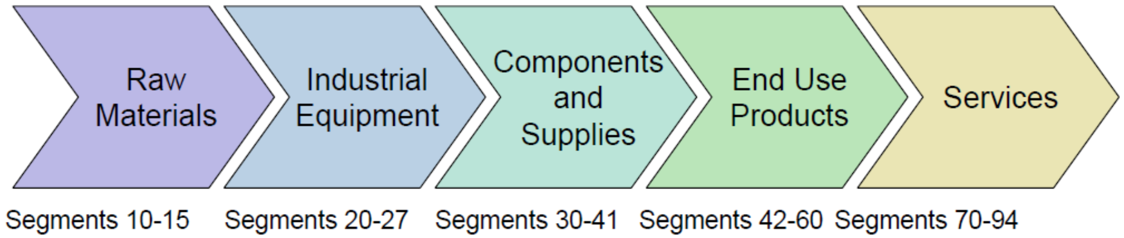
- Formal knowledge representation
- Provides the means to flexibly process knowledge

→ Basis to identify whether **filler** can manufacture yoghurts with **white chocolate balls**



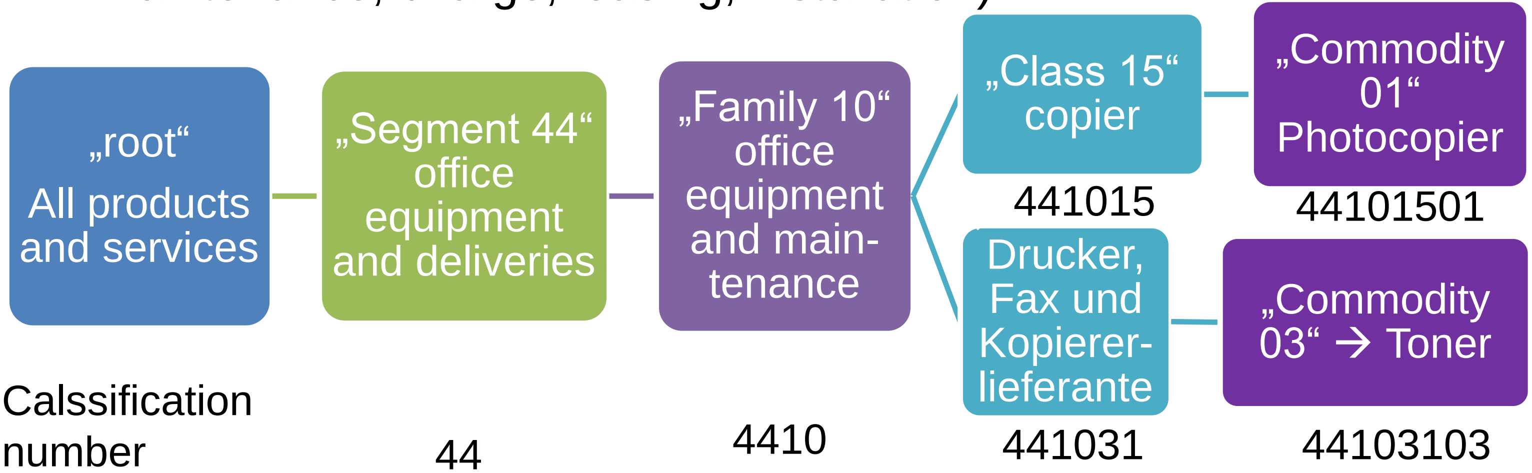
Mapping of **technical system**'s characteristics with **requirements from product** and **production process** by means of ontologies

Hierarchical open convention for classifying products and services (Segment, Family, Class, Commodity)



Specification

- UNSPSC-Code: 8 numbers
- Titel: Text with max. 120 letters
- Definition: Description text
- Business function (optional): 2 numbers (meaning e.g.: maintenance, charge, leasing, installation)

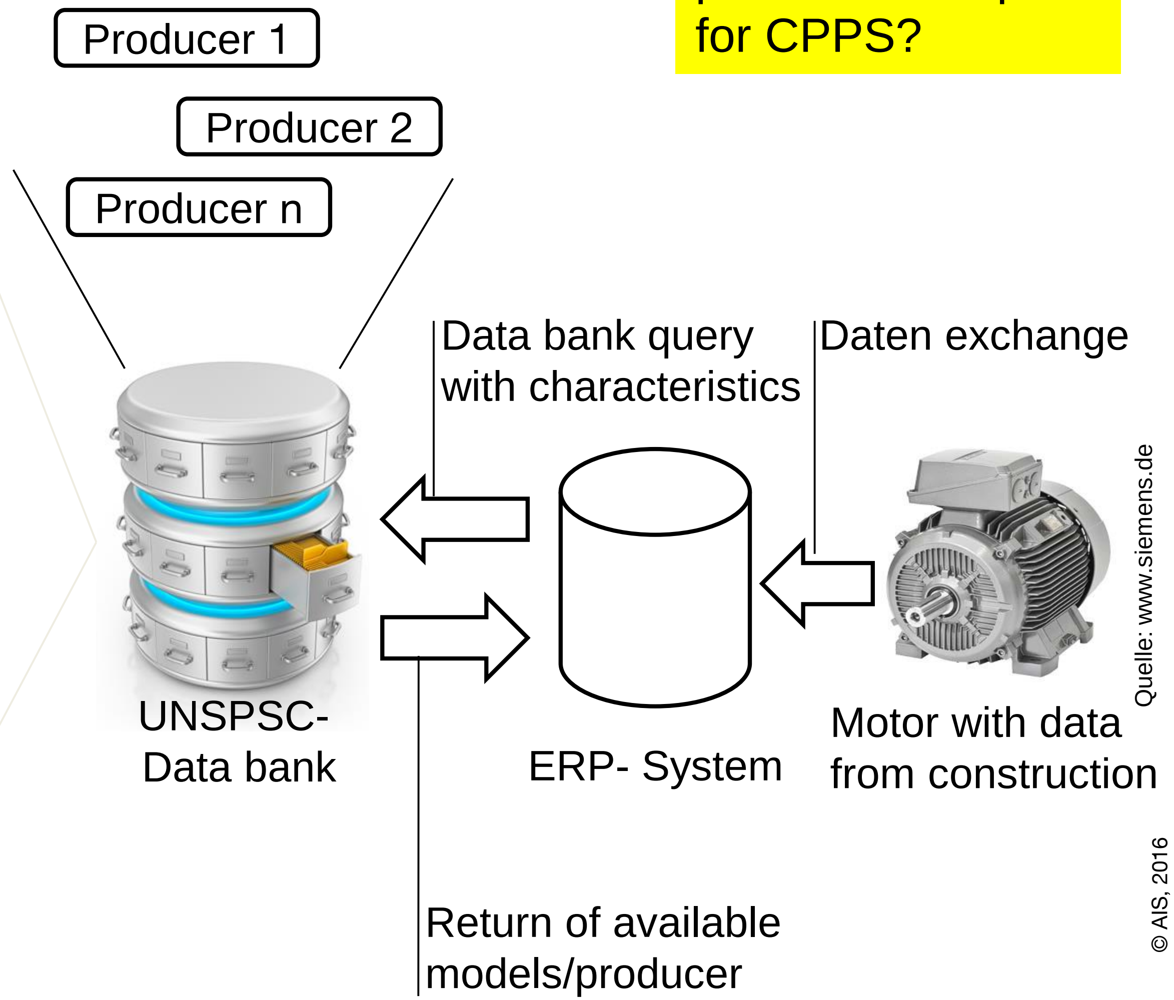


Specification of properties through standard characteristics:

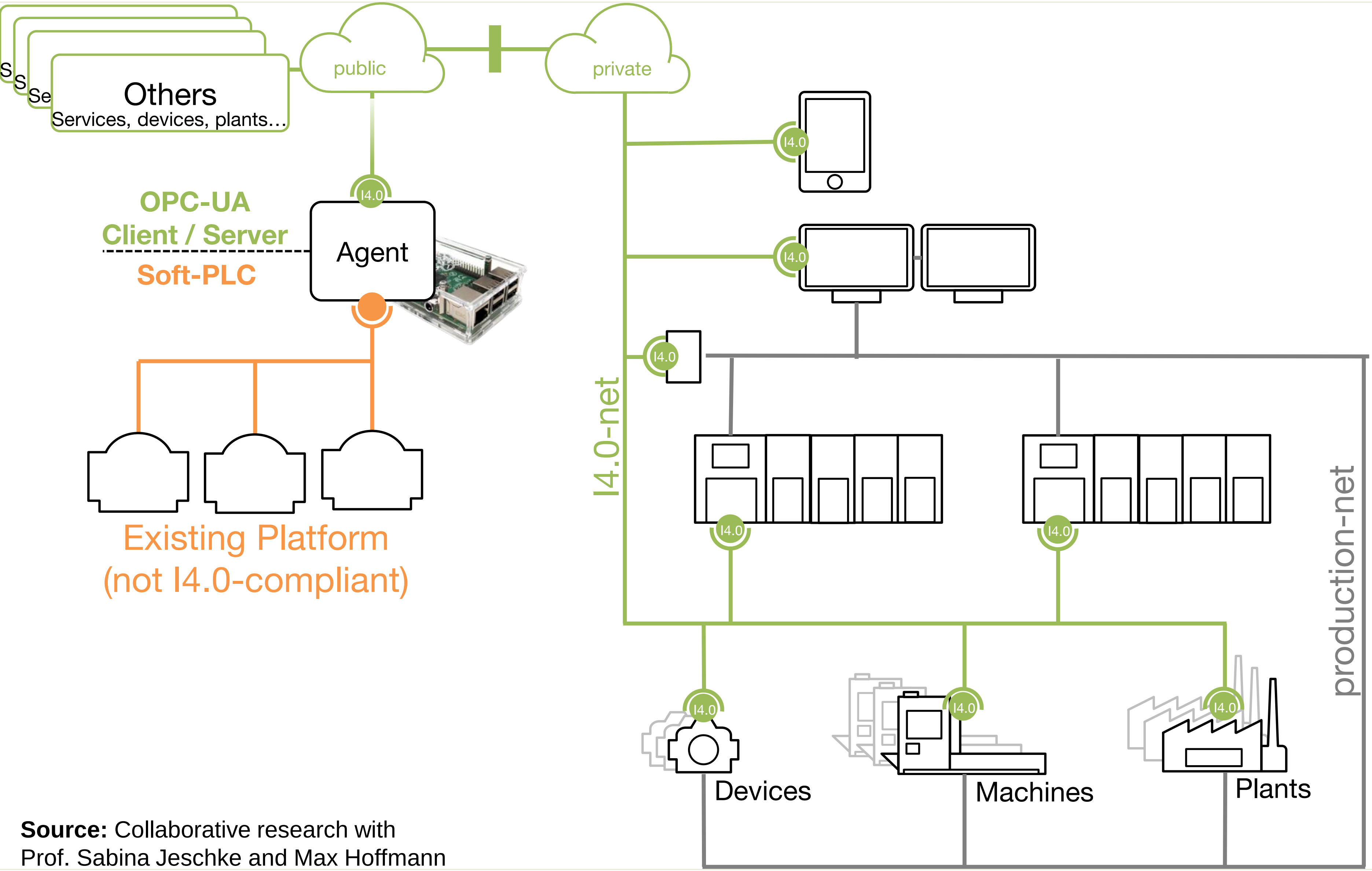
What lacks exist in product description for CPPS?

Characteristics 26101603

- min. rotational speed
- Field speed
- Type of cooling
- Size of the motor (DC)
- Design of the motor (DC)
- Nominal rotational speed
- Allowable stress
- Power loss
- Protective system
- Rated power
- Power loss
- Type of exciter
- Mode



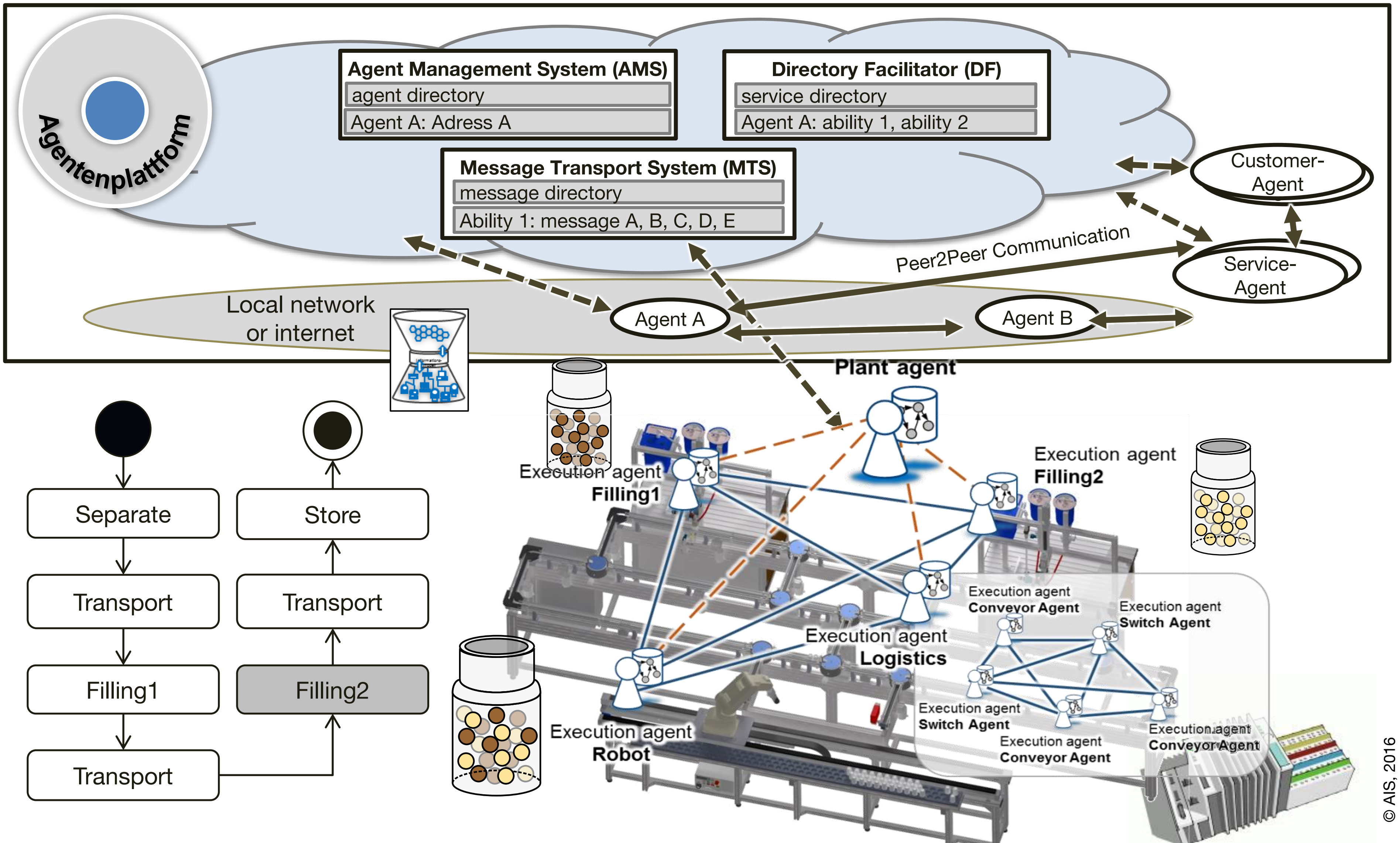
Agents as interfaces for Industrie 4.0 extensions to OPC UA or MQTT for CPPS



Source: Collaborative research with
Prof. Sabina Jeschke and Max Hoffmann

Source: cf. ABB AG / Plattform I4.0

Self-adapatation of an CPPS



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

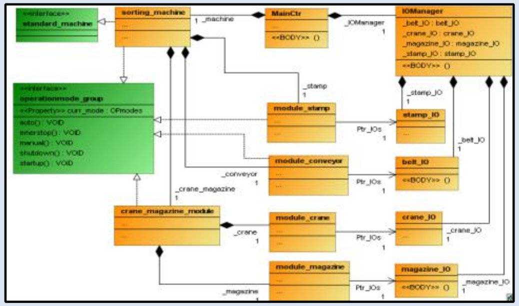
Source: B. Vogel-Heuser: Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik. In: Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer, 2014.

Advanced Aspects on Model-driven Engineering and implementation of field level agents

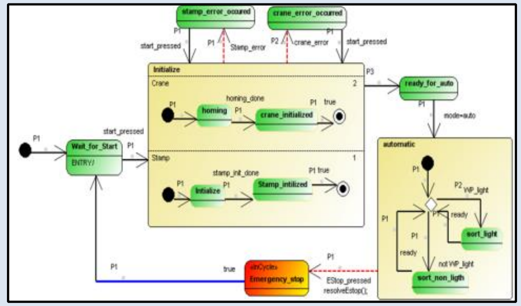
1. Introduction of TUM and AIS
2. Field level agents in automation
3. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
4. **Use case: Sensor reconfiguration modeled with SysML and code generation from SysML Model to PLC‘s**
5. Agents‘ knowledge of valid combinations or sequences modeled with OCL
6. Metrics for adaptivity
7. Conclusion and future work

UML-based implementation of IEC 61131-3 code with SysML expansion for software agents

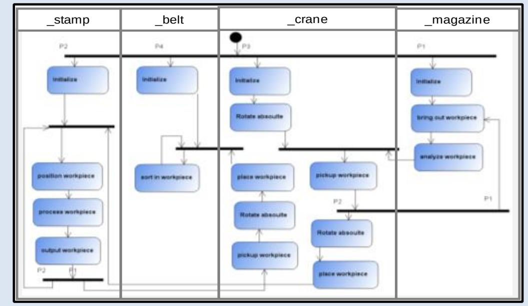
UML-Plugin for CODESYS V3



Class diagram

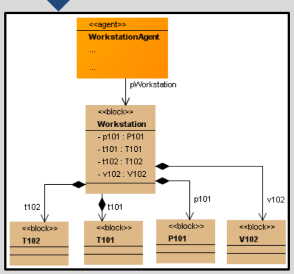


State diagram

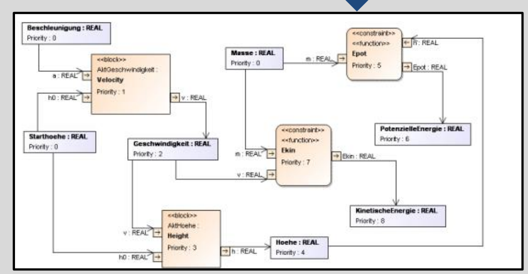


Activity diagram

Code generation for IEC 61131-3 (bidirectional synchronization)



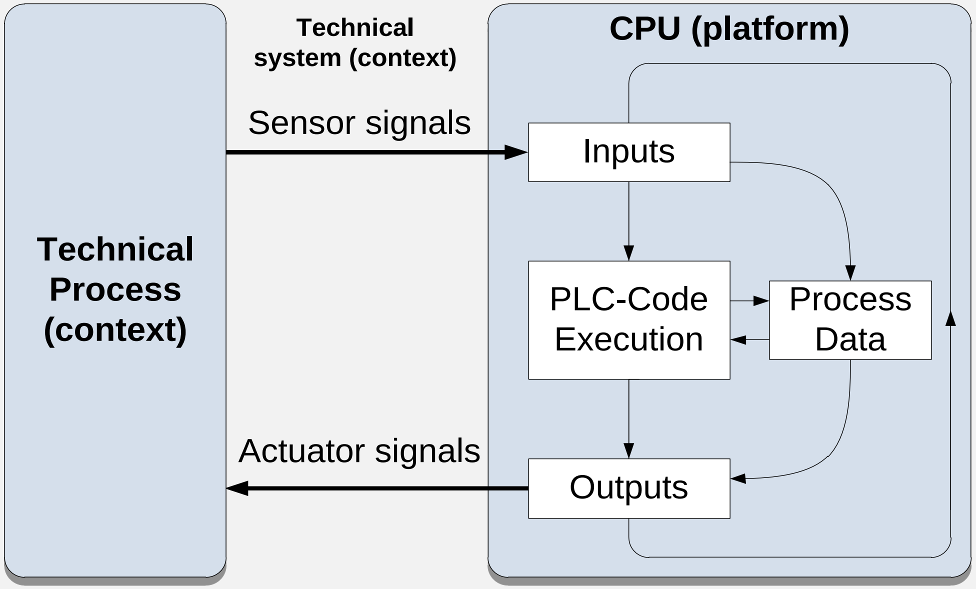
Block definition diagram



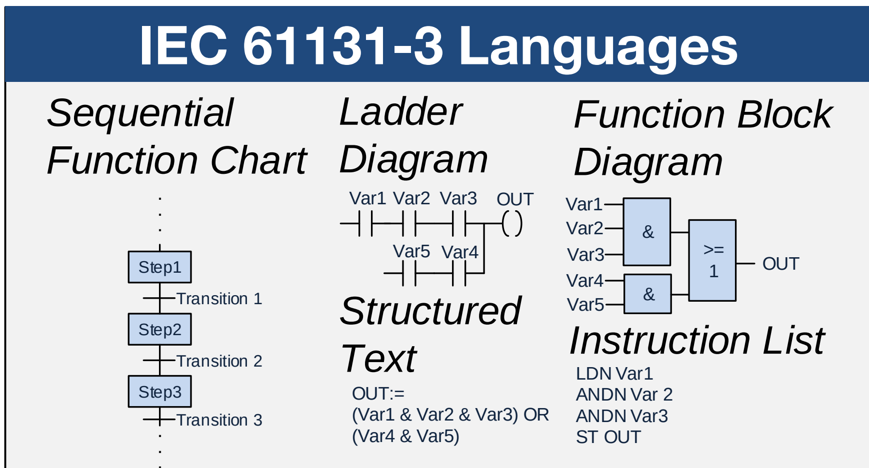
Parametric diagram

SysML-Plugin „KREA“ for TwinCAT 3 (CODESYS Family)

Technical constraints of the automation system

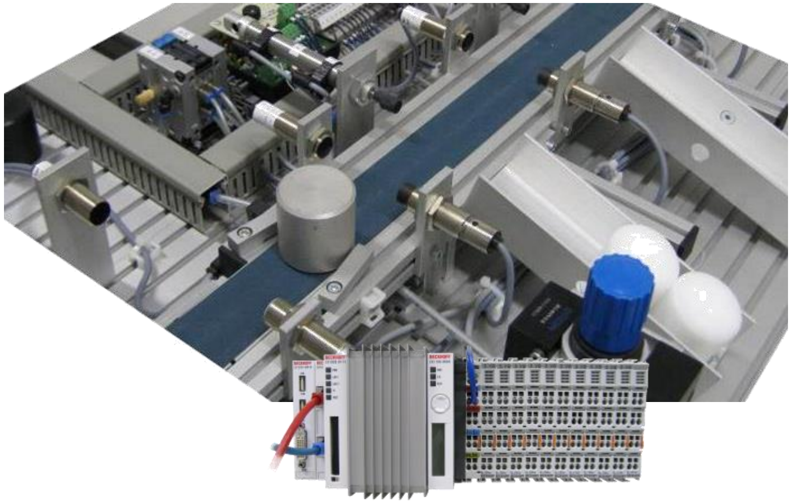


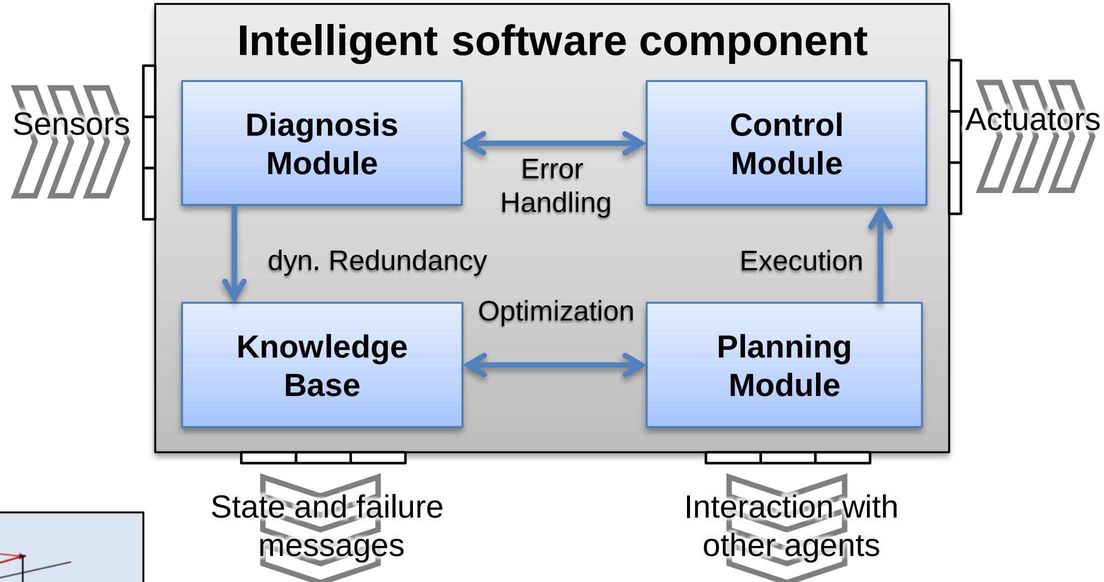
- Real-time requirements of automated production systems → hard real-time for the platform (PLC)
- Cyclic behavior of the platform (1µs – 1s)
- Classical PLC as well as Soft-PLC (PC-based) programmed in IEC 61131-3 Languages
- Increasing amount of IPC and C, C-derivatives
- ***Online change is mandatory***



IEC 61131-3 Programming Languages

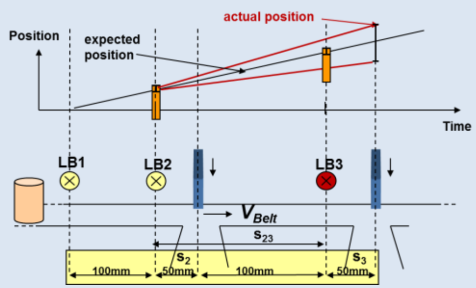
- Proprietary programming languages: Structured Text (ST), Ladder Diagram (LD), Instruction List (IL), Sequential Function Chart (SFC), Function Block Diagram (FBD)
- **Upcoming: C**





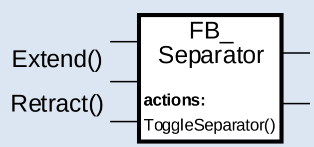
Diagnosis Module

- Evaluation of sensors values
- Execution of failure diagnosis



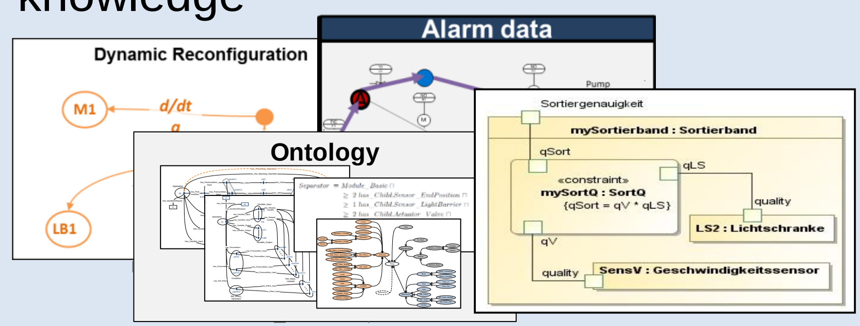
Control Module

Control of the plant module or other sub-agents

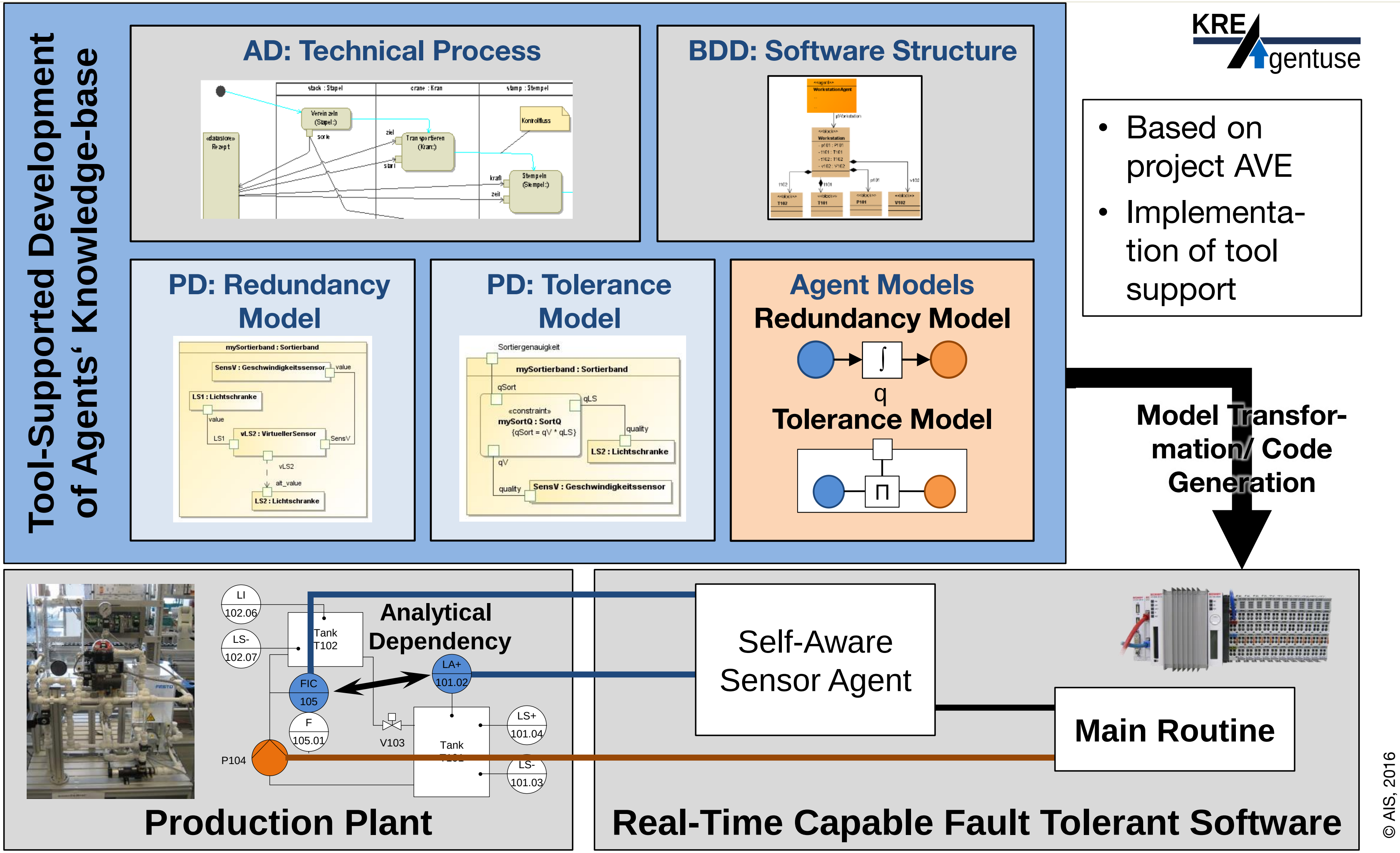


Knowledge Base

Models of the agents‘ local knowledge

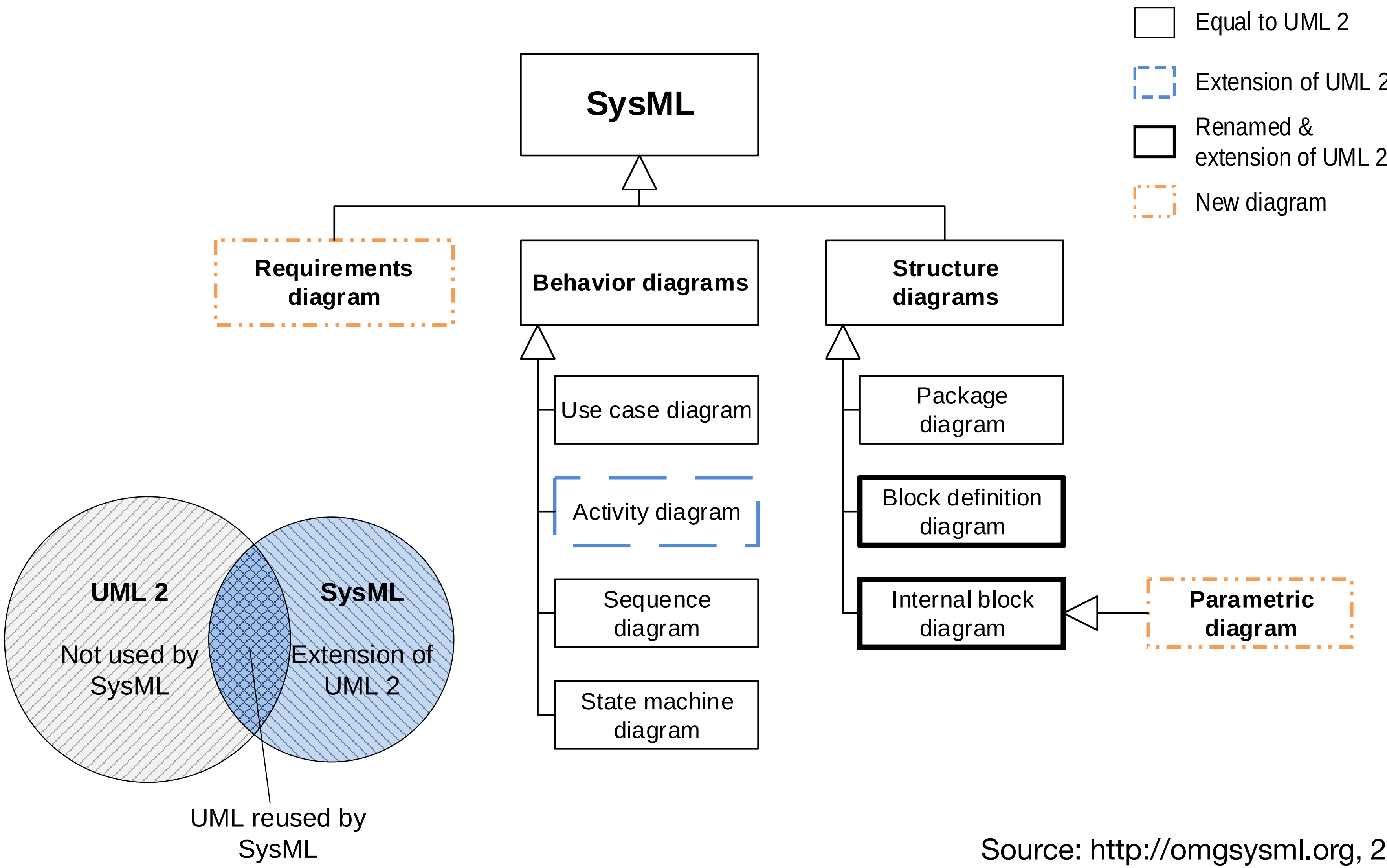


KREAagentuse:
SysML-based automation software development



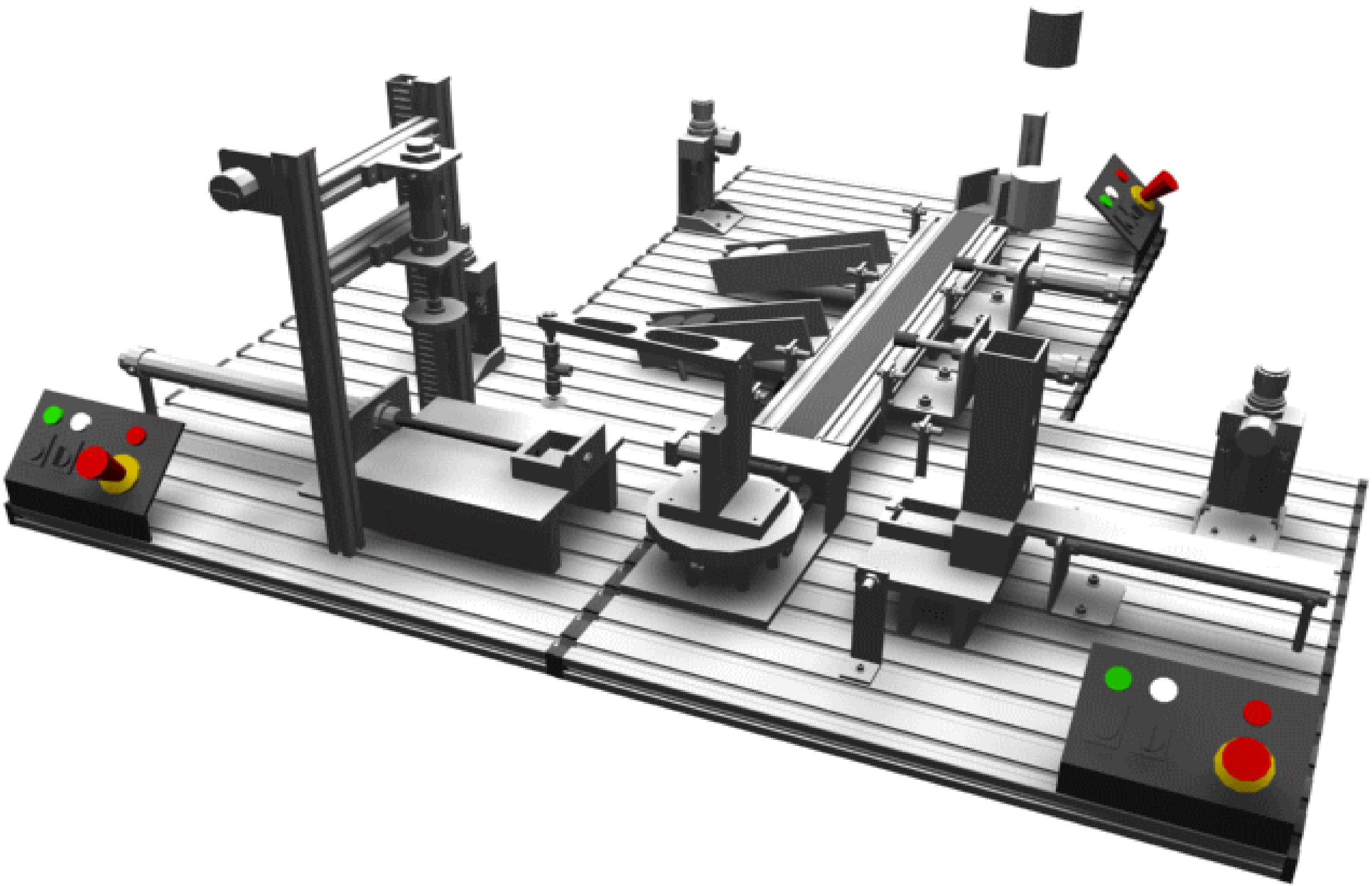
Source: Frank et al. 2011, Schütz et al. 2012, DFG funded project KREAagentuse

Construction of SysML diagrams



Source: http://omgsysml.org, 2007

Introduction of the small lab scale production system pick-and-place-unit (PPU)



Requirement in text form: The workpiece must be stamped for at least 10 s with a force of at least 10 N.

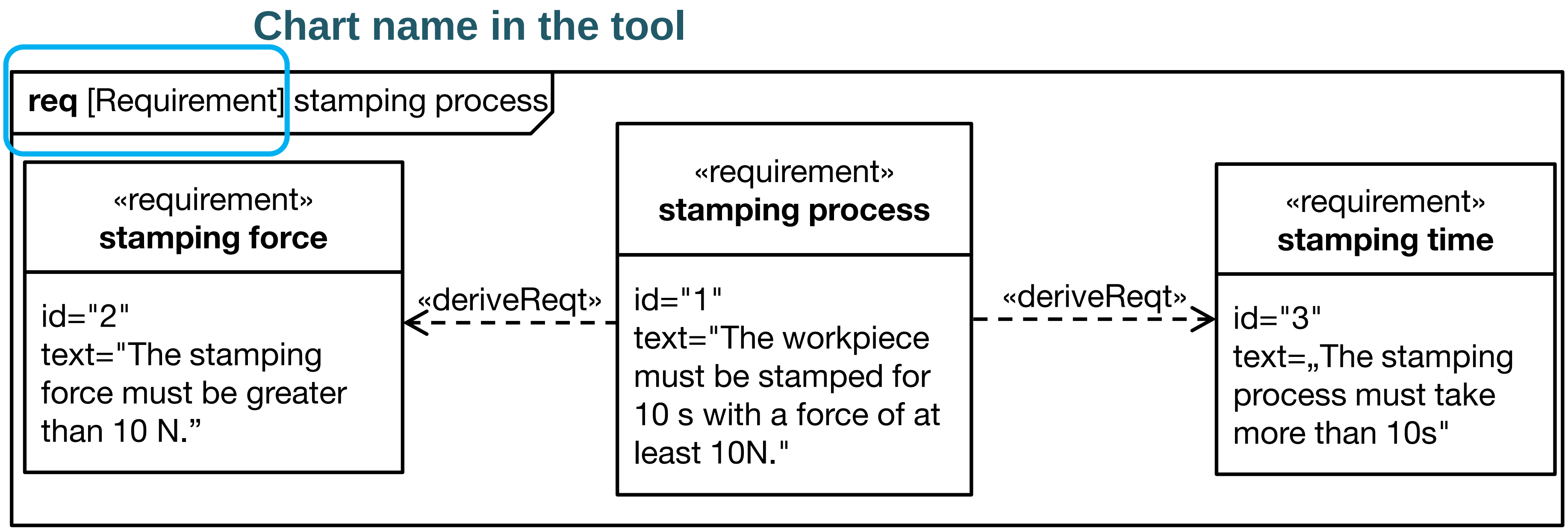
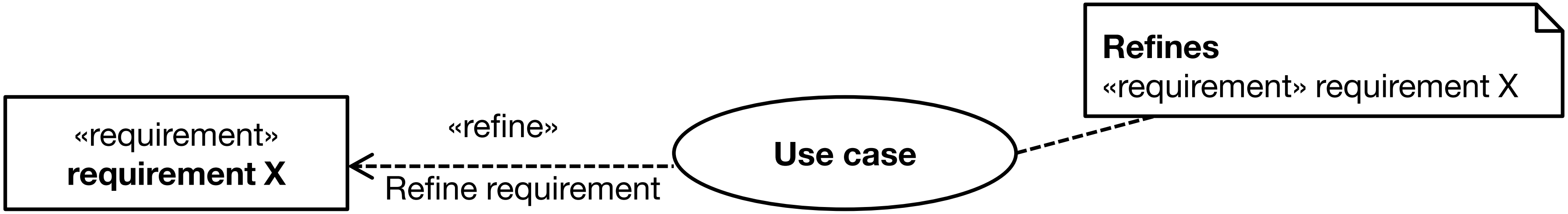


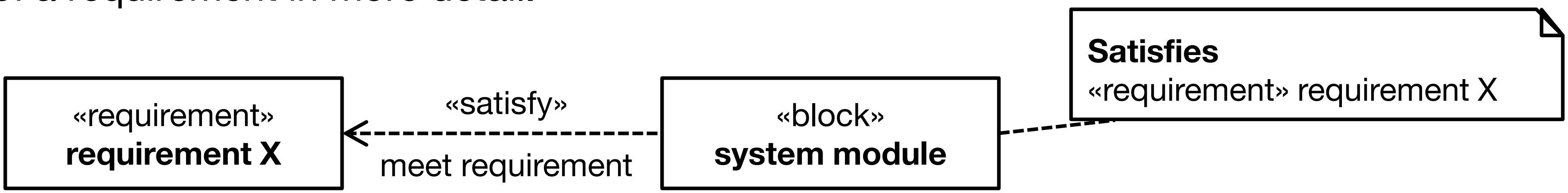
table [Requirement] Stempelprozess

id	name	text
1	stamping process	"The workpiece must be stamped for 10 s with a force of at least 10N”
2	stamping force	“The stamping force must be greater than 10 N”
3	stamping time	“The stamping process must take more than 10s”

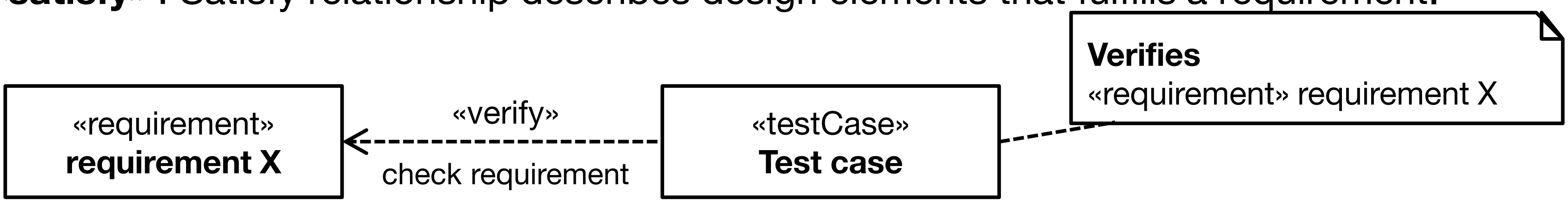
Requirement diagram – Relationships between requirements



«refine» : Refine relationship describes that a model element describes the properties of a requirement in more detail.

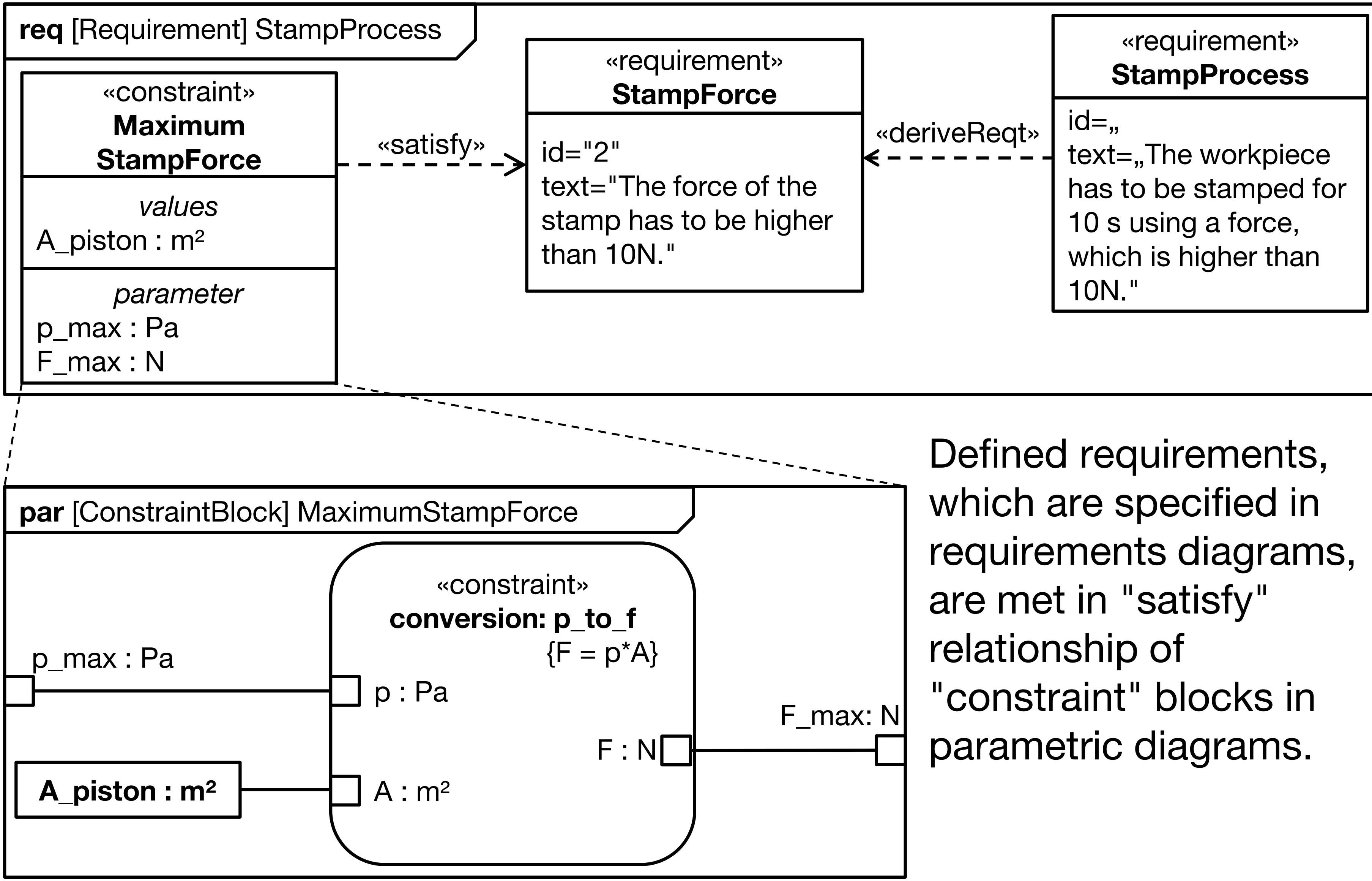


«satisfy» : Satisfy relationship describes design elements that fulfills a requirement.



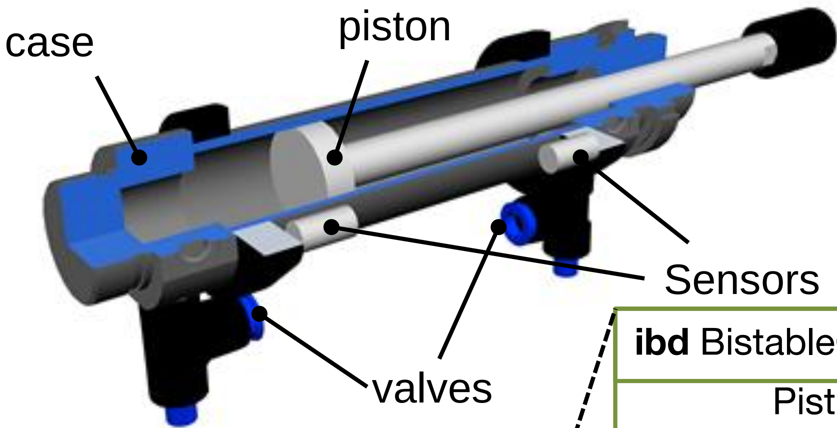
«verify» : Verify relationship combines a test case with the requirement that is checked by the test case.

Relationship between requirements diagram and parametric diagram

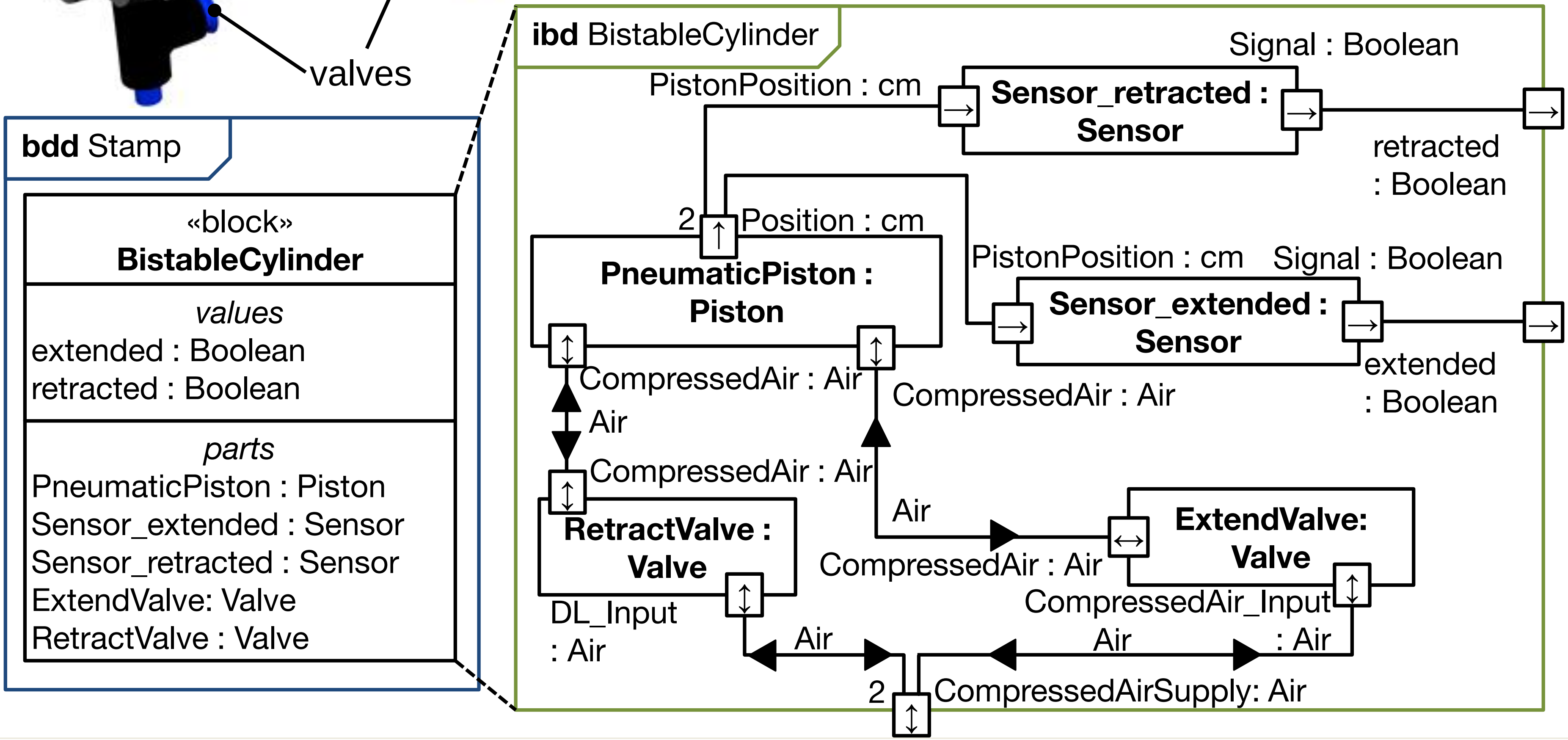


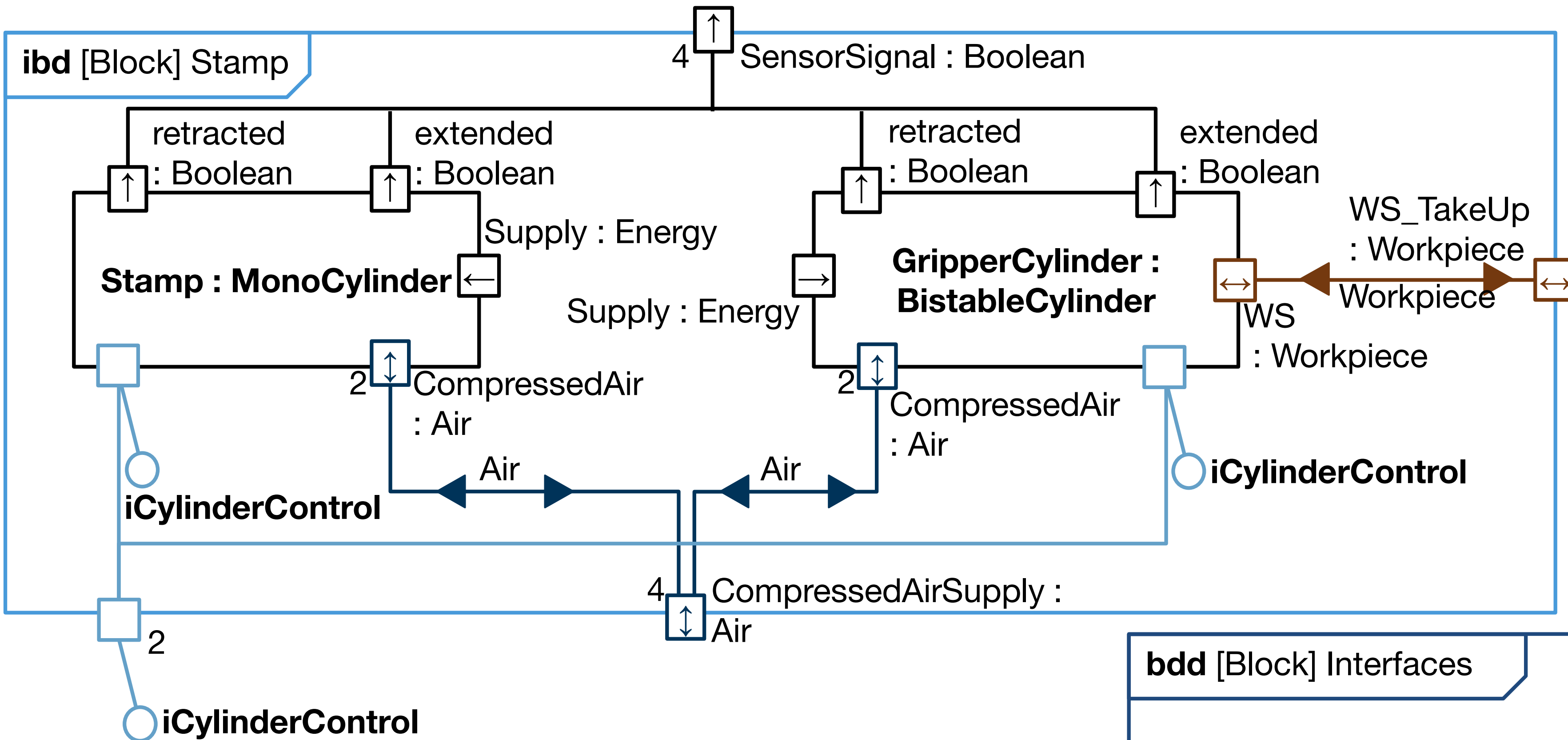
Defined requirements, which are specified in requirements diagrams, are met in "satisfy" relationship of "constraint" blocks in parametric diagrams.

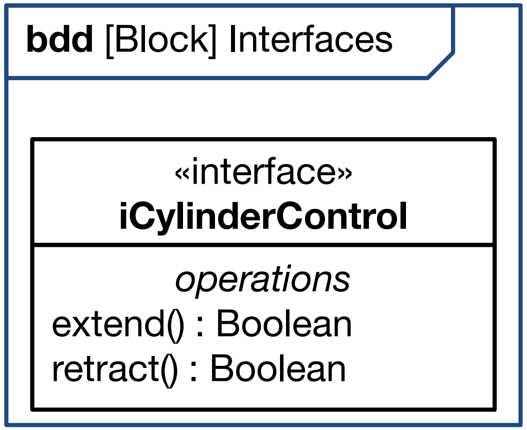
Relationship between block definition diagram (BDD) and internal block diagram (IBD)



The relationships between parts of a block defined in the BDD are represented in the IBD.

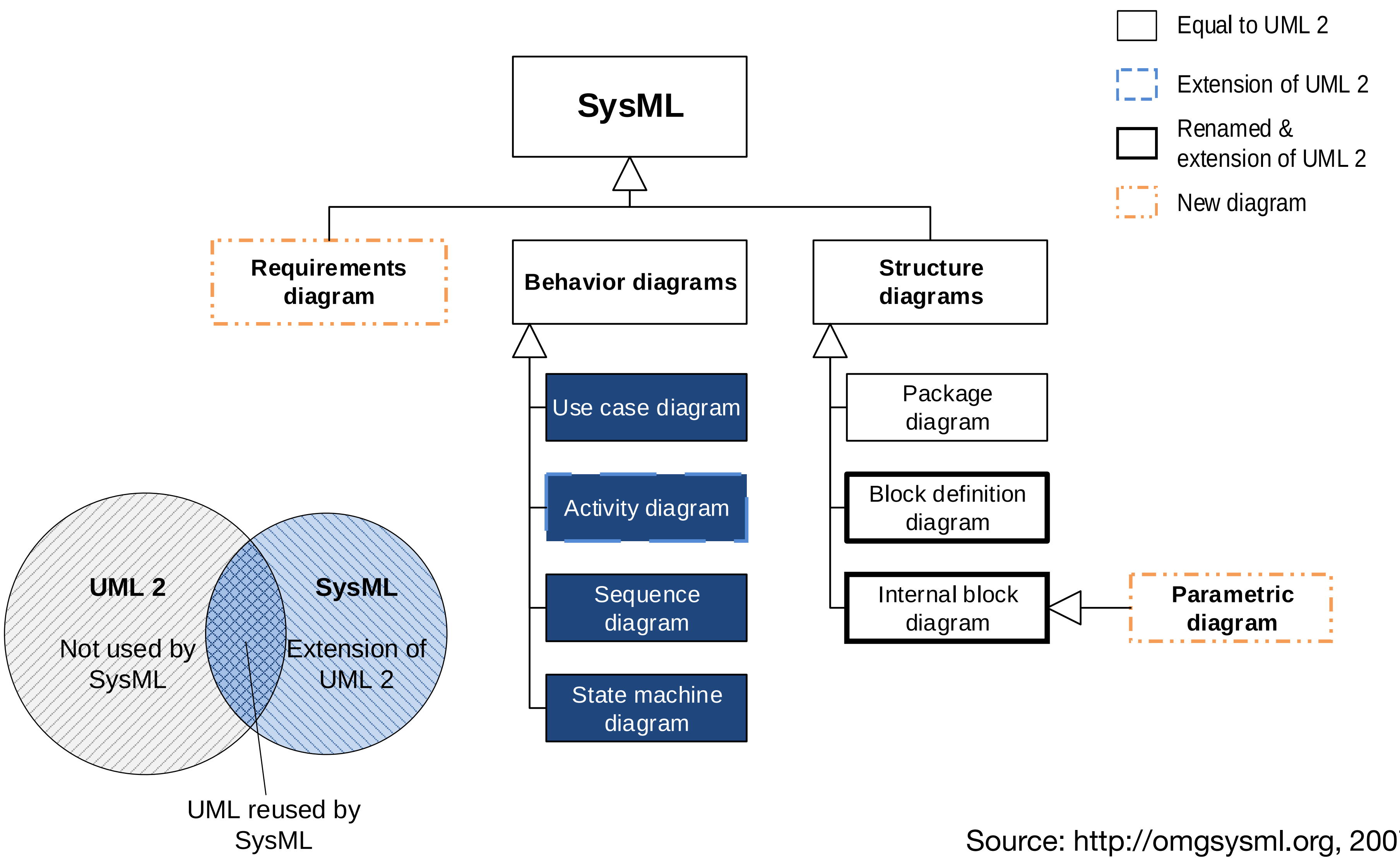






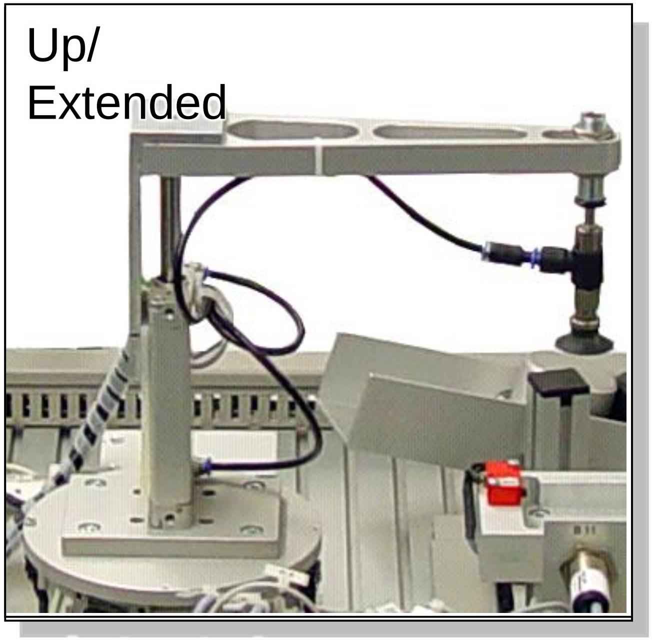
- Modelling power is neglected to retain the overview
- Air and workpiece are concrete objects, which are transmitted

Construction of SysML diagrams



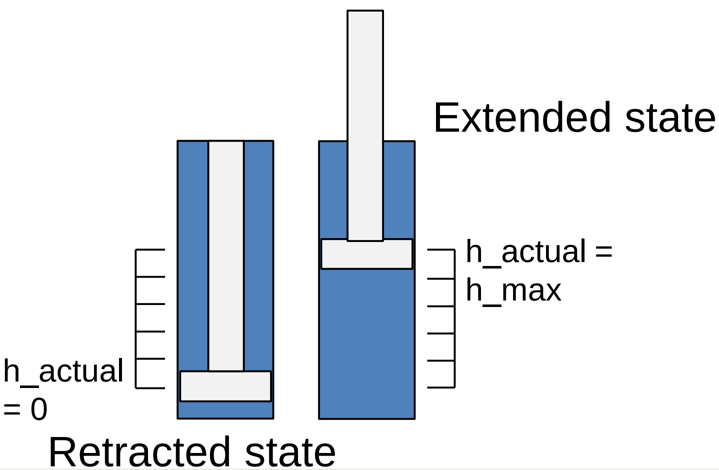
Source: http://omgsysml.org, 2007

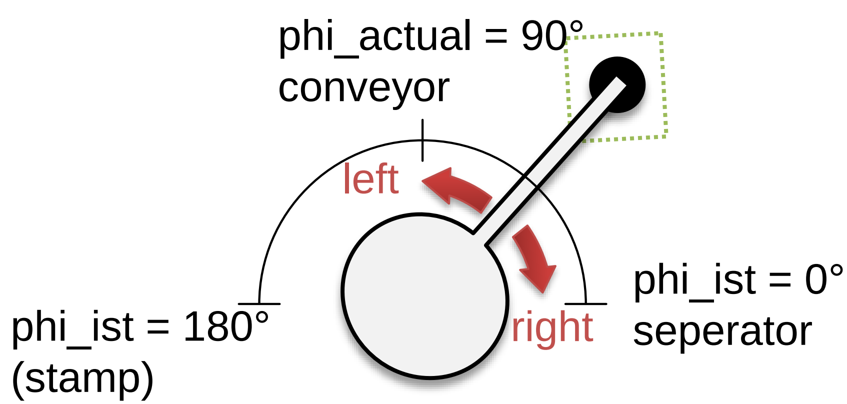
State-based behavior– Important boundary conditions to model the crane



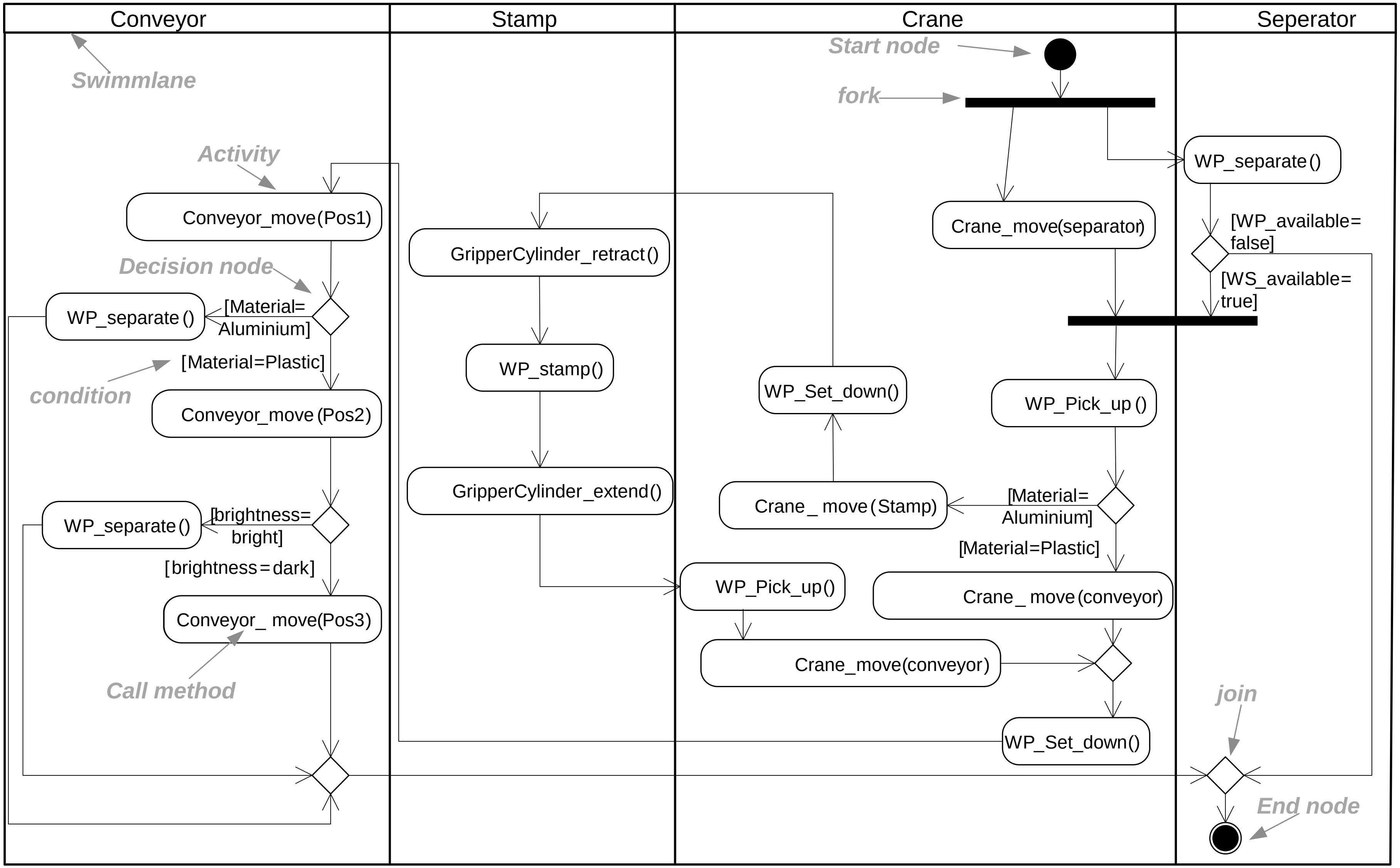
Down/retracted

- Crane is able to move to retracted and extended positions
- A pneumatic cylinder is used to extend the crane



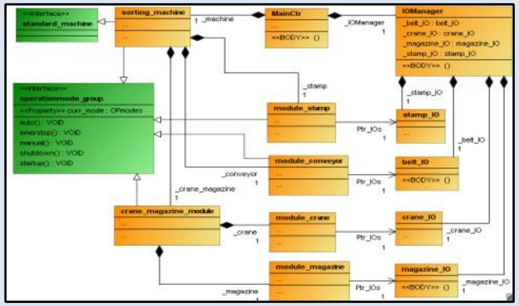


Behavior model: Activity diagram
Application example: Sorting process

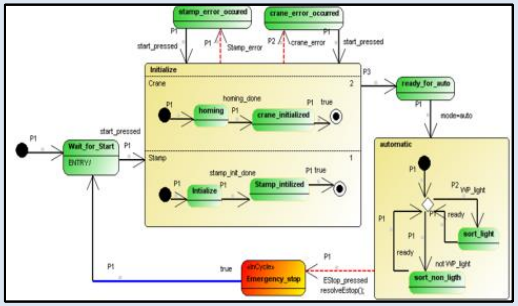


UML-based Implementation of IEC 61131-3 Code with SysML expansion for software agents

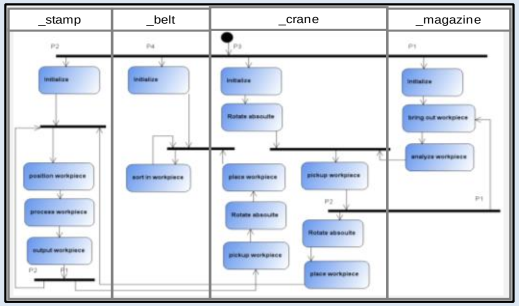
UML-Plugin for CODESYS V3



Class diagram

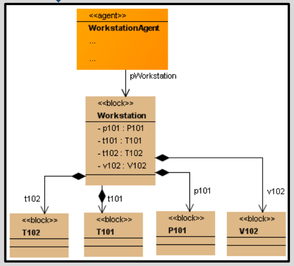


State diagram

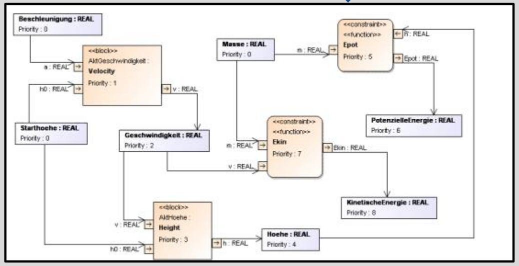


Activity diagram

Code generation for IEC 61131-3 (bidirectional synchronization)



Block definition diagram



Parametric diagram

SysML-Plugin „KREA“ for TwinCAT 3 (CODESYS Family)

Which kinds of knowledge can be modelled with SysML?

Use cases (use case diagram)

Parametric relations

Interdisciplinary relations, especially internal structures of (mechatronic) systems

How can this knowledge be used during runtime?

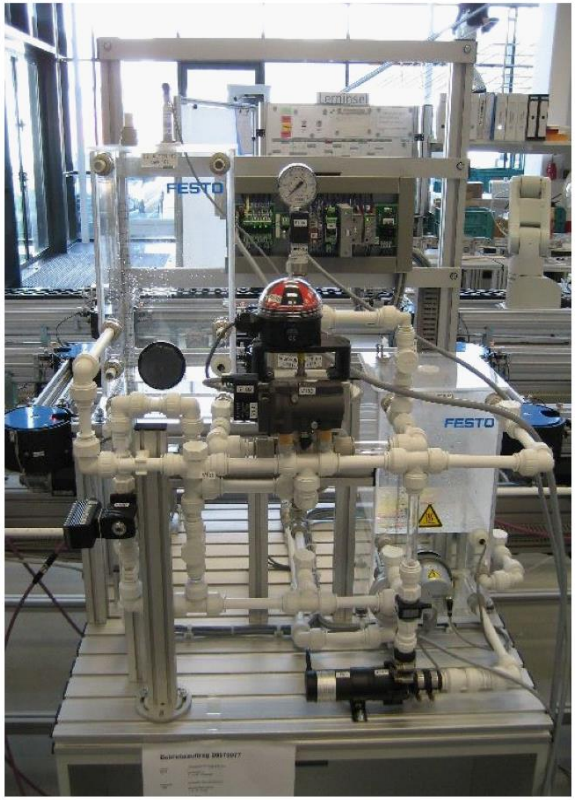
Automatic model transformation/ Code generation/ „Model is code“

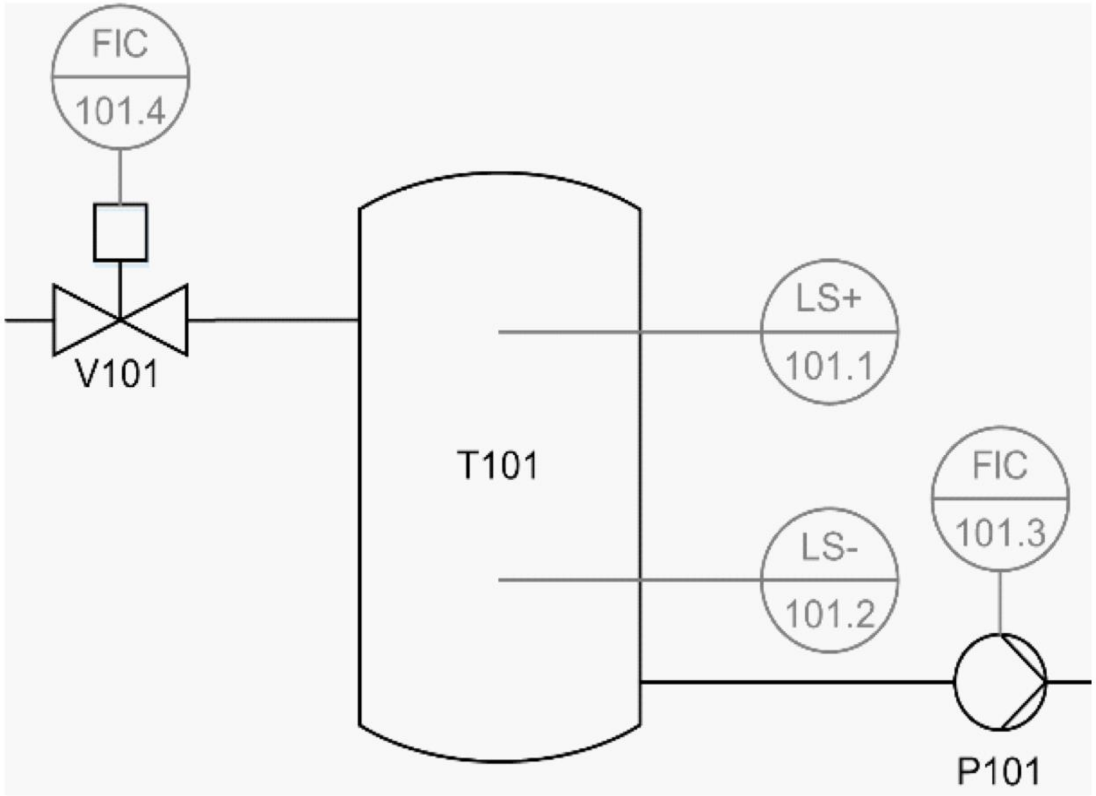
What is missing?

Automatic processing

Rules of consistency of the model (keyword „boundary conditions“)

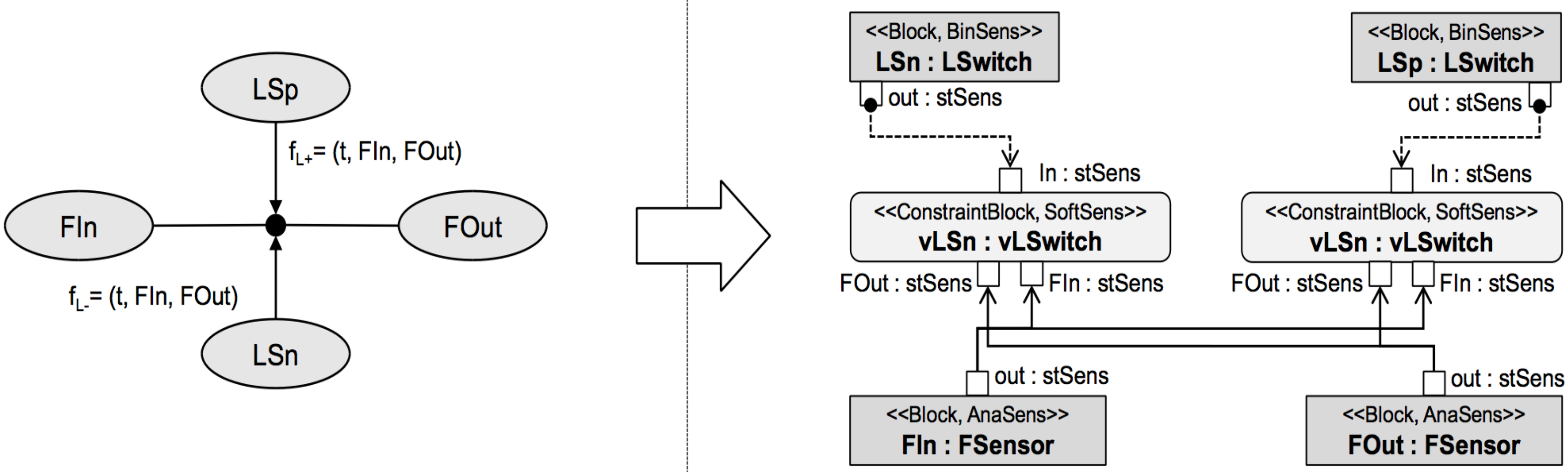
Tank with upper and lower filling level sensors, valve and pump



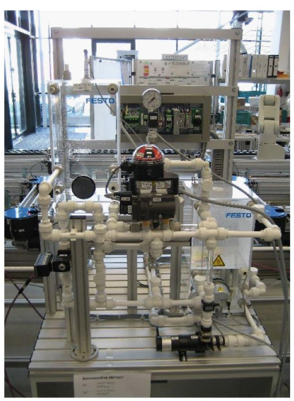


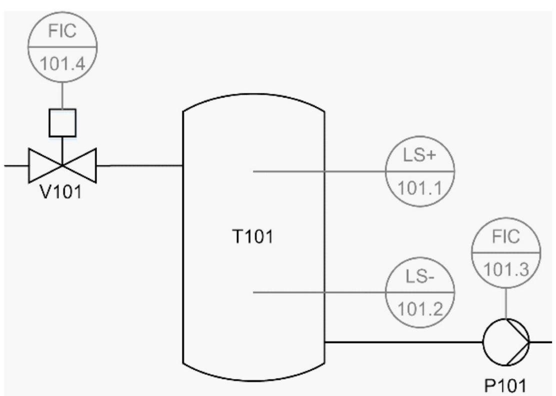
Source: P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche: „Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications“, 2016

Redundancy Model of Tank according to [SWL+13] and mapping to PAR



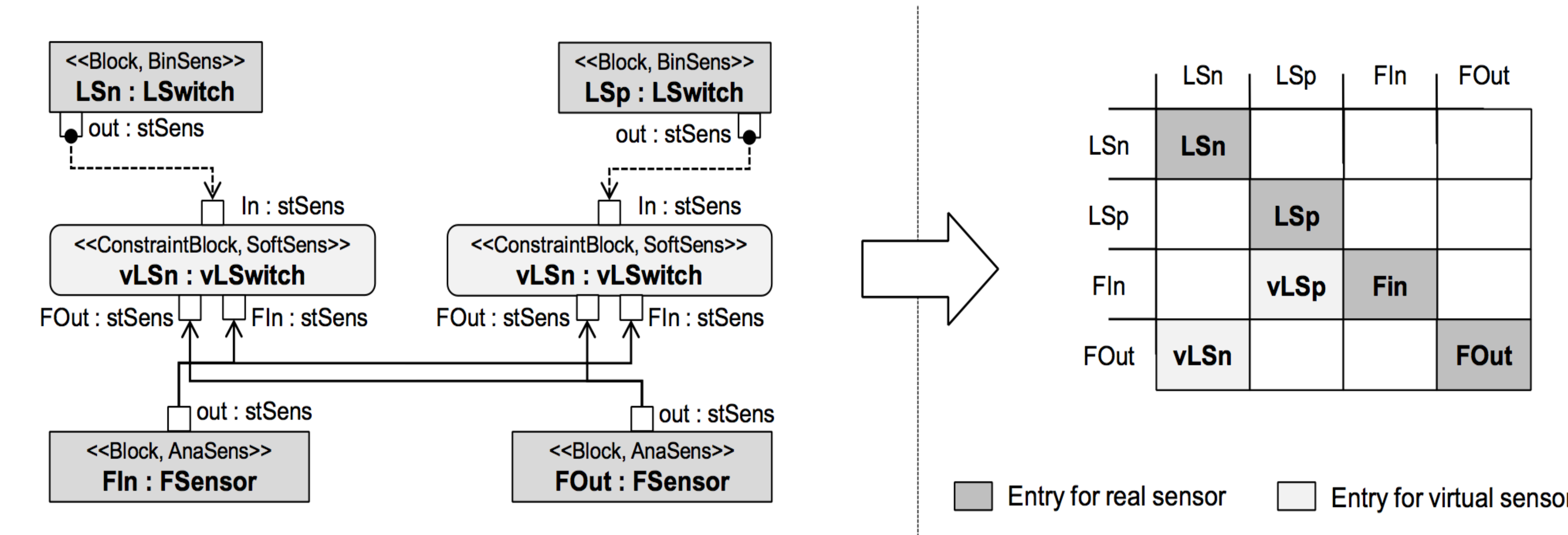
FIn = FIC 101.4   FOut = FIC 101.3   LSn = LS- 101.2   LSp = LS+ 101.1





Source: P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche: „Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications“, 2016

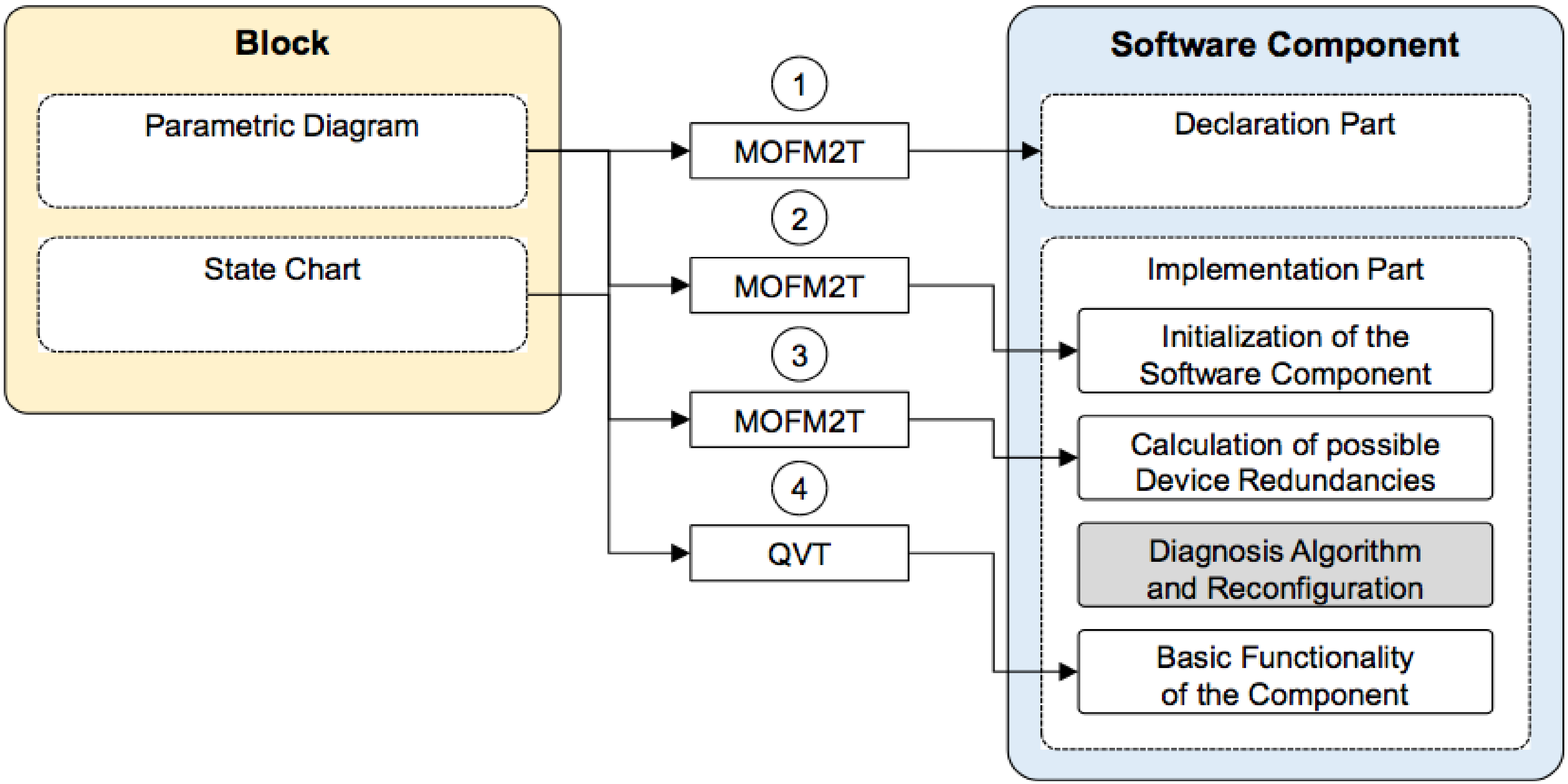
Mapping of PAR onto initialization of the redundancy matrix



FIn = FIC 101.4 FOut = FIC 101.3 LSn = LS- 101.2 LSp = LS+ 101.1

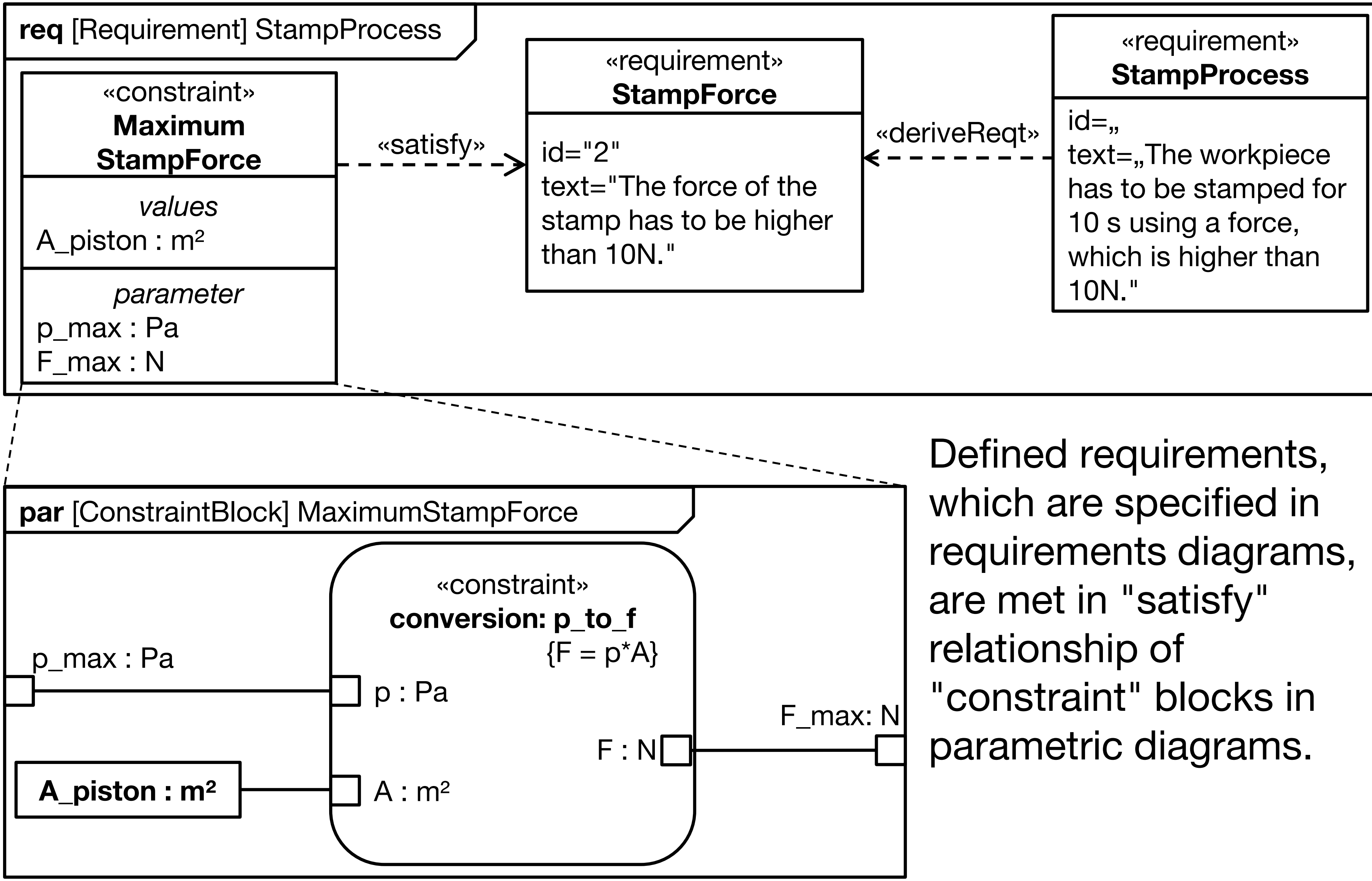
Source: P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche: „Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications", 2016

Agent-based self-aware, self-describing CPPS-module



Source: P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche: „Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications“, 2016

Relationship between requirements diagram and parametric diagram



Defined requirements, which are specified in requirements diagrams, are met in "satisfy" relationship of "constraint" blocks in parametric diagrams.

Exemplary mapping between SysML and IEC 61131-3: block definition diagram

block definition diagram

- Describes a system's structure in code conforming to IEC61131-3
- Visualization and structuring of the system, standard building blocks are stored centrally

Example: energy calculation for the vertical fall

SysML

«block»
EnergyCalculation

values
Acceleration: REAL
StartHigh : REAL
Velocity: REAL
Mass : REAL
High : REAL
PotentialEnergy: REAL
KineticEnergy: REAL

operations

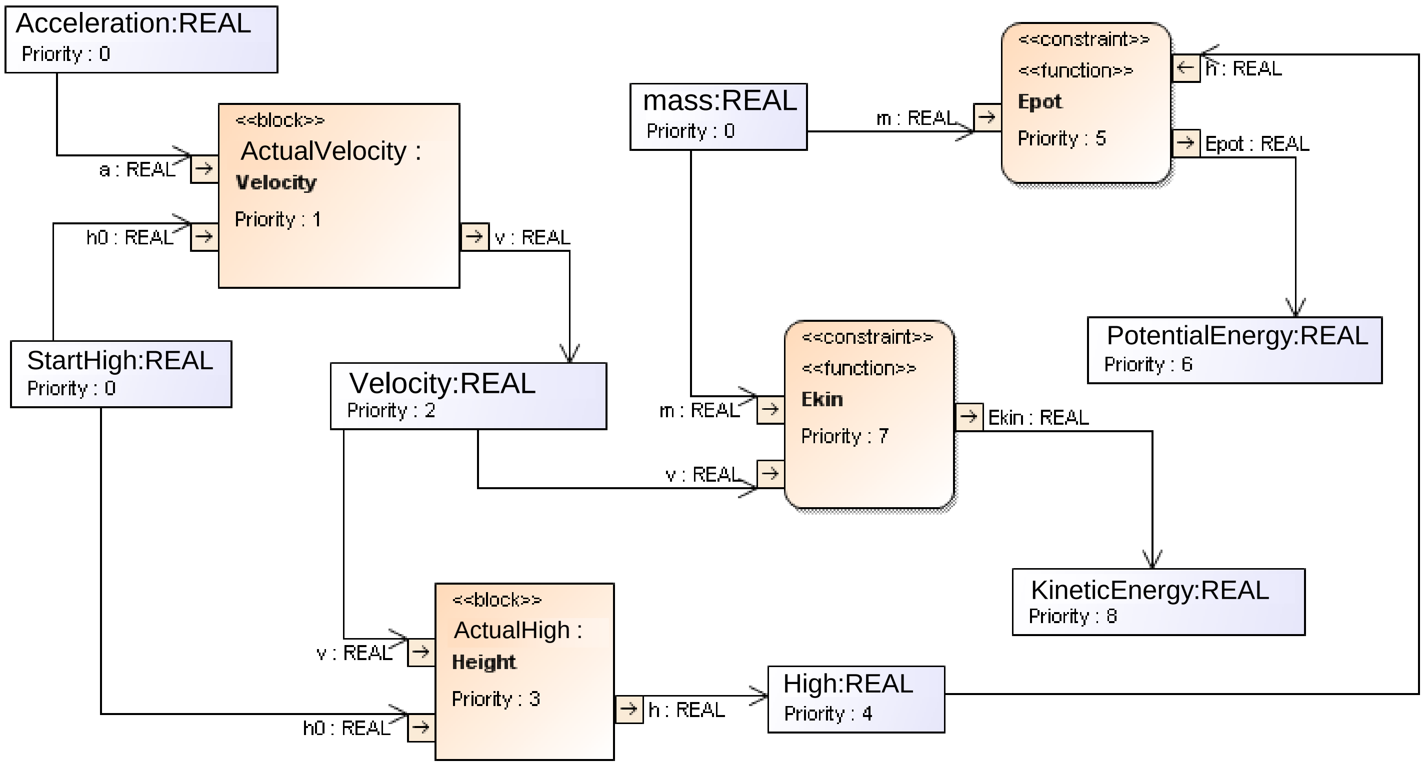
IEC 61131-3
(Block → Function Block)

```
FUNCTION_BLOCK EnergyCalculation
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
      Acceleration: REAL;
      StartHigh : REAL;
      Velocity: REAL;
      Mass : REAL;
      High : REAL;
      PotentialEnergy: REAL;
      KineticEnergy: REAL;
END_VAR
```

Parametric diagram

- Calling of IEC 61131-3 functions and function blocks
- Visualization and structuring of parametric relationships

SysML

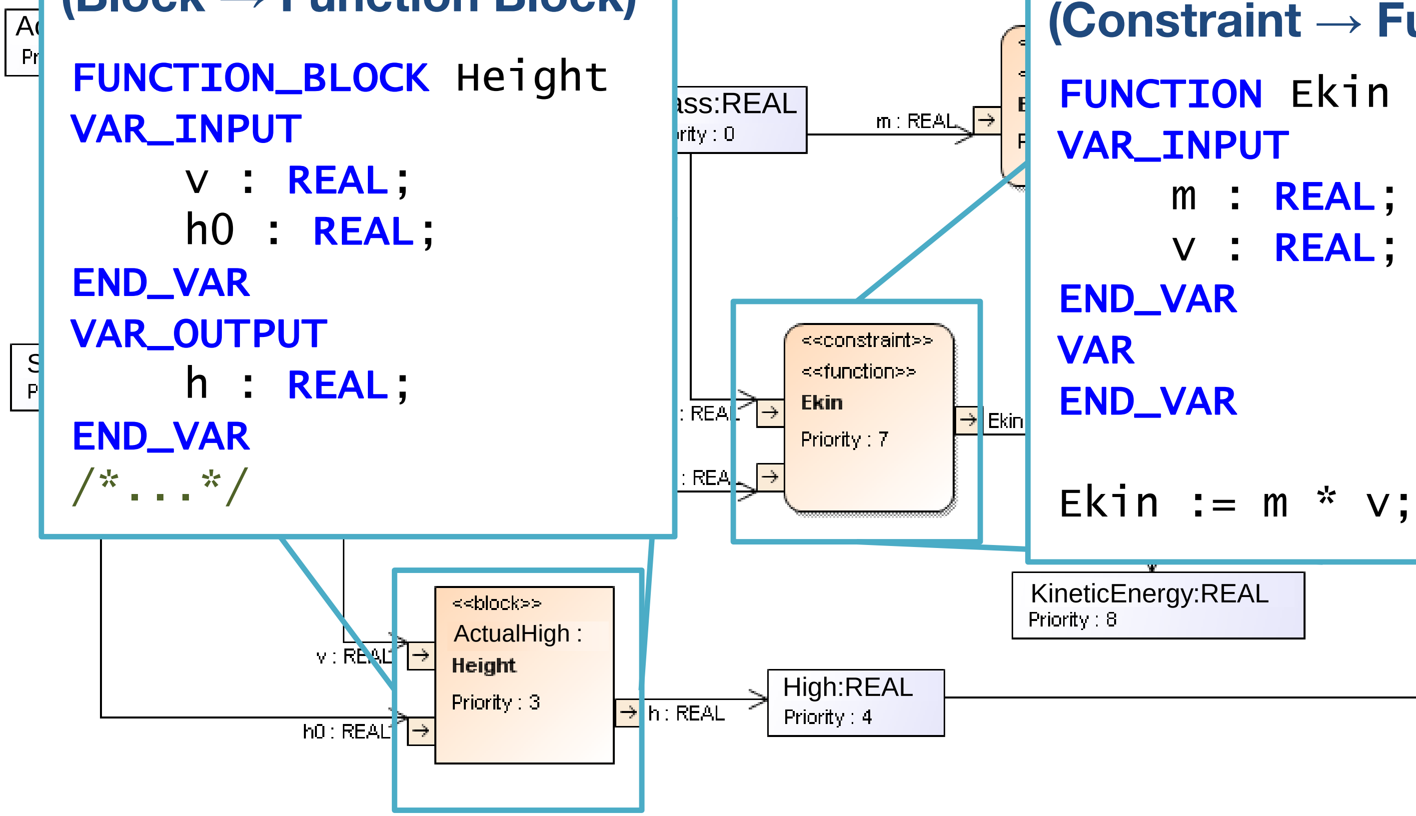


Exemplary mapping between SysML and IEC 61131-3: Parametric diagram

Parametric diagram

- Calling of IEC 61131-3 functions and function blocks
- Visualization and structuring of parametric relationships

SysML



IEC 61131-3 (Block → Function Block)

```
FUNCTION_BLOCK Height
VAR_INPUT
      v : REAL;
      h0 : REAL;
END_VAR
VAR_OUTPUT
      h : REAL;
END_VAR
/*...*/
```

IEC 61131-3 (Constraint → Function)

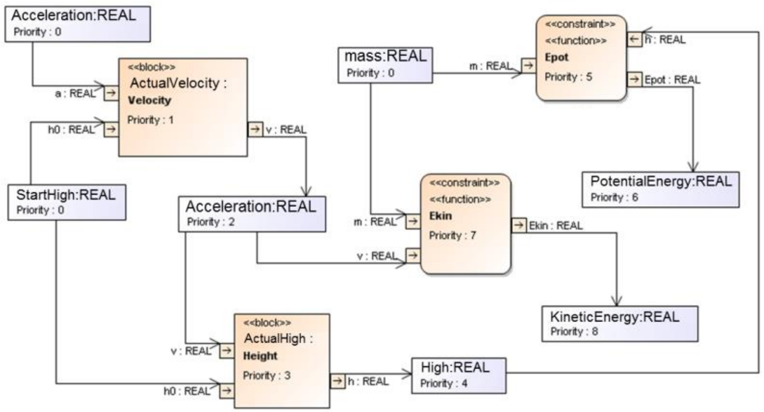
```
FUNCTION Ekin : REAL
VAR_INPUT
      m : REAL;
      v : REAL;
END_VAR
VAR
END_VAR

Ekin := m * v;
```

Parametric diagram

- Calling of IEC 61131-3 functions and function blocks
- Visualization and structuring of parametric relationships

SysML



Problem

What information is missing in pure SysML parametric diagram to automatically generate code?

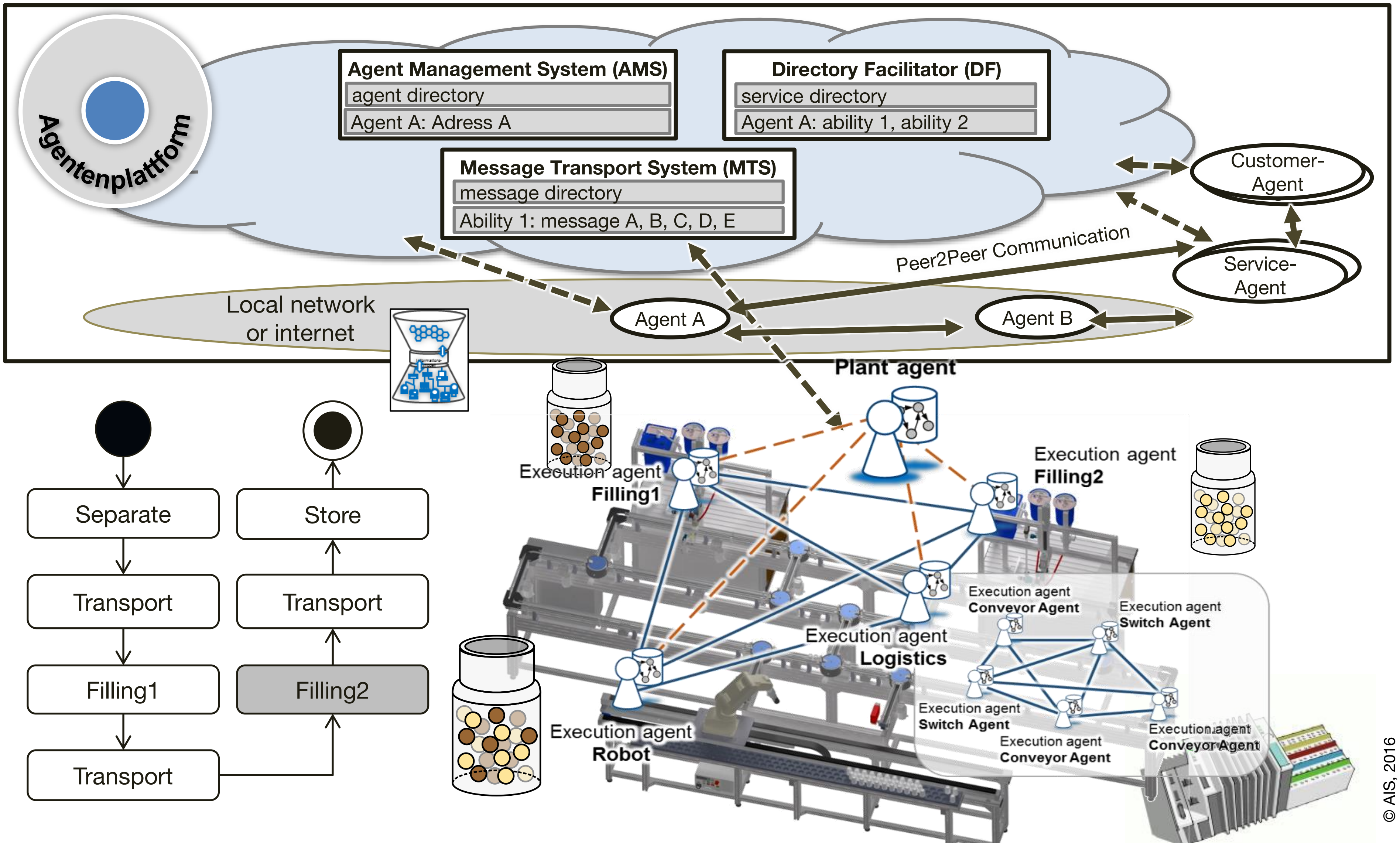
IEC 61131-3

```
ActualSpeed (a := Acceleration, h0 := StartHigh);
Speed:= ActualSpeed.h;
ActualHigh(v := Speed, h0 := StartHigh);
High:= ActualHigh.h;
PotentialEnergy:= Epot(m := Mass, h := high);
KineticEnergy := Ekin(m := Mass, v := Speed);
```

Advanced Aspects on Model-driven Engineering and implementation of field level agents

1. Introduction of TUM and AIS
2. Field level agents in automation
3. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
4. Use case: Sensor reconfiguration modeled with SysML and code generation from SysML Model to PLC‘s
5. **Agents‘ knowledge of valid combinations or sequences modeled with OCL**
6. Metrics for adaptivity
7. Conclusion and future work

Self-adapatation of an CPPS



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.

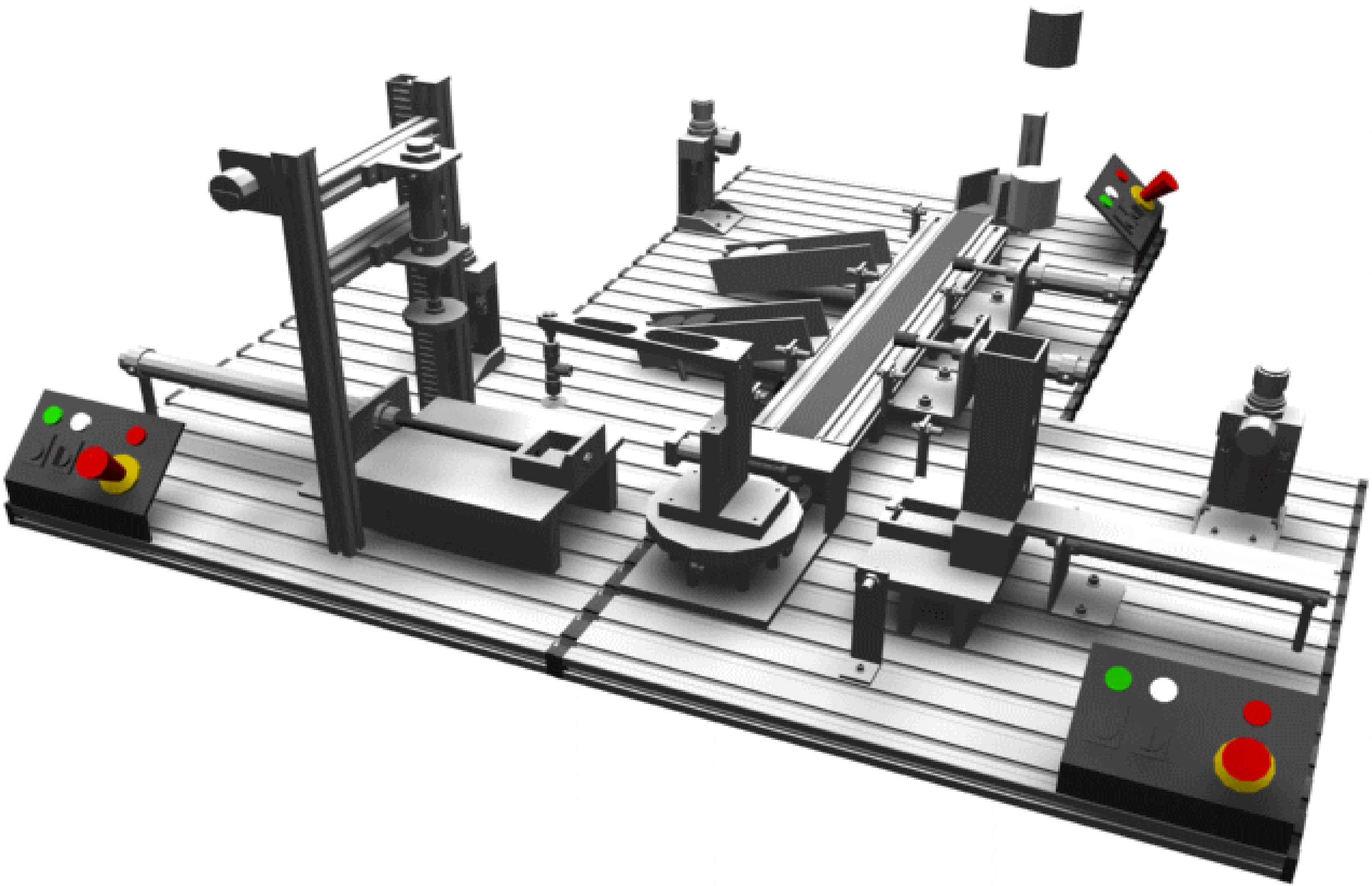
OCL as a formal language

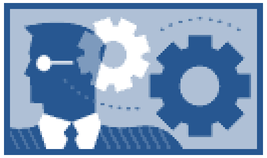
- Software language to specify conditions for UML
- Easy to read
- Pure expression language, no change in the original model
- No programming language, i.e. aspecially
  - No formulation og program logic or control flow
- Typed language
  - Each expression in OCL has a particular type
  - Each OCL expression must use the correct type (e.g. no comparisson of strings and integers)
  - Status of the object is not changed during the validation

Application of OCL: Specification of

- Invariants in class diagrams
- Pre- and postcondotion for
- Conditions in sequence and state diagrams
- Condition of the UML metamodel

Introduction of the small lab scale production system pick-and-place-unit (PPU)

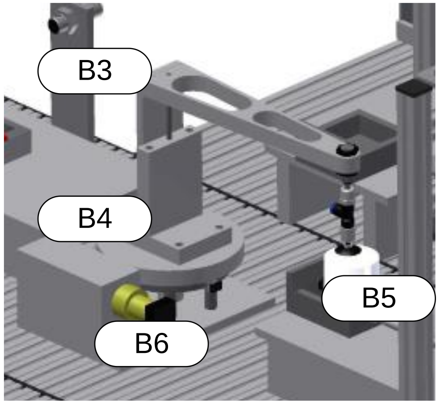


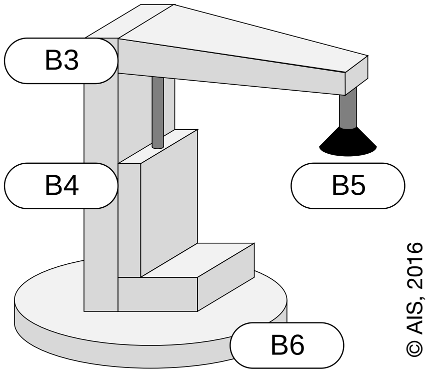


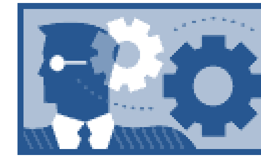
Formulate the following conditions for the crane:

- Because of construction conditions the crane is only allowed to move in a rotation angle of $0° < \varphi \leq 360°$.
- The pre- and postconditions at the bearing for the collection of material by the crane (Kran::MaterialAufnehmen()) are:
  - **Precondition**: Crane unloaded, lowered, at horizontal bearing position ($\varphi = 90°$)
  - **Postcondition**: Crane unloaded, lowered, at horizontal bearing position ($\varphi = 90°$)
- The pre- and postconditions for the rotation of an angle X (Kran::Drehen(x)) of the crane are:
  - **Precondition:** Crane isn‘t allowed to leave the angle range
  - **Postcondition:** Crane is at a new position

Sensor	Beschreibung	Data type
B3	Lift cylinder at top position	Boolean
B4	Lift cylinder at bottom position	Boolean
B5	Vacuum gripper loaded	Boolean
B6	Rotation angle of the rotary base	Integer







Formulate the following conditions for the crane:

- Because of construction conditions the crane is only allowed to move in a rotation angle of $0° < \varphi \leq 360°$.

```
context Kran
inv: B6 > 0 and B6 <= 360
```

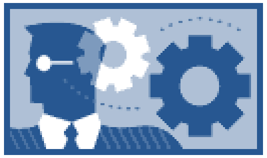
- The pre- and postconditions at the bearing for the collection of material by the crane (Kran::MaterialAufnehmen()) are:
  - **Precondition**: Crane unloaded, lowered, at horizontal bearing position ($\varphi = 90°$)
  - **Postcondition**: Crane unloaded, lowered, at horizontal bearing position ($\varphi = 90°$)

```
context Kran::MaterialAufnehmen()
pre:   not B5 and not B3 and B4 and B6 = 90
post:  B5 and not B3 and B4 and B6 = 90
```

- The pre- and postconditions for the rotation of an angle X (Kran::Drehen(x)) of the crane are:
  - **Precondition:** Crane isn‘t allowed to leave the angle range
  - **Postcondition:** Crane is at a new position

```
context Kran::Drehen(x)
pre:   B6 + x > 0 and B6 + x <= 360
post:  B6 = B6@pre + x
```

Sensor	Beschreibung	Datentyp
B3	Hebezylinder an oberer Position	Boolean
B4	Hebezylinder an unterer Position	Boolean
B5	Vakuumgreifer beladen	Boolean
B6	Drehwinkel der Drehbasis	Integer



How can assumptions in a system model be considered related to its implementation?

- Model refining with additional assurances
- Invariants limit the allowable state space of implementations
- Assurances (pre- and postconditions) limit possible state transitions of implementations

In this context, what is the OCL?

- OCL is a specified language to formulate logical formulas
- Necessary assurances can be integrated by OCL
- UML in combination with OCL enables detailed system models

What information does a model inspector need to test the requirements on an automated model?

- A system model describes structur and behaviour of a system (e.g. UML/OCL)
- A formal description of the requirements

In this context, what is the temporal logic for?

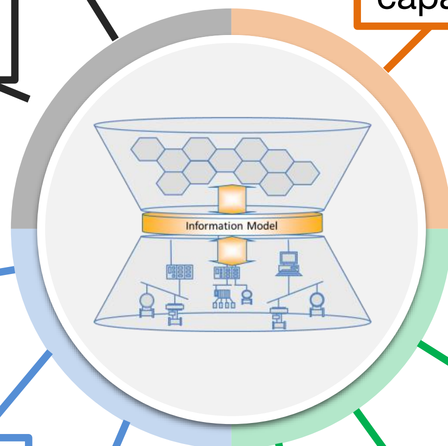
- Temporale logic enables the formulation of qualitative and temporal properties, e.g. „always“ or „sometimes“
- Temporal logic can be used to describe the requirements for system behaviour

Characteristics of Cyber-Physical Production Systems (CPPS) – Industrie 4.0

Data processing for humans

Assistance systems for Engineering

Data processing and integration for humans



Architecture models (reference architecture)

Architecture models (reference architecture) 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

Flexible production units, **adaptable** to modified product requirements, allow also structural changes

Description of product and operating resources, e.g. ontology, for independent analysis, presentation, organization and execution of a production process

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)

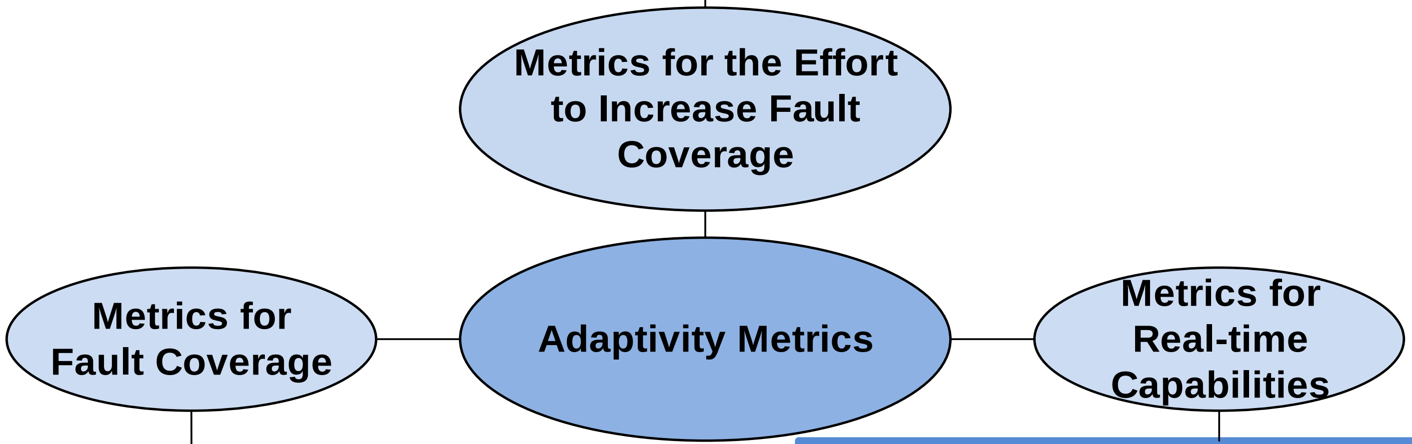
Source: B. Vogel-Heuser, G. Bayrak, U. Frank: Forschungsfragen in "Produktautomatisierung der Zukunft". acatech Materialien. 2012.

Advanced Aspects on Model-driven Engineering and implementation of field level agents

1. Introduction of TUM and AIS
2. Field level agents in automation
3. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
4. Use case: Sensor reconfiguration modeled with SysML and code generation from SysML Model to PLC‘s
5. Agents‘ knowledge of valid combinations or sequences modeled with OCL
6. **Metrics for adaptivity**
7. Conclusion and future work

Proposed metrics for adaptivity for aPS

MEICAI: Minimal Effort to Increase Fault Coverage Adaptivity Index
FHAI: Fault Handling Adaptivity Index
$FHAI_{model}$: Model Based Fault Handling Adaptivity Index



BECAI: Basic Event Coverage Adaptivity Index
BENCAI: Basic Event Not Coverage Adaptivity Index
$FLCAI_j$: Fault Level Coverage Adaptivity Index

plcDFAI: PLC-cycles to detect faults
plcIFAI: PLC-cycles to isolate faults
plcSTSAI: PLC-cycles to switch to soft sensor
plcFCLAI: PLC fault compensation latency adaptivity index

Sc12f: Additional Sensor for Fault Detection, Isolation and Handling

Source: Birgit Vogel-Heuser, Susanne Rösch, Juliane Fischer, Thomas Simon, Sebastian Ulewicz and Jens Folmer: „Fault handling in PLC-based Industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation", 2015

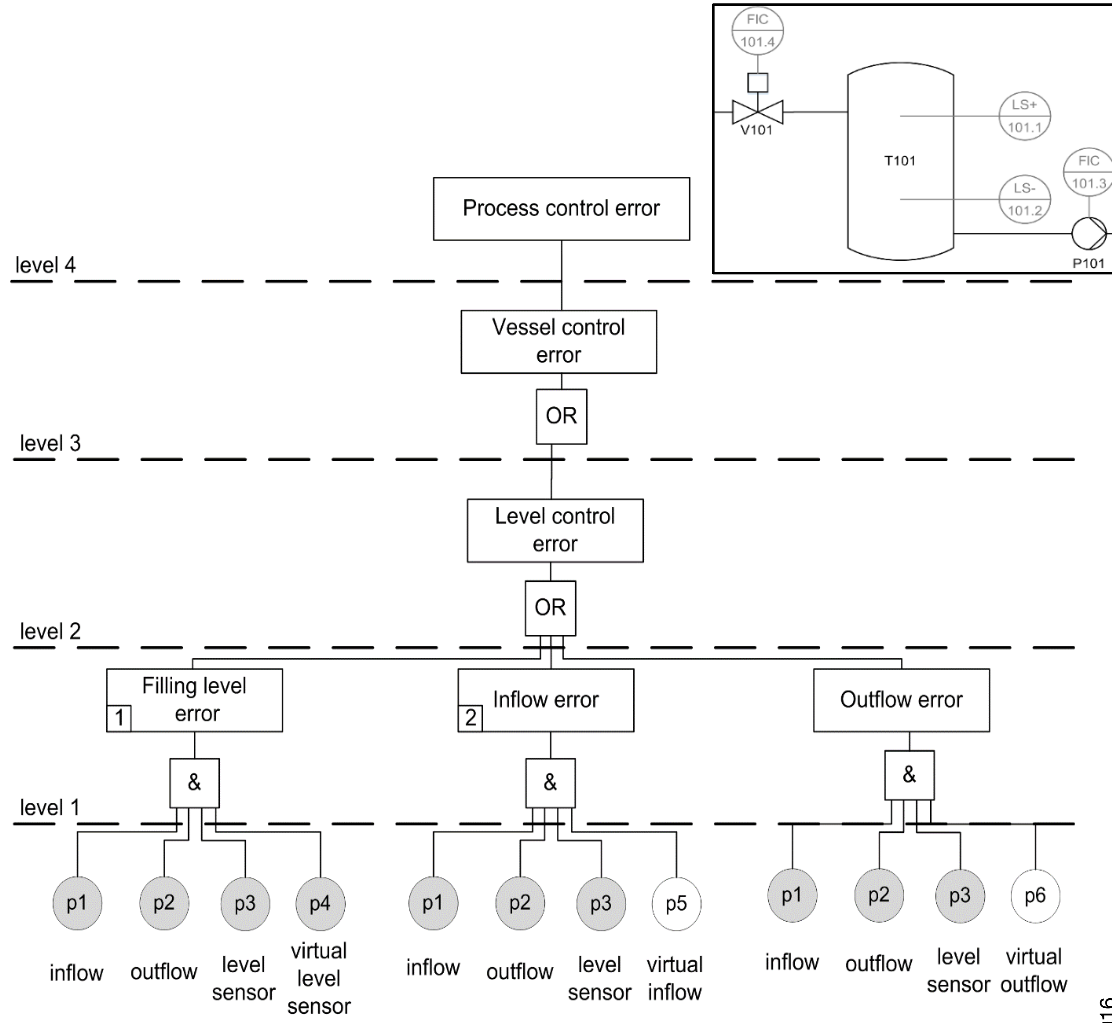
Metrics for real-time capabilities:
For sensore failure (plcDFAI) only cycle time is needed. For the calculated value compared to reacting to the actual sensor value (plcIFAI + plcSTAI) no additional time is needed:
plcFCLAI = plcDFAI + ~~plcIFAI~~ + ~~plcSTAI~~
plcFCLAI = plcDFAI = t_{cycles}

Metrics for the programming effort needed to increase fault coverage:
$FHAI_x$ = software elements

$$FHAI = \frac{FHAI_{new} + FHAI_{adapted} + FHAI_{removed}}{FHAI_{new} + FHAI_{adapted} + FHAI_{removed} + FHAI_{old}}$$

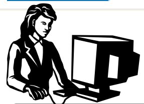
$$= \frac{15+1+0}{15+1+0+35} = \frac{16}{51} = 0.31$$

Therefore to detect and cover a failure of the upper tank filling sensor, 31% of the software code had to be modified or added



Source: Birgit Vogel-Heuser, Susanne Rösch, Juliane Fischer, Thomas Simon, Sebastian Ulewicz and Jens Folmer: „Fault handling in PLC-based Industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation”, 2015

Industry 4.0 - puzzle pieces- open research issues

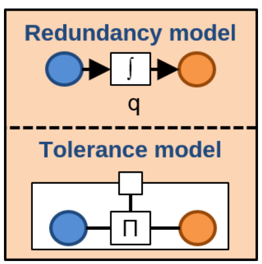


Data processing for humans

Data processing and integration for human

Marketplace of production units

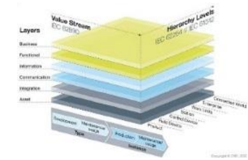
Data consistency about different „stakeholders“ in different engineering phases and crafts

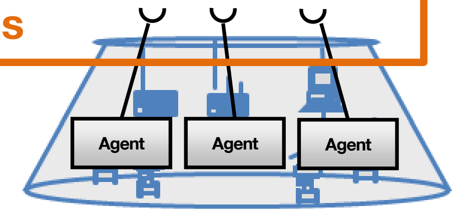


Production units with **inherent capabilities (learning)**

Flexible production units, **adaptable** to modified product requirements, allow also structural changes

Architecture models





Intelligent products and production units

Reconfiguration, recovery, restart of production units

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

Description of product (classification and ontologies) – consistency checking

- ➢ Metrics have to be adapted / further developed for benchmarking aPS designs and operation behavior regarding Industry 4.0

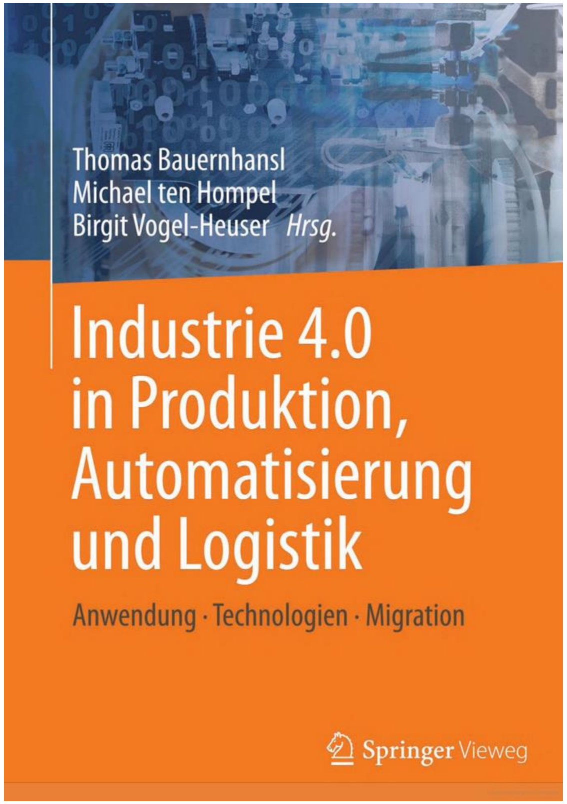
Source: Vogel-Heuser, B.; Rösch, S.; Fischer, J.; Simon, T.; Ulewicz, S.; Folmer, J.: *Fault handling in PLC-based Industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation*. In: Journal of Software Engineering and Applications, Vol. 9, No. 1, 2016, PP. 1-43.

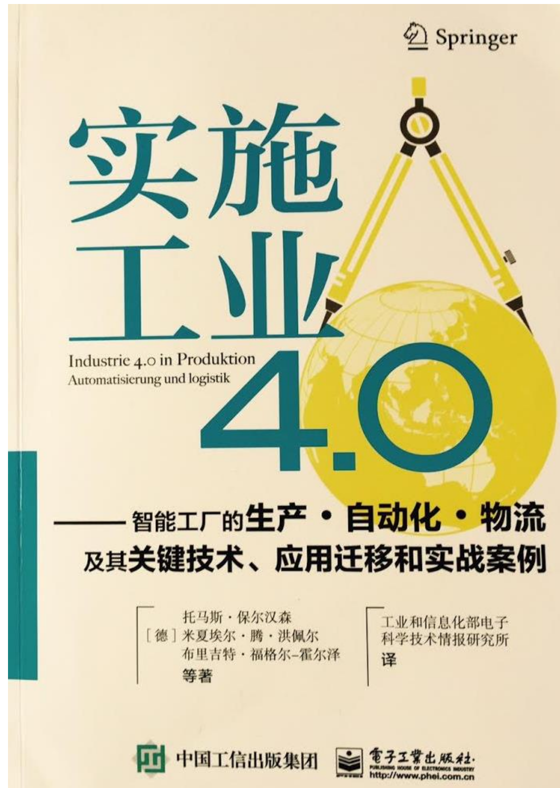
Program of the agent expert forum

27th September 2016	
13:00	**Welcome and introduction** Prof. Dr.-Ing. Birgit Vogel-Heuser, Technical University of Munich
Session 1	
13:15	**A modeling approach for ad hoc reconfiguration of industrial agents** Prof. Dr.-Ing. Birgit Vogel-Heuser Technical University of Munich
13:45	**Concept for an agent-based industry 4.0 component** Dipl.-Ing. Daria Ryashentseva and Ambra Calá, M.Sc. Otto-von-Guericke University Magdeburg
14:15	**Dezentrale Produktionssysteme in der Pharmabranche - ein Fallbeispiel** Dr.-Ing. Michael Rausch Harro Höfliger
14:45	**Komponenten-orientiertes Engineering** Dipl.-Ing (FH) Georg Scharf Bachmann electronic
15:15	**Coffee break**
Session 2	
15:45	**Agents everywhere! Utilizing agent-based approaches from the supply chain down to machinery and components** Prof. Dr.-Ing. Bernd Hellingrath University of Münster
16:15	**Challenges in the design, deployment and assessment of intelligent cyber-physical automation solutions** Luis Ribeiro, PhD Linköping University
16:45	**Agent-based Modelling and Simulation** Prof. Paulo Leitão Polytechnical Institute of Bragança
17:15	**Agent-based assistance system in automation technology - Positioning and Outlook** Prof. Dr.-Ing. Michael Weyrich University of Stuttgart
17:50	**Bus ride to Oktoberfest (Munich)**
18:30	**Evening event**

28th September 2016	
Session 3	
09:00	**Design of CPS-based Digital-Twin PHM in Future Industrial Systems** Prof. Dr. Jay Lee University of Cincinnati
09:30	**Engineering environment (i.e. tools and methods) for component-based manufacturing systems** Dr. Bilal Ahmad University of Warwick
10:00	**A multi agent system approach for reconfiguration of automated material flow modules during runtime** Daniel Regulin, M.Sc Technical University of Munich
10:30	**Real-time software agents for implementing cyber-physical production systems** Dipl.-Inf. Sebastian Theiss, Dresden University of Technology
11:00	**Guided tour of the institute AIS**
11:30	**Coffee break**
Session 4	
11:45	**Engineering agent based multimodal energy grid control solutions** Dipl.-Ing. Tobias Linnenberg Helmut Schmidt University Hamburg
12:15	**Engineering with Software Agents to Simulate Multiphysics Problems in Parallel** Desirée Vögeli, M.Sc. University of Stuttgart
12:45	**Industrial agents: their potentials and when they will actually take control over factories?** Dr. Andrei Lobov Tampere University of Technology
13:15	**Farewell** Prof. Dr.-Ing. Birgit Vogel-Heuser Technical University of Munich
13:30	**Lunch**

Print to appear Oct. 2016





Authors: Birgit Vogel-Heuser, Thomas Bauernhansl, Michael ten Hompel
Handbuch available online:
http://link.springer.com/referencework/10.1007%2F978-3-662-45537-1

http://i40d.ais.mw.tum.de

Thank you for your attention.

Birgit-Vogel-Heuser

Institute of Automation and Information Systems (AIS)
Faculty of Mechanical Engineering, Technische Universität München
Munich, Germany