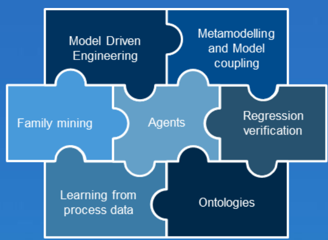
Model-driven Engineering and implementation of field level agents with IEC 61131-3



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

Ordinaria
Automation and Information Systems (AIS)
Mechanical engineering,
Technische Universität München
www.ais.mw.tum.de; vogel-heuser@tum.de

**Agent Summer School –
INDIN Emden 2017**

Introduction of Technical University of Munich

- **40,124 students**
- **14 faculties**
- **3 Integrative Research Centers**
- **7 Corporate Research Centers**
- **13,741 female students**
- **10,103 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	5,448	611	1,136

Automation
and Information Systems
Technical University of Munich

Institute of Automation and Information Systems (AIS)

Technical University of Munich

- The leading university in mechanical and electrical engineering in Germany

Rankings 2017

- Technical University of Munich:
  - 47th at the Academic Ranking of World Universities (Shanghai-Ranking)
  - 64th at the QS World University Ranking
- Faculty of Maschinenwesen:
  - 24th 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

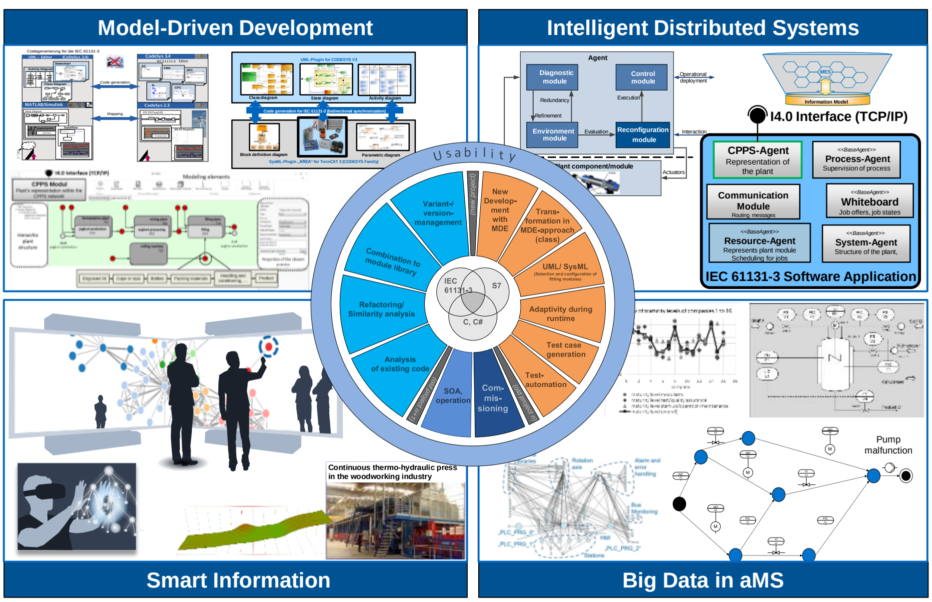
- approx. **20** PhD students
- **9** technicians, trainees (software engineering)

Automation
and Information Systems
Technical University of Munich

Design For FUTURE

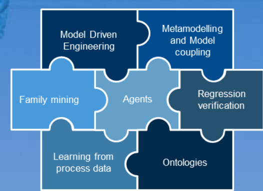
Research Topics
Automation and Information Systems

SFB 768



Model-driven Engineering and implementation of field level agents with IEC61131-3

1. Field level agents in automation
2. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
3. Use case sensor reconfiguration level 0: Model Driven Sensor reconfiguration with SysML – tank and pushers
   - SysML Introduction- Modeling instead of coding
   - OCL and Ontology introduction for knowledge description
4. Use cases in intralogistics
5. Metrics for adaptivity and flexibility
6. Conclusion and future work



Motivation, trends and challenges

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

Automation
and Information Systems
Technical University of Munich

Overall equipment effectiveness (OEE) by self-aware, self-reconfiguring, self-recovering and self-restarting

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 and**
- **self-restarting**

Availabilit y losses

Power losses

Effectiveness loss

Quality losses

© AIS TUM

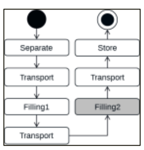
Automation
and Information Systems
Technical University of Munich

Flexibility interdependency triangle of Product, Process, Resource

Process (techn.)

Law authorities:
More energy efficient process

Product



Important Changes are not independent !

Individualized, often changed

Resource (aMS)

Resource change:
Economical conditions
(higher efficiency)

Customer:
Improved product

Input/Signals

Output/Signals

Availability, flexibility

self-aware, self-reconfiguring, self-recovering and self-restarting

© AIS TUM

Automation
and Information Systems
Technical University of Munich

Criteria defining adaptability / flexibility in aPS

Criterion considered — Adaptability / flexibility fulfilled?

Product Process Resource (PPR) interdependance

Field	Criterion	Adaptable	Flexible	Comments
Product	(A1) Product parameters			
	(A2) Product functions			
	(A3) Product structure			
	(A4) Product quality			
	(A5) Lot size			
	(A6) Overall output			
	(A7) Product mix.			
Process	(B1) Production technology			
	(B2) Environmental impact			
	(B3) Number of process steps			
	(B4) Process control			
Resource	(C1) Resource parameters			
	(C2) Resource control			
	(C3) Resource functions			
	(C4) Resource structure			
	(C5) Reachable quality			
	(C6) Number of resources			
	(C7) Resource throughput			

[Source: Lüder, A., Vogel-Heuser, B., Schmidt, N., Prieler, J.: „Metric based modelling of flexibility properties of demonstration plants", 2017]

Automation
and Information Systems
Technical University of Munich

Case study for a simple lab demonstrator

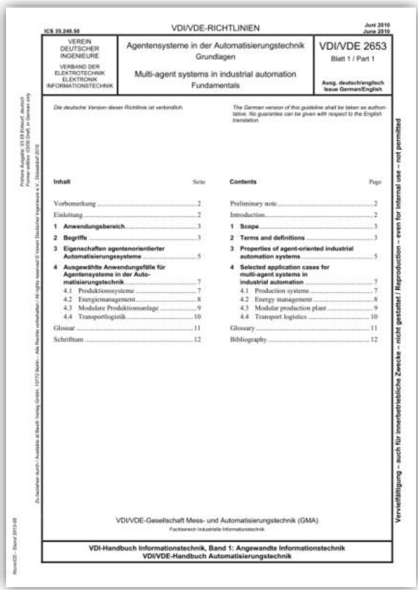
Field	Criterion	Adaptable	Flexible	Comments
Product	(A1) Product parameters	Yes	Yes	Different workpieces (WPs) with varying weights
	(A2) Product functions	No	No	
	(A3) Product structure	Yes	Yes	2 plastic workpiece (WP) types, 3 metal WP types
	(A4) Product quality	No	No	Quality not part of the model
	(A5) Lot size	Yes	Yes	Additional WPs
	(A6) Overall output	Yes	No	Output not modelled so far (what is output in xPPU?)
	(A7) Product mix.	Yes	No	Different WPs with varying weights
Process	(B1) Production technology	Yes	No	New stamping mechanism? -> new stamp necessary
	(B2) Environmental impact	No	No	Is not reflected in PPR model
	(B3) Number of process steps	Yes	No	The number of production steps is modelled in PPR model
	(B4) Process control	No	No	

Field	Criterion	Adaptable	Flexible	Comments
Resource	(C1) Resource parameters	Yes	No	Update of machine tools parameters by PLC programming
	(C2) Resource control	Yes	No	Update of machine tool control by PLC programming
	(C3) Resource functions	Yes	No	Changes in resource functions by PLC programming
	(C4) Resource structure	Yes	No	Change of number of PLC controllers
	(C5) Reachable quality	No	No	Quality not part of the model
	(C6) Number of resources	Yes	No	Additional slide
	(C7) Resource throughput	Yes	No	Changed processing time related parameters within machine tool control by PLC programming

[Source: Lüder, A., Vogel-Heuser, B., Schmidt, N., Prieler, J.: „Metric based modelling of flexibility properties of demonstration plants", 2017]

Automation
and Information Systems
Technical University of Munich

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	Contents	Page
Vorbemerkung	2	Preliminary note	2
Einleitung	2	Introduction	2
1 Anwendungsbereich	3	1 Scope	3
2 Begriffe	3	2 Terms and definitions	3
3 Eigenschaften agentenorientierter Automatisierungssysteme	5	3 Properties of agent-oriented industrial automation systems	5
4 Ausgewählte Anwendungsfälle für Agentensysteme in der Automatisierungstechnik	7	4 Selected application cases for multi-agent systems in industrial automation	7
4.1 Produktionssysteme	7	4.1 Production systems	7
4.2 Energiemanagement	8	4.2 Energy management	8
4.3 Modulare Produktionsanlage	9	4.3 Modular production plant	9
4.4 Transportlogistik	10	4.4 Transport logistics	10
Glossar	11	Glossary	11
Schrifttum	12	Bibliography	12

VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA)

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

Vervielfältigung – auch für innerbetriebliche Zwecke – nicht gestattet / Reproduction – even for internal use – not permitted

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

Automation
and Information Systems
Technical University of Munich

Revision of the guideline VDI/VDE 2653 – Part 1

- 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", „Microgrids", „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

– **New part 4: metrics**

Automation
and Information Systems
Technical University of Munich

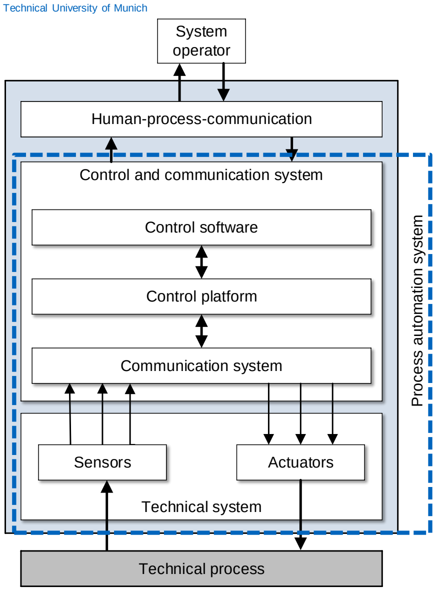
Morphological box for classifying agents- Which agent types are most appropriate for field level agents?

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

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.

Automation
and Information Systems
Technical University of Munich

Technical process, automation system, process automation systems



[Source: Lauber, R.; Göhner, P.: Prozessautomatisierung 1, 1999]

System operator
Controls the process automation system
Process automation system
Human-process-communication
- Visualization for the system operator
- Interface between system operator and process automation system

Automation computer system
- **Control software: Application**
- **Control platform: Electrotechnical hardware/ computer systems**
  - **Programmable Logic Controllers (PLC)**
  - Microcontrollers (µC)
  - Personal Computer (PC) respectively Industry PC (IPC)
  - Process Control System (PCS)

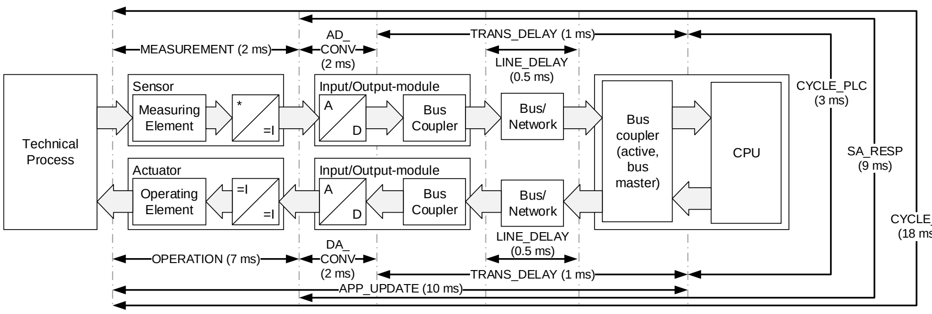
Communication system
- **Bus system**

Technical System (Automation object)
- **Contains additional sensors and actuators**

Technical process
Production process: Transformation of matter, energy or information in order to change physical quantities

Automation
and Information Systems
Technical University of Munich

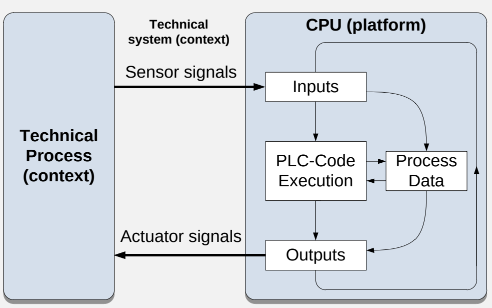
Delay chain of the communication network from measuring to operation tasks



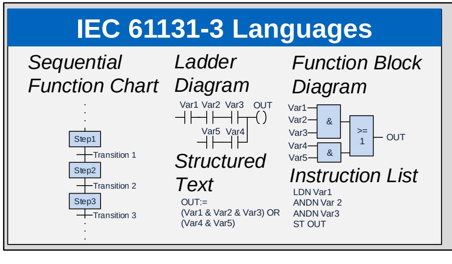
Source: B. Vogel-Heuser, S. Feldmann, T. Werner, C. Diedrich: „Modeling network architecture and time behavior of Distributed Control Systems in industrial plant automation", In: 37th Annual Conference of IEEE Industrial Electronics (IECON), Nov. 2011.

Automation
and Information Systems
Technical University of Munich

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

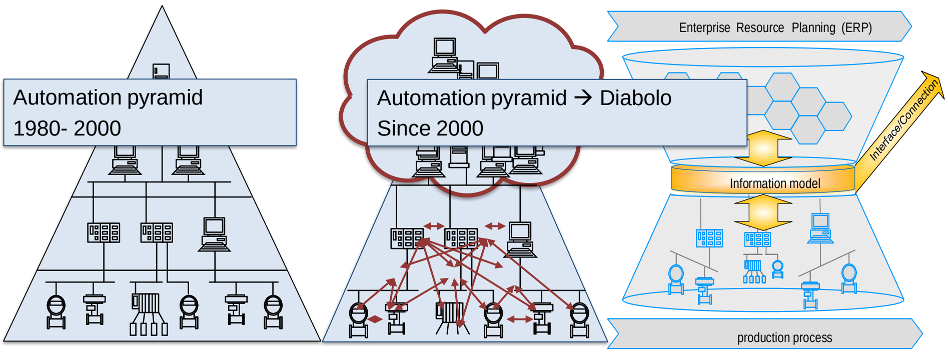


IEC 61131-3 Programming Languages

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

Automation
and Information Systems
Technical University of Munich

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

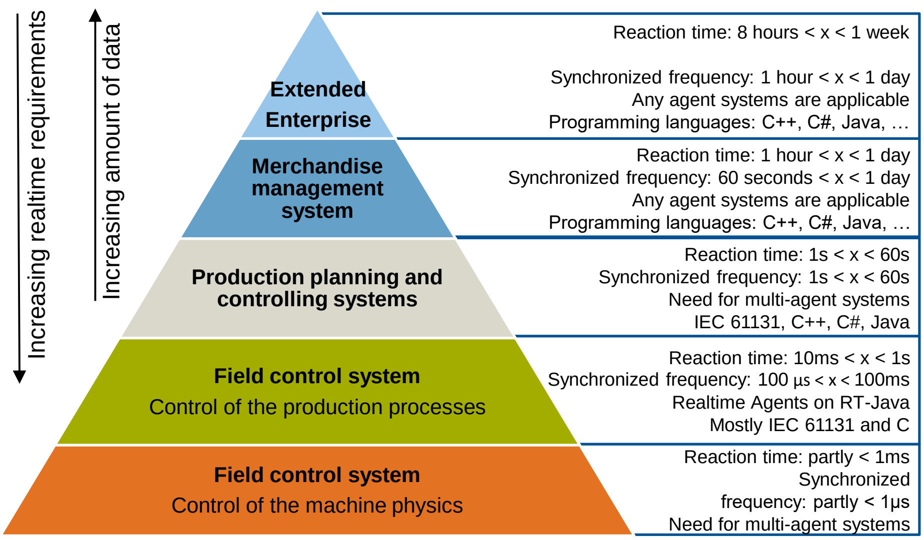
Results

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.

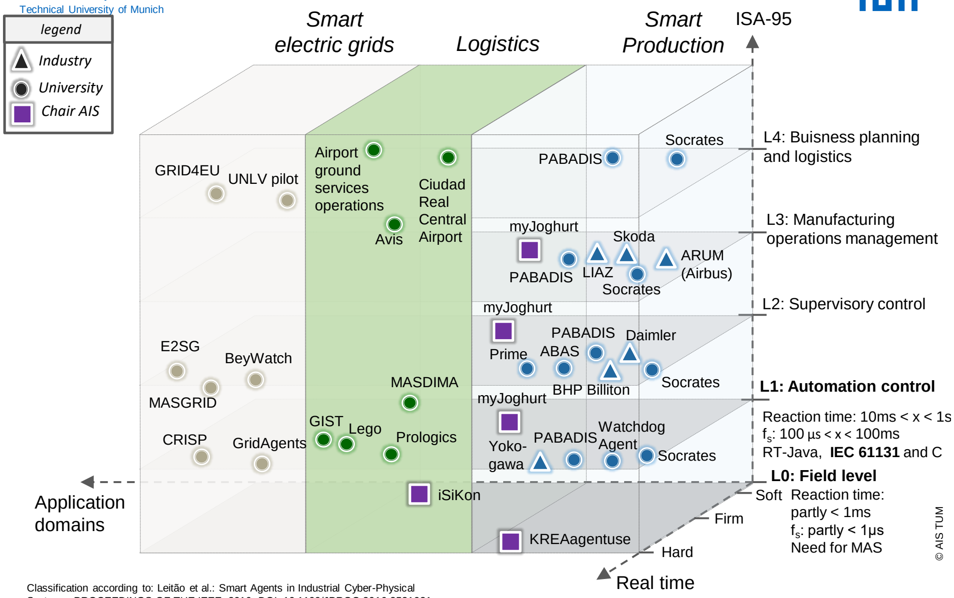
Automation
and Information Systems
Technical University of Munich

Fields of application for agents



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

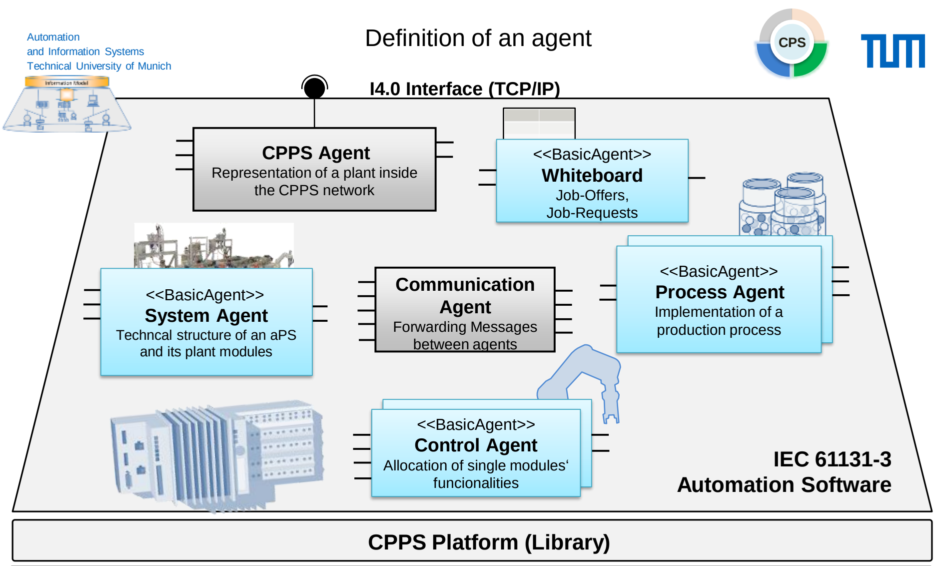
State of the art- agents in CPS/CPPS



Classification according to: Leitão et al.: Smart Agents in Industrial Cyber-Physical Systems, PROCEEDINGS OF THE IEEE, 2016, DOI: 10.1109/JPROC.2016.2521931

Automation
and Information Systems
Technical University of Munich

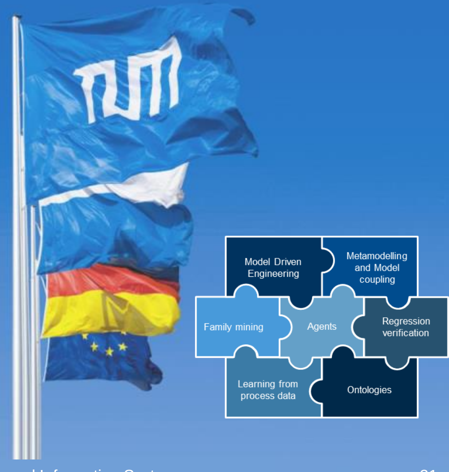
Definition of an agent



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

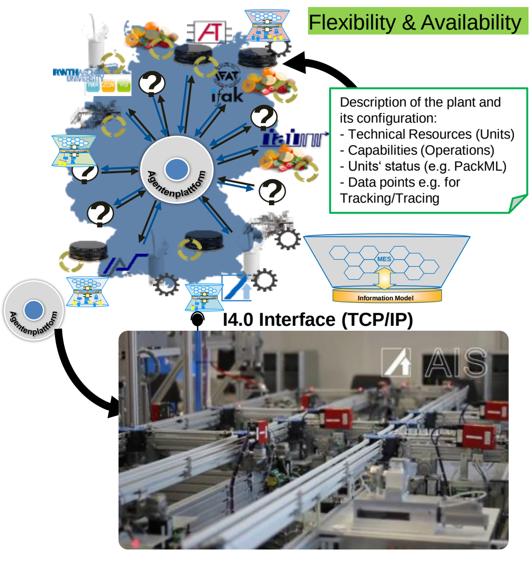
Model-driven Engineering and implementation of field level agents with IEC61131-3

1. Field level agents in automation
2. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
3. Use case sensor reconfiguration level 0: Model Driven Sensor reconfiguration with SysML – tank and pushers
   - SysML Introduction- Modeling instead of coding
   - OCL and Ontology introduction for knowledge description
4. Use cases in intralogistics
5. Metrics for adaptivity and flexibility
6. Conclusion and future work



Automation
and Information Systems
Technical University of Munich

MyJoghurt – Established Demonstrator for Industrie 4.0
Agent based cooperation of heterogeneous machines



Research concept

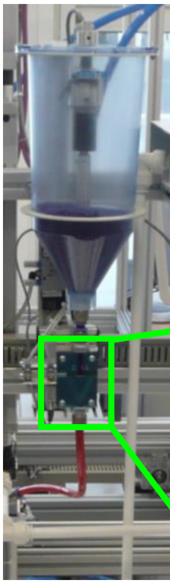
- **Objectives:** Improve flexibility and reliability of machines and plants
- **Challenges**: Reconfiguration during runtime to adapt to new products
- **Approach:** Model based approach with agents and product, process and resource description (ecl@ss, AutomationML, OPC-UA)
- **Results:** reconfiguration in case of a failure in between different plants run by different labs at german universities has been shown, concept for automatic adaption for new products based on ontologies is working

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

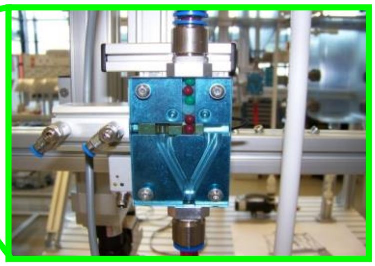
Now officially part of the PLATTFORM INDUSTRIE 4.0

Automation
and Information Systems
Technical University of Munich

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}



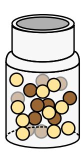
<AutomationML/>

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

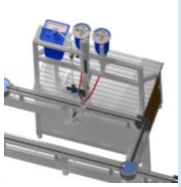
Input Diameter	
Name	Input Diameter
Description	
Value	8
Default Value	
Unit	mm
DataType	xs:string
Attribute1	
Output 1 Diameter	
Name	Output 1 Diameter
Description	
Value	8
Default Value	
Unit	mm
DataType	xs:string
Attribute1	
Attribute2	
Output 2 Diameter	
Name	Output 2 Diameter
Description	
Value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

Automation
and Information Systems
Technical University of Munich

Checking attributes of ressource model and product model with ontologies



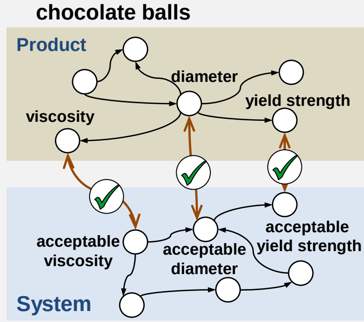
Product description
- **Name**: White chocolate balls
- **Viscosity**: 2.5 Pa*s
- **Yield strength**: 20 Pa
- **Diameter**: 0.5 cm
- **Aggregation state**: solid



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

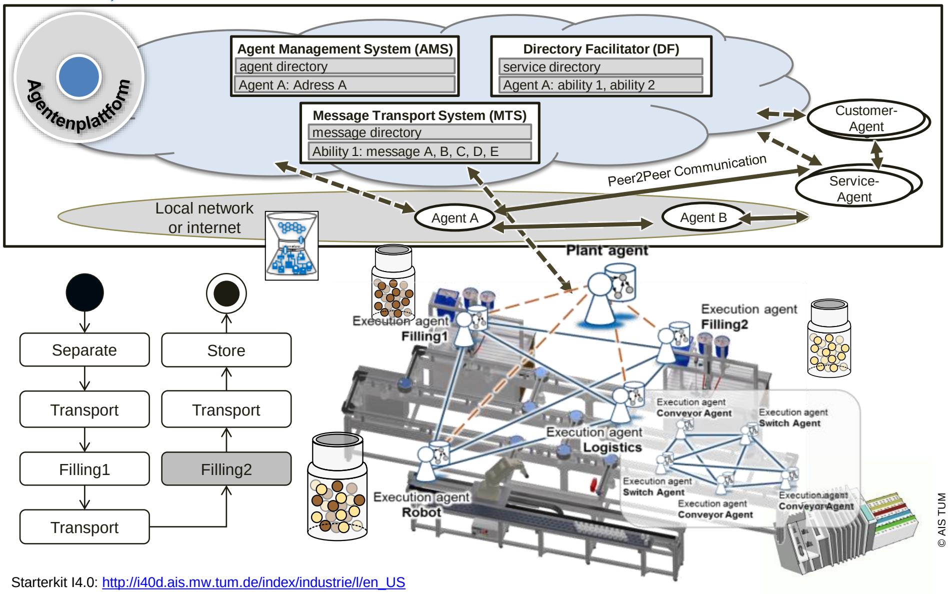
→ Basis to identify whether **filler** can manufacture yoghurts with **white chocolate balls**



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

Automation
and Information Systems
Technical University of Munich

Self-adapatation of an CPPS



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

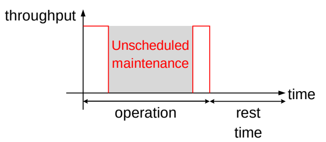
Source: B. Vogel-Heuser: Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik. In: Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer, 2014.

Automation
and Information Systems
Technical University of Munich

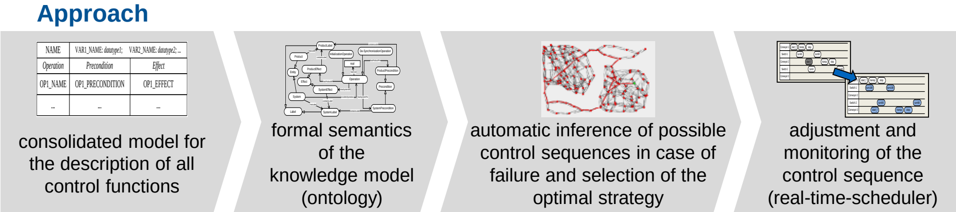
Research topic 4: knowledge-based reconfiguration

Motivation: knowledge-based reconfiguration

- Unforeseen failures and breakdowns
- Bad manageability of unforeseen failures
- High costs because of unscheduled downtime (loss of production)

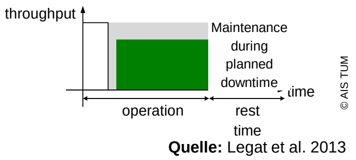


Approach



Previous Results

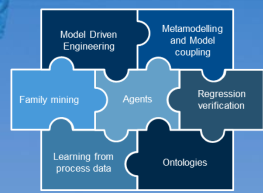
- Increased availability of production plants in case of component failure
- Reduced throughput loss in case of a reconfiguration (in comparison to model-based approaches)



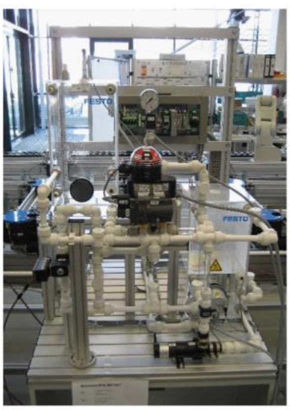
Quelle: Legat et al. 2013

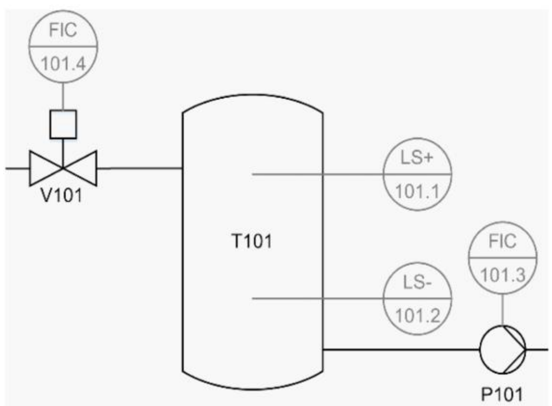
Model-driven Engineering and implementation of field level agents with IEC61131-3

1. Field level agents in automation
2. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
3. Use case sensor reconfiguration level 0: Model Driven Sensor reconfiguration with SysML – tank and pushers
   - SysML Introduction- Modeling instead of coding
   - OCL and Ontology introduction for knowledge description
4. Use cases in intralogistics
5. Metrics for adaptivity and flexibility
6. Conclusion and future work



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





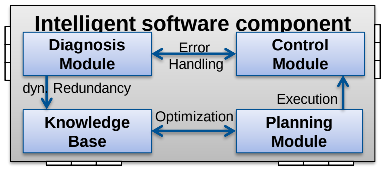
© AIS TUM

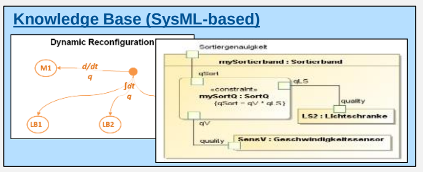
Source: P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche: „Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications“, 2016

Automation
and Information Systems
Technical University of Munich

Level 0 - fault tolerant sensor (Model based with agent)

Application in process control





PLC (IEC 61131-3)

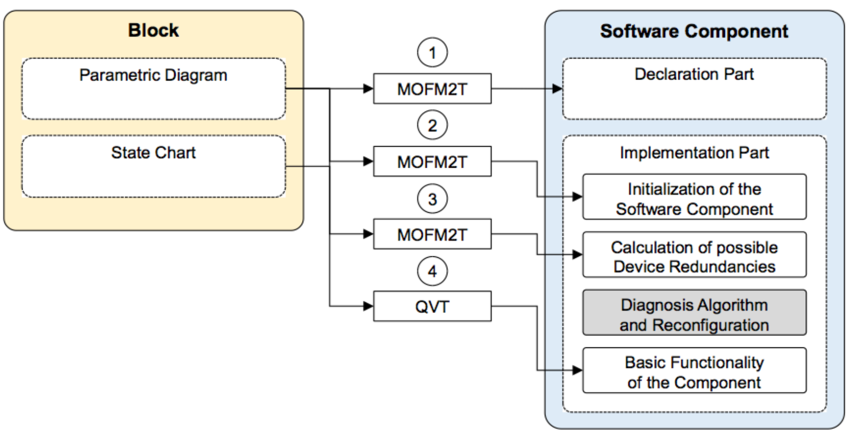
ADR(A1.out)	ADR(A2.out)	ADR(B1.out)	ADR(B2.out)
	ADR(VA2.out)		
		ADR(VB1.out)	
			ADR(VB2.out)

Templates referred to the MOFM2T-standard

Implementation of redundancy matrix

Automation
and Information Systems
Technical University of Munich

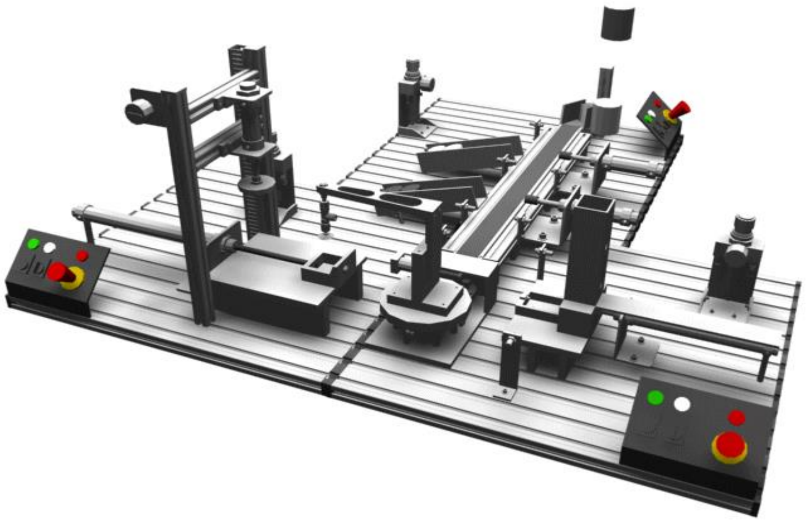
Agent-based self-aware, self-describing CPPS-module



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

Automation
and Information Systems
Technical University of Munich

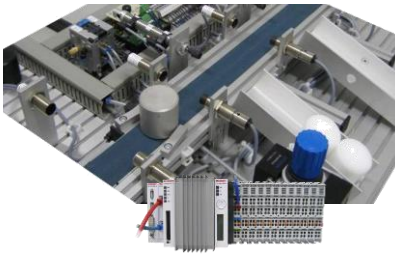
Introduction of the small lab scale production system pick-and-place-unit (PPU)

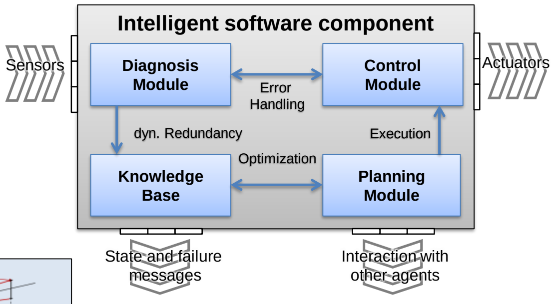


© AIS TUM

Automation
and Information Systems
Technical University of Munich

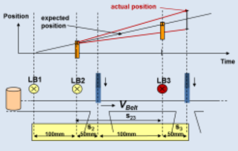
Intelligent Software Components on PLCs





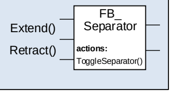
Diagnosis Module

- Evaluation of sensors values
- Execution of failure diagnosis



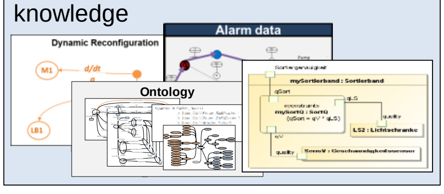
Control Module

Control of the plant module or other sub-agents



Knowledge Base

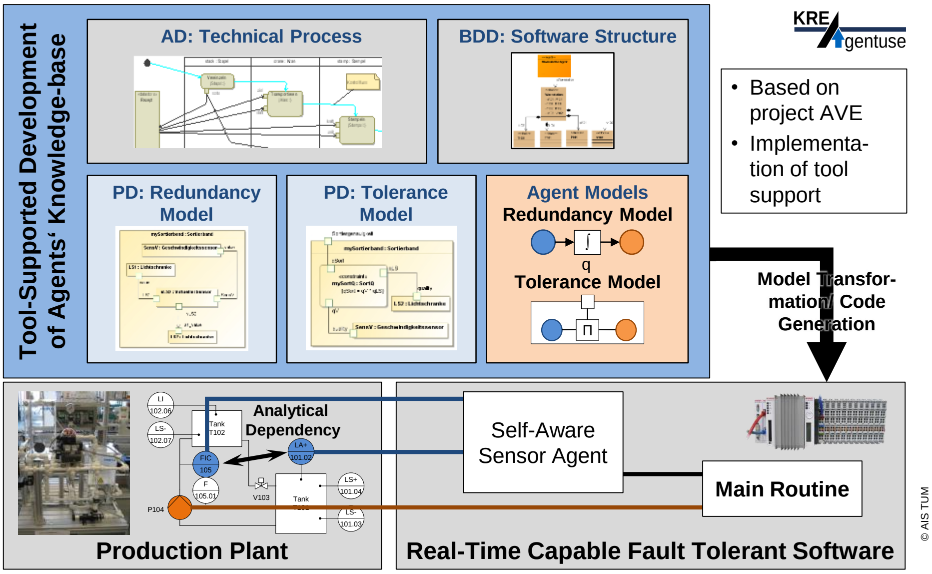
Models of the agents' local knowledge



© AIS TUM

Automation
and Information Systems
Technical University of Munich

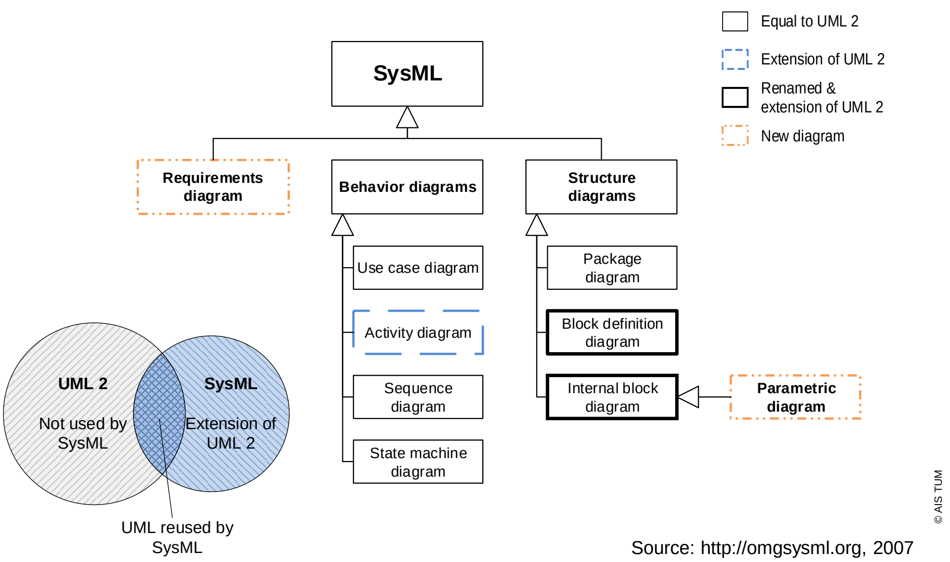
KREAagentuse: SysML-based automation software development



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

Automation
and Information Systems
Technical University of Munich

Construction of SysML diagrams



Source: http://omgsysml.org, 2007

Automation
and Information Systems
Technical University of Munich

Requirement diagram of the stamping process

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

Chart name in the tool

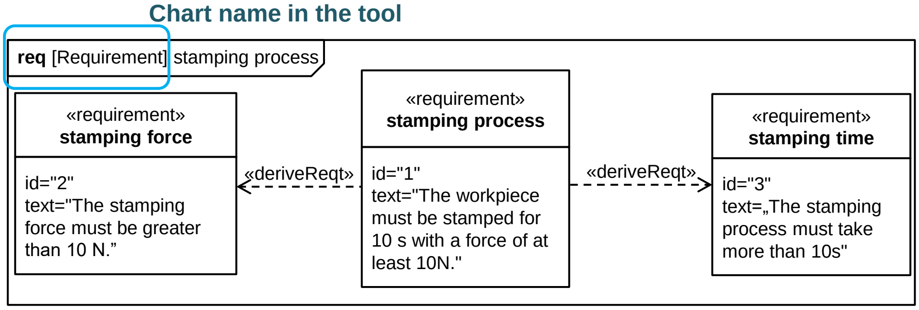
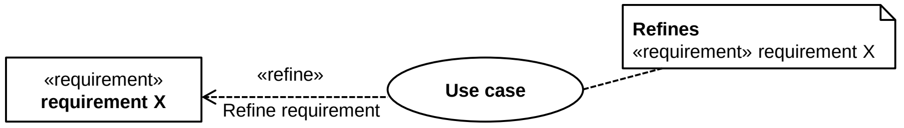


table [Requirement] Stempelprozess

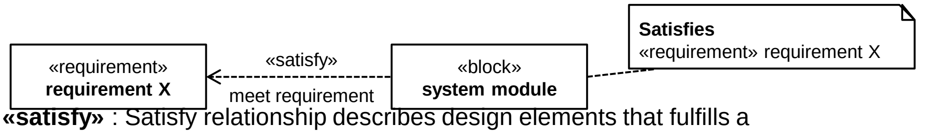
id	name	text
1	stamping process	"The workpiece must be stamped for 10 s with a force of at least 10N"
2	stamping force	"The stamping force must be greater than 10 N"
3	stamping time	"The stamping process must take more than 10s"

Automation
and Information Systems
Technical University of Munich

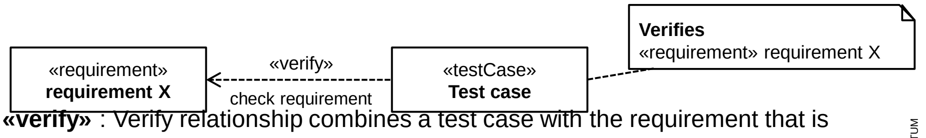
Requirement diagram – Relationships between requirements



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



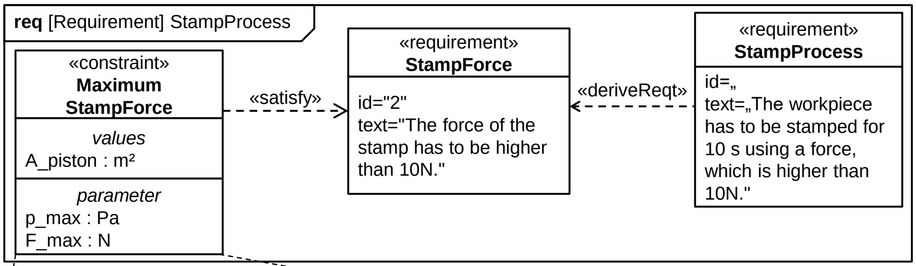
«satisfy» : Satisfy relationship describes design elements that fulfills a requirement.

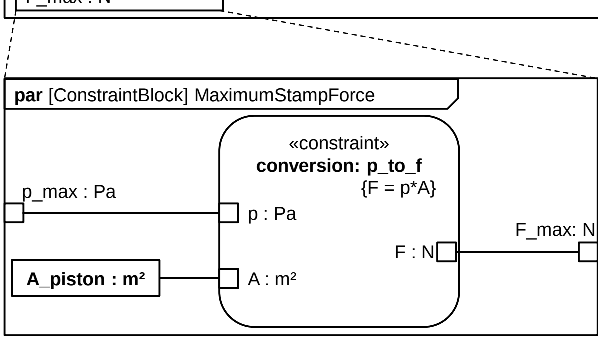


«verify» : Verify relationship combines a test case with the requirement that is checked by the test case.

Automation
and Information Systems
Technical University of Munich

Relationship between requirements diagram and parametric diagram

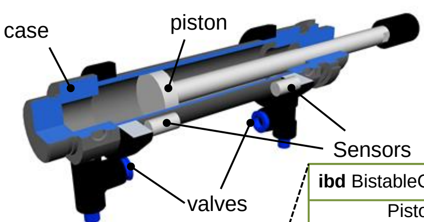




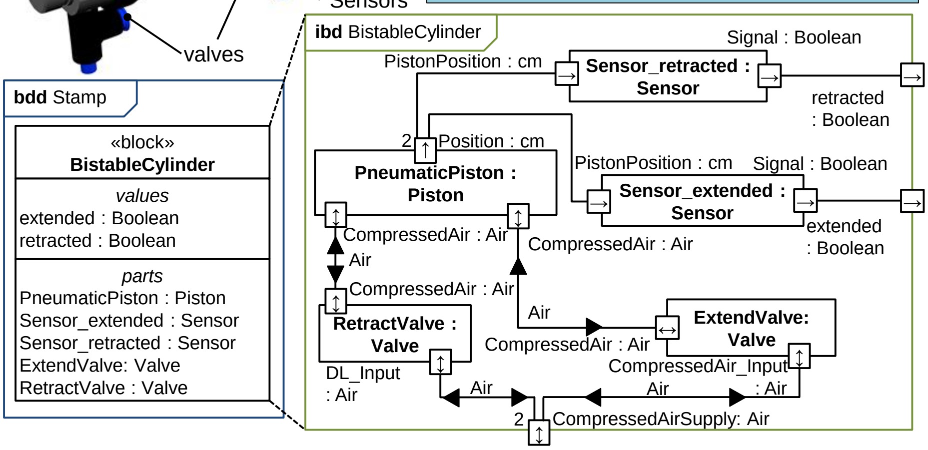
Defined requirements, which are specified in requirements diagrams, are met in "satisfy" relationship of "constraint" blocks in parametric diagrams.

Automation
and Information Systems
Technical University of Munich

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



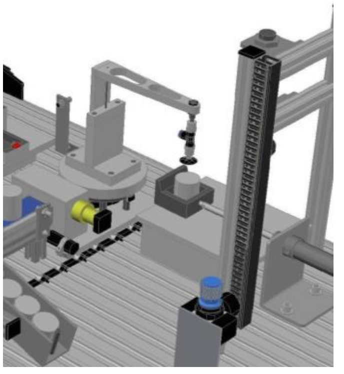
The relationships between parts of a block defined in the BDD are represented in the IBD.



Automation
and Information Systems
Technical University of Munich

Brainstorming

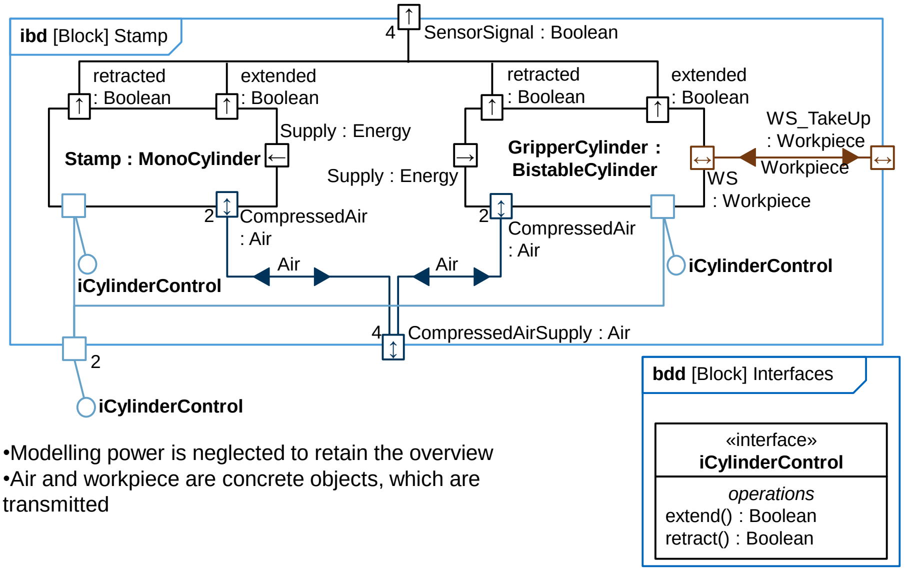
Which ports, which interfaces, which object flows have to be specified for the stamp of the PPU?



© AIS TUM

Automation
and Information Systems
Technical University of Munich

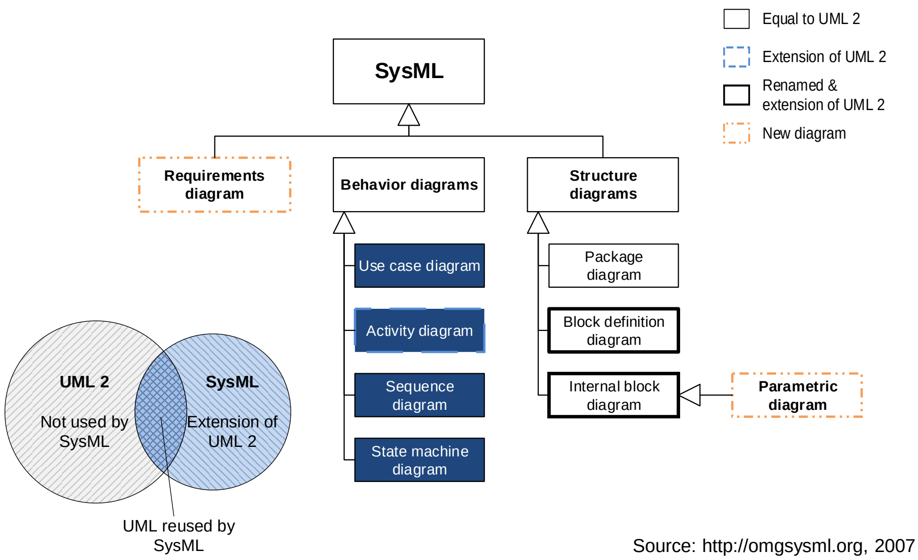
Structure model – Internal block diagram (IBD)
Example: Sorting plant: Block stamp



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

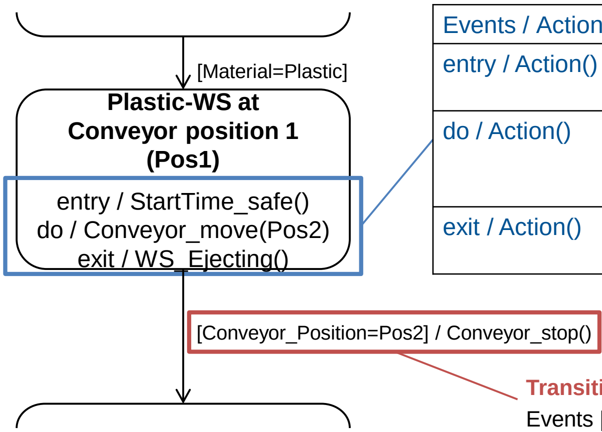
© AIS TUM

Construction of SysML diagrams



Source: http://omgsysml.org, 2007

Behavior model – state diagram
state description of the sorting plant



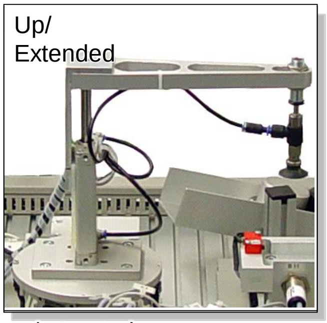
Events / Actions() in state diagram	
entry / Action()	is executed, while entering the state
do / Action()	is executed as long as the condition is active, or until it is completed
exit / Action()	is executed, while exiting the state

Transitions in state diagram:
Events [Condition]/Action

- In a state several methods of a class can be called (or other behavior diagrams)
- The state is exited if the subsequent transition is active: Then the Do-action is canceled and the Exit-action is executed once

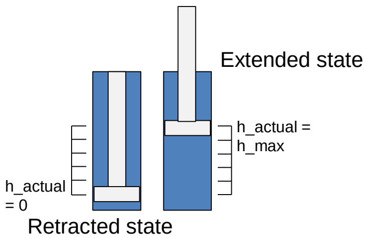
Automation
and Information Systems
Technical University of Munich

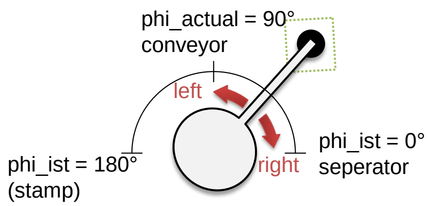
State-based behavior– Important boundary conditions to model the crane



Down/retracted

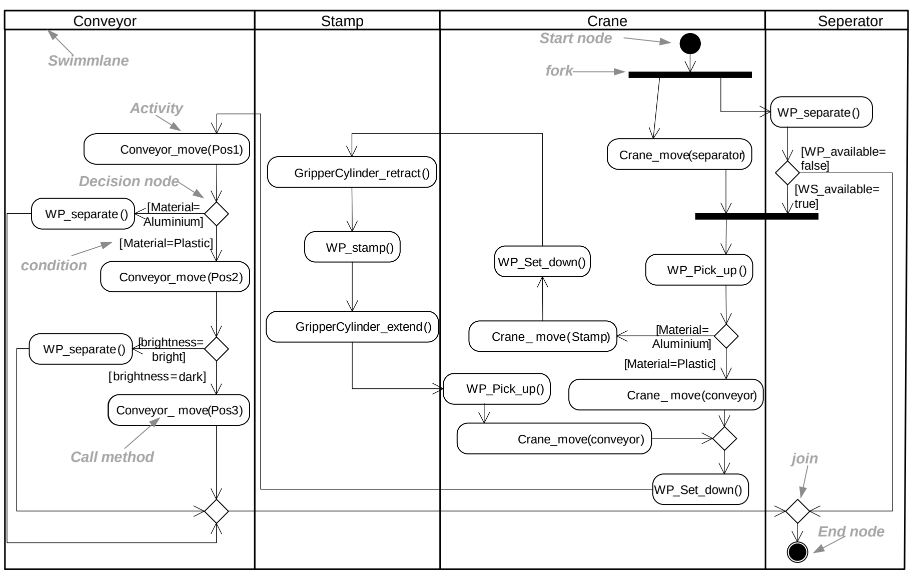
- Crane is able to move to retracted and extended positions
- A pneumatic cylinder is used to extend the crane





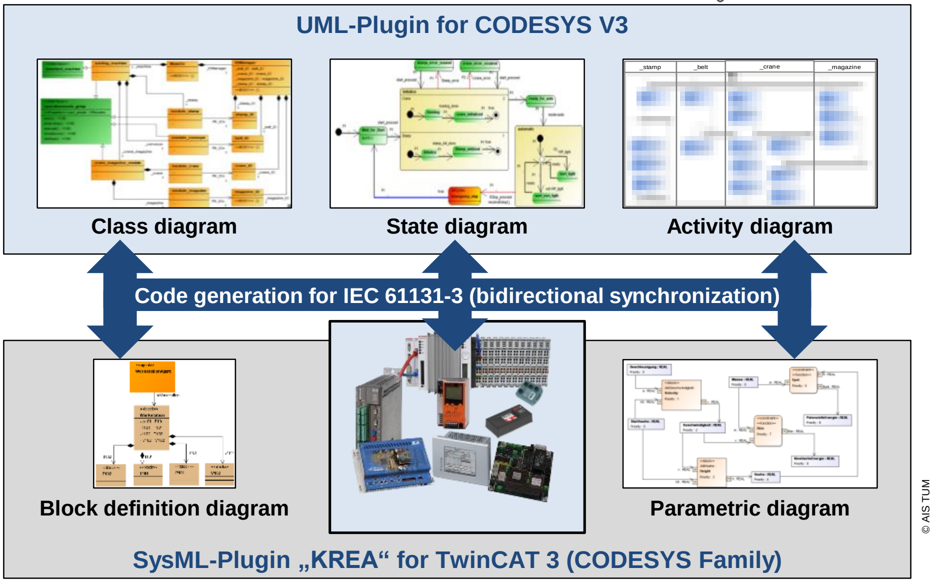
Automation
and Information Systems
Technical University of Munich

Behavior model: Activity diagram Application example: Sorting process



Automation
and Information Systems
Technical University of Munich

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



Automation
and Information Systems
Technical University of Munich

How can knowledge of software agents be modelled? Which kinds of knowledge have to be modelled?

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“

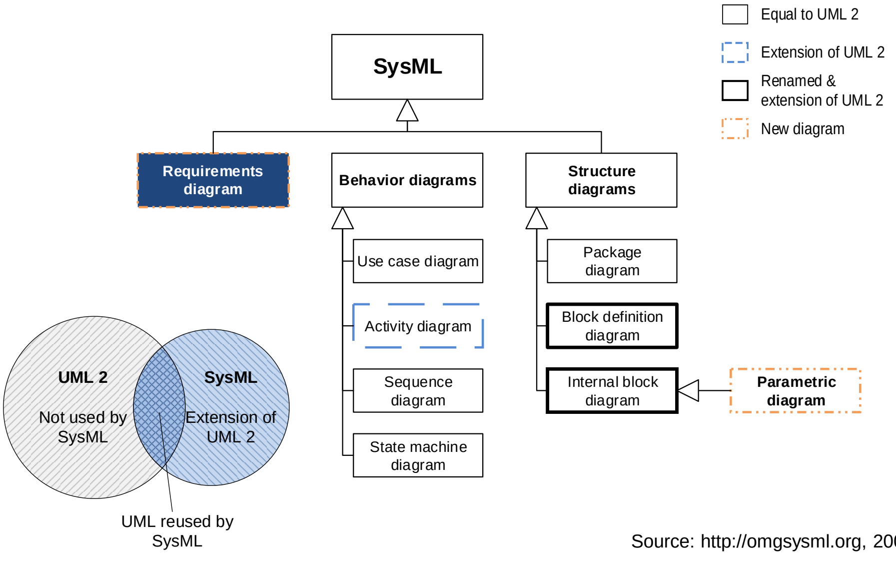
What is missing?

Automatic processing

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

Automation
and Information Systems
Technical University of Munich

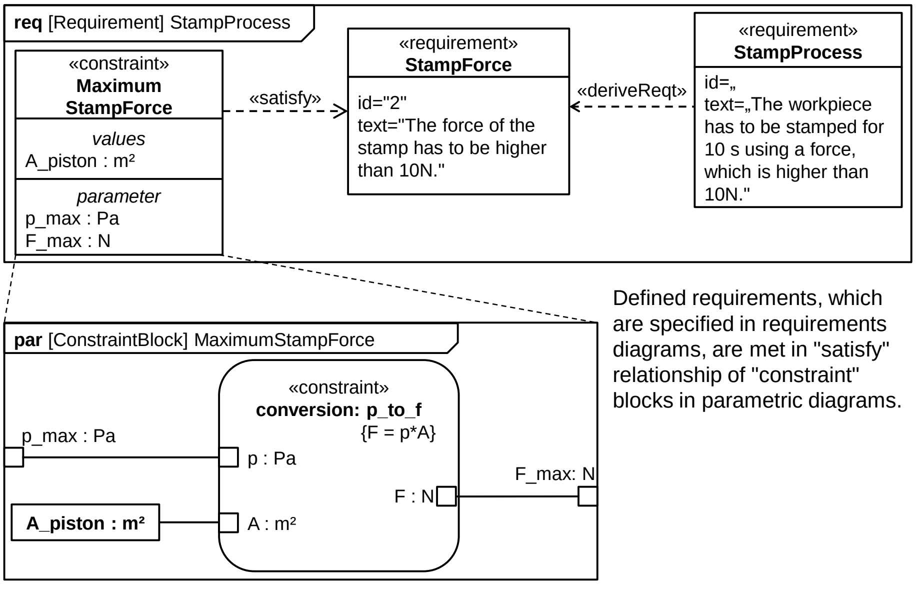
Construction of SysML diagrams



Source: http://omgsysml.org, 2007

Automation
and Information Systems
Technical University of Munich

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.

Automation
and Information Systems
Technical University of Munich

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

KREAgentuse

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
```


Automation
and Information Systems
Technical University of Munich

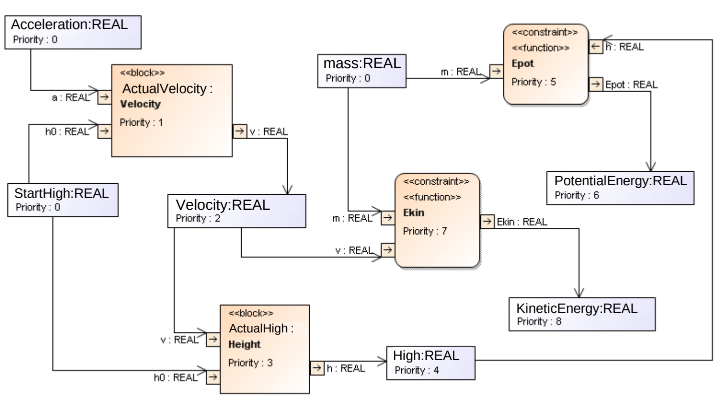
Exemplary mapping between SysML and IEC 61131-3: Parametric diagram

KREAgentuse

Parametric diagram

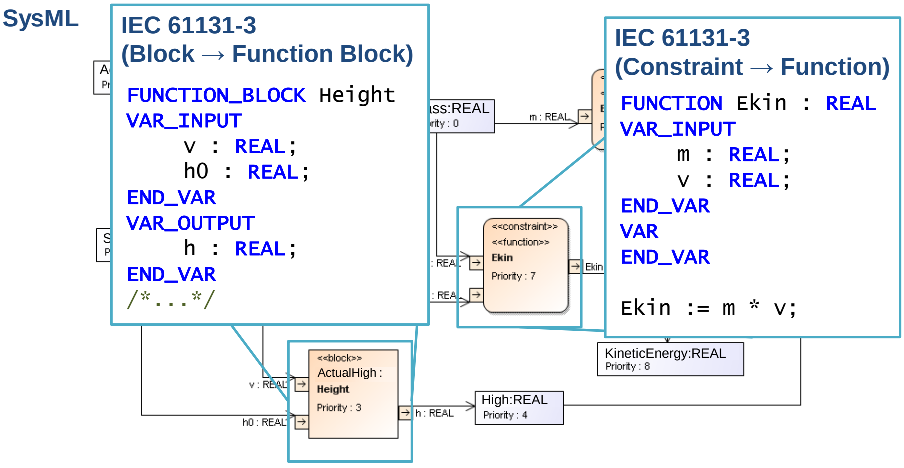
- Calling of IEC 61131-3 functions and function blocks
- Visualization and structuring of parametric relationships

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

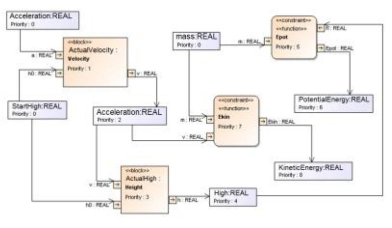


Automation
and Information Systems
Technical University of Munich

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



Problem

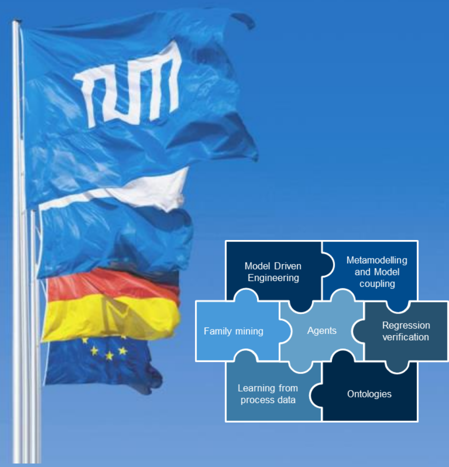
What information is missing in pure SysML parametric diagram to automatically generate code?

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);
```

Model-driven Engineering and implementation of field level agents with IEC61131-3

1. Field level agents in automation
2. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
3. Use case sensor reconfiguration level 0: Model Driven Sensor reconfiguration with SysML – tank and pushers
   - SysML Introduction- Modeling instead of coding
   - OCL and Ontology introduction for knowledge description
4. Use cases in intralogistics
5. Metrics for adaptivity and flexibility
6. Conclusion and future work



Introduction of Object Constraint Language (OCL) to model knowledge of sequences allowed

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

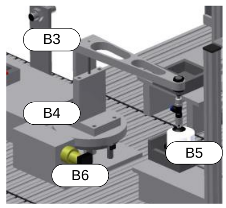
- Invariants in class diagrams
- Pre- and postcondotion for
- Conditions in sequence and state diagrams
- Condition of the UML metamodel

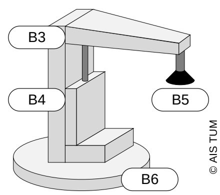
Interactive exercise – material collection by the crane

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 (Crane::GrabMaterial()) 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 (Crane::Turn(X)) of the crane are:
- **Precondition:** Crane is not allowed to leave the angle range
- **Postcondition:** Crane is at a new position

Sensor	description	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





Automation
and Information Systems
Technical University of Munich

Interactive exercise – material collection by the crane

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 Crane
inv: B6 > 0 and B6 <= 360
```

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

```
context Kran::GrabMaterial()
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 (Crane::Turn(X)) of the crane are:
  - **Precondition:** Crane isn't allowed to leave the angle range
  - **Postcondition:** Crane is at a new position

```
context Crane::Turn(X)
pre:   B6 + x > 0 and B6 + x <= 360
post:  B6 = B6@pre + x
```

Sensor	description	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

What for is the OCL or the temporal logic needed?

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?

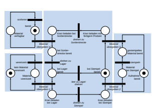
- A system model describes structur and behaviour of a system (e.g. UML/OCL)
- A formal description of the requirements

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

Summary of the learned content

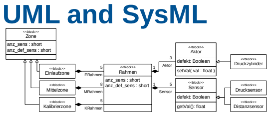
Petri-nets



- Modeling of control processes
- Formal verification of process attributes

Knowledge: transition activation (pre-, postedges)

UML and SysML



- Modeling of (software) systems
- Graphical documentation

OCL

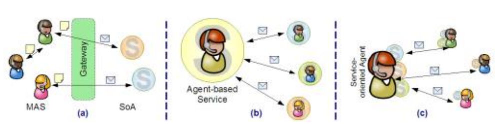
```
context Einlaufzone
inv:  ERahmen->forAll( r : Rahmen |
      r.Sensor->select(s |
          s.oclIsTypeOf(Drucksensor))->size()=3 and
      r.Sensor->select(s |
          s.oclIsTypeOf(Distanzsensor))->size()=2)
```

- Extension mechanism
- Formulation of boundary conditions, invariants, pre- and postconditions

Knowledge: correctness of the models, pre- and postconditions of the operations

Intelligent Systems

SOA
Agents
CPPS



- Implementation of intelligent systems in the field of automation technology
- Agents, service-oriented architectures, Cyber-Physical Production Systems

Automation
and Information Systems
Technical University of Munich

Declarative models and their processing: Rule-based models vs. Ontology-based models

Rule-based models	Ontology-based models
• Modeling paradigm: IF/THEN-rules and -facts → description of concrete situations	• Modeling paradigm: Description of quantities and their relations → Description of a class of situations
• Formalization of „simpler“ knowledge	• Formalization of „more complex“ knowledge
• Inference based on facts → Derivation of new facts based on known facts	• Inference is based on quantity descriptions → derivation of new groups
• Control of complexity by choosing the inference mechanism	• Control of complexity because of expressiveness of the ontology
• Difficult maintenance of big rule bases → size limitation of a rule base	• Simpler maintenance of big ontologies → bigger knowledge bases possible
• Consistency checks only by using additional instruments	• Automatic checks of consistency done by the reasoner

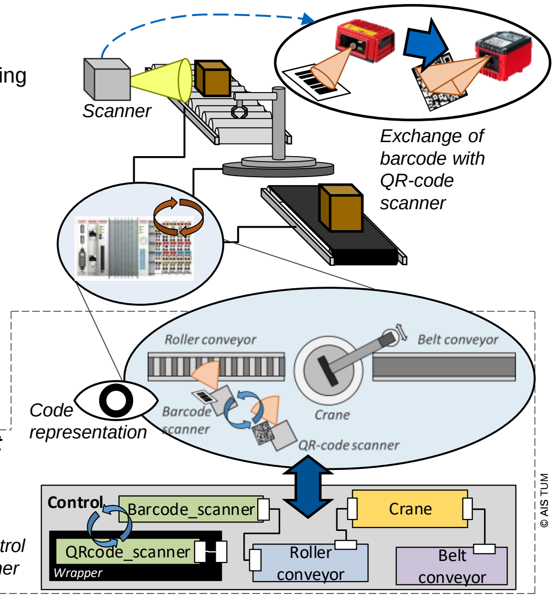
Model-driven Engineering and implementation of field level agents with IEC61131-3

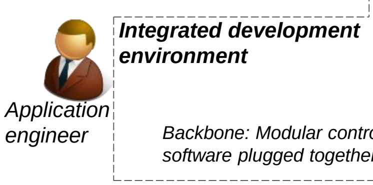
1. Field level agents in automation
2. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
3. Use case sensor reconfiguration level 0: Model Driven Sensor reconfiguration with SysML – tank and pushers
   - SysML Introduction- Modeling instead of coding
   - OCL and Ontology introduction for knowledge description
4. Use cases in intralogistics
5. Metrics for adaptivity and flexibility
6. Conclusion and future work

Version-Update in intralogistics (ACOMA)

Approach

- Modularized model-driven engineering approach
- backward compatibility approach of software modules





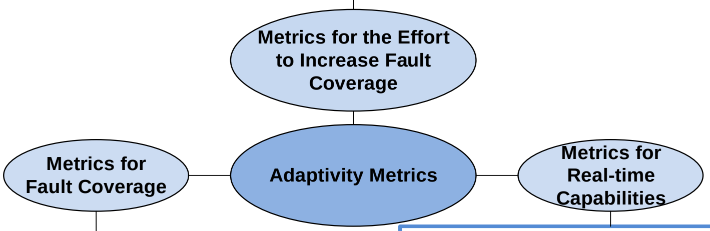
Model-driven Engineering and implementation of field level agents with IEC61131-3

1. Field level agents in automation
2. Agents@PLC for CPPS and Industrie 4.0 – my joghurt demonstrator
3. Use case sensor reconfiguration level 0: Model Driven Sensor reconfiguration with SysML – tank and pushers
   - SysML Introduction- Modeling instead of coding
   - OCL and Ontology introduction for knowledge description
4. Use cases in intralogistics
5. Metrics for adaptivity and flexibility
6. Conclusion and future work

Automation
and Information Systems
Technical University of Munich

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

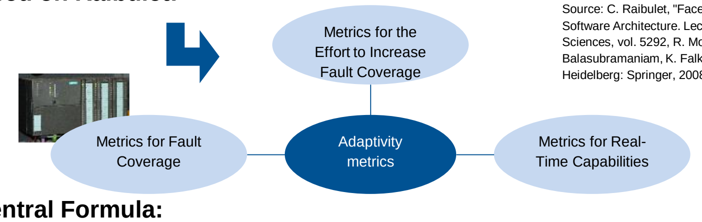
Automation
and Information Systems
Technical University of Munich

Adaptivity Metric

Metrics for adaptivity according to Raibulet
Defined Architectural, Structural and Performance metrics

Set of metrics for aPS focusing on code changes due to reconfiguration according to Vogel-Heuser et al. based on Raibulet:

Source: C. Raibulet, "Facets of Adaptivity" in Software Architecture. Lecture Notes in Computer Sciences, vol. 5292, R. Morrison, D. Balasubramaniam, K. Falkner, Eds., Berlin, Heidelberg: Springer, 2008, pp. 342–345.



Central Formula:

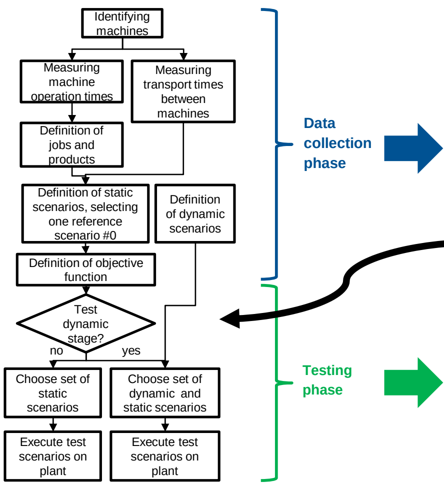
$$AI_{model} = \frac{(V(G)_{new} - V(G)_{old})}{0.5 \cdot (V(G)_{new} + V(G)_{old})}$$

Minimal Modelling Effort Adaptivity Index AI_{model}
Relative modeling effort for adaptability comparing McCabes cyclomatic complexity V(G) of the adapted program ($V(G_{new})$) with the classical one ($V(G_{old})$)

Problem: Metric focuses on code for evaluation. Lines of Code might not be an adequate measuring unit to compare plants in different fields of industry
→ need for new, more general metric to include the overall performance of CPPS

Automation
and Information Systems
Technical University of Munich

The benchmarking process proposed by Trentesaux at one glance



Output:
- List of machines, operations, jobs, products and process times (operation and transport)
- List of static and dynamic production scenarios
- Reference scenario
- Objective function for reference scenario

Output:
- Test results
- QLS (and QTS) rating

Thank you for your attention!

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

Ordinaria
Automation and Information Systems (AIS)
Mechanical engineering,
Technische Universität München
www.ais.mw.tum.de; vogel-heuser@tum.de