A Model Driven Approach for Requirements Engineering of Industrial Automation Systems

Hongchao Ji (Bosch Rexroth AG, Germany)
Oliver Lenord (Bosch Rexroth AG, Germany)
Dieter Schramm (University Duisburg Essen, Germany)

05.09.2011
Zürich

4th International Workshop on Equation-Based Object-Oriented Modeling Language and Tools

Rexroth
Bosch Group

Content

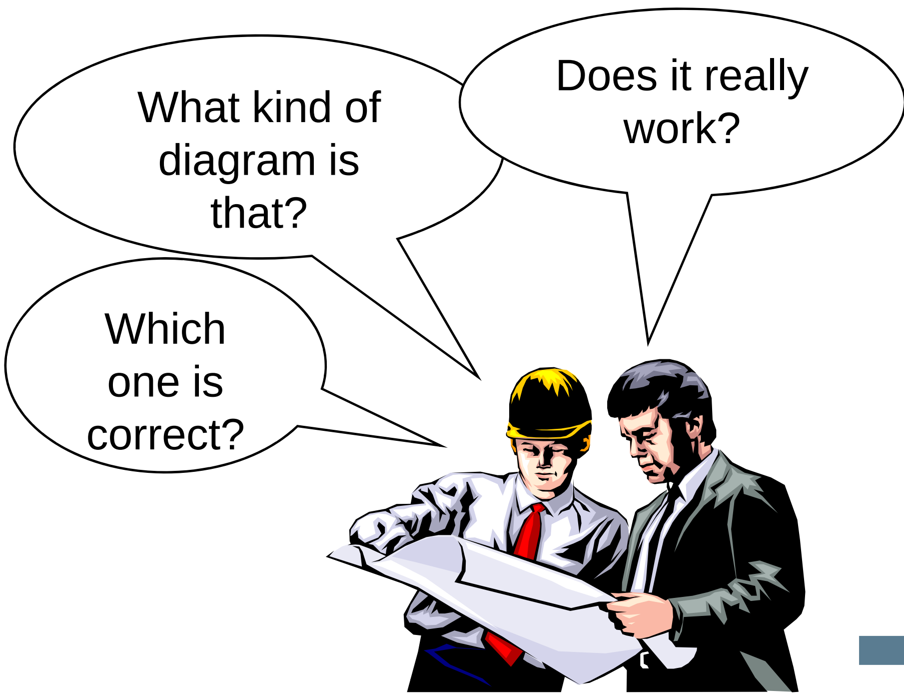
1

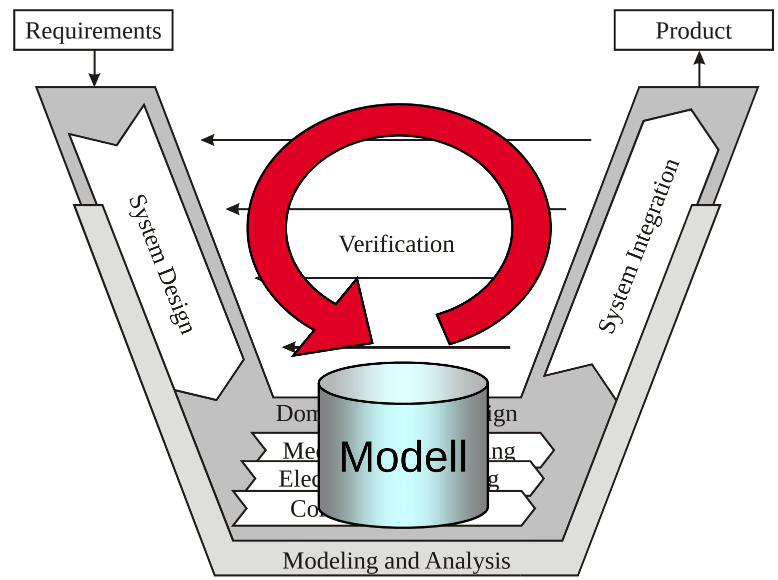
- **Introduction**
- Model Driven Requirements Engineering
- Case Study: Hydrostatic Press System
- Conclusion and Outlook

Motivation

What are the problems?

- Textual specifications ➔ Inconsistencies between documents
- No universal modeling language ➔ Misunderstanding among different disciplines
- Manuel verification process ➔ Hard to detect the design error at the early stage





V-Model according to VDI 2206

➔ **Model Driven Requirements Engineering**

Introduction

- Model Driven Requirements Engineering (MDRE)
  - *Development method providing the means for using **models** to direct the course of understanding, design, construction and deployment.* **[OMG: MDA Guide]**

- The deficiencies of **SysML** as the modeling language in MDRE
  - Requirement constructs are not sufficiently defined
  - It is not capable to describe dynamic models

Goals of this work

- Analyze the applicability of MDRE in the field of industrial automation
- Extend the SysML requirement constructs
- Integration with Modelica by using ModelicaML

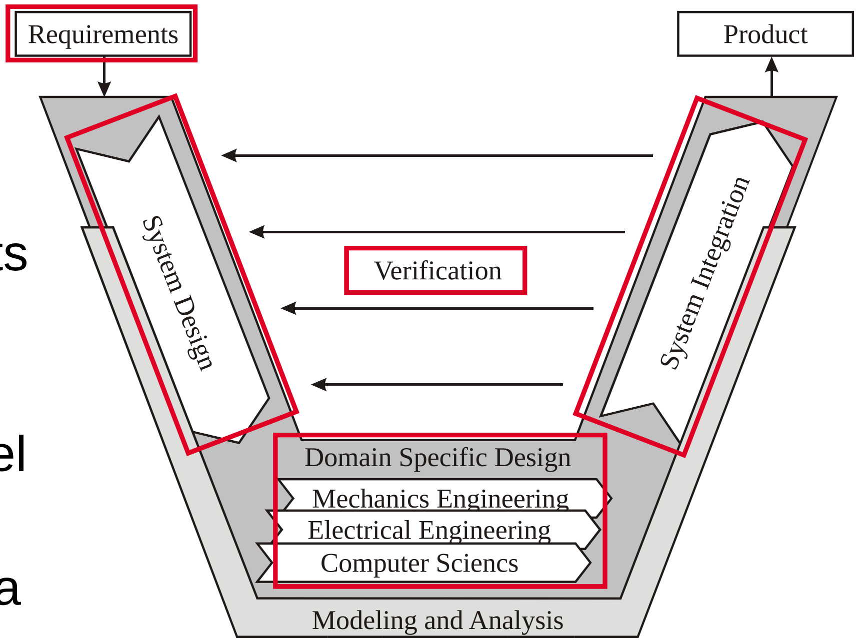
Content

2

- Introduction
- **Model Driven Requirements Engineering**
- Case Study: Hydrostatic Press System
- Conclusion and Outlook

Application of MDRE in Industrial Automation

- Modeling languages and tools used in each design phase
  - **Requirements:** SysML requirement constructs extended by the MDRE4BR profile
  - **System design:** SysML diagrams used to model the descriptive system design
  - **Domain-specific design:** domain-specific drawing tools reflecting the SysML concepts is desired
  - **System integration:** Modelica models created or generated from the design model by using ModelicaML
  - **Verification:** run the simulation of Modelica model to verify the system design against system requirements



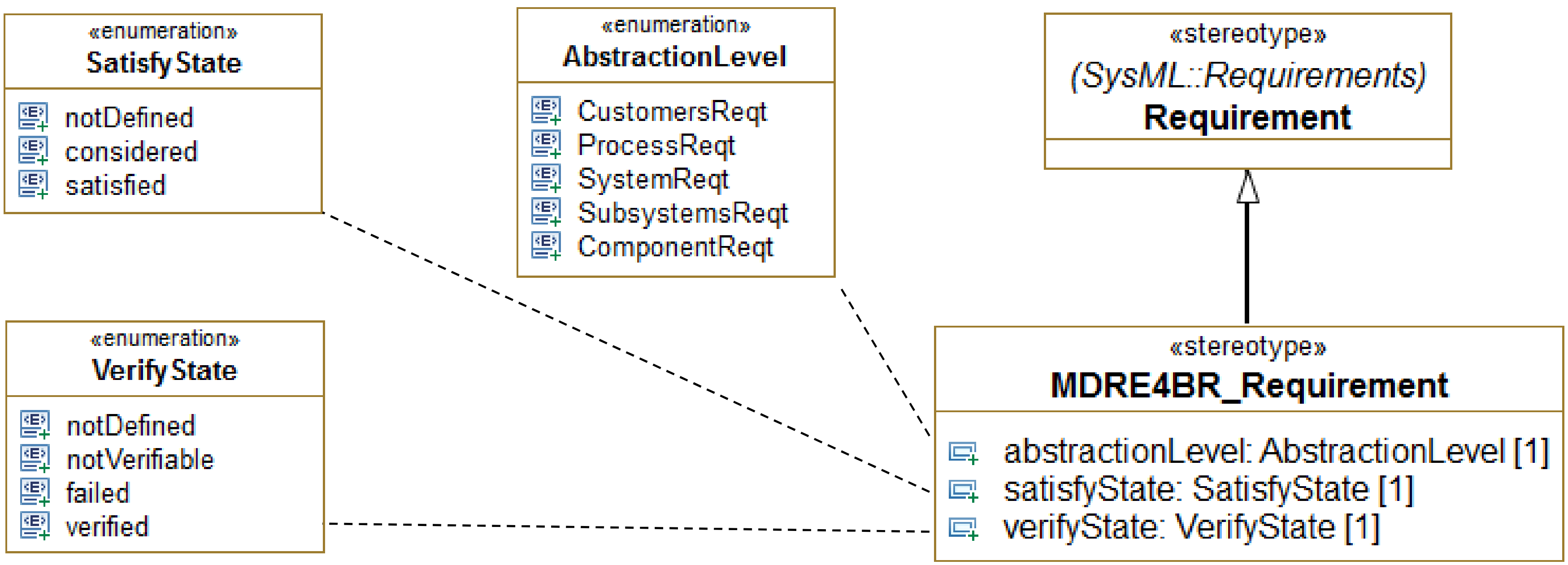
MDRE4BR Profile

- The shortcomings of SysML requirements constructs
  - No classification of requirements
  - Limited traceability concept
  - No executable test cases

- Objective of the MDRE4BR profile
  - UML profile to extend the SysML requirements constructs
  - Enhanced definitions of requirements
  - Extend the traceability links
  - Verification process compliant with the development of industrial automation systems
  - Integration with Modelica

Definition of Requirements

- **Abstraction Level** helps to cluster the requirements in a break down structure
- **Satisfy State** check the coverage of the requirement
- **Verify State** identifies the verification results

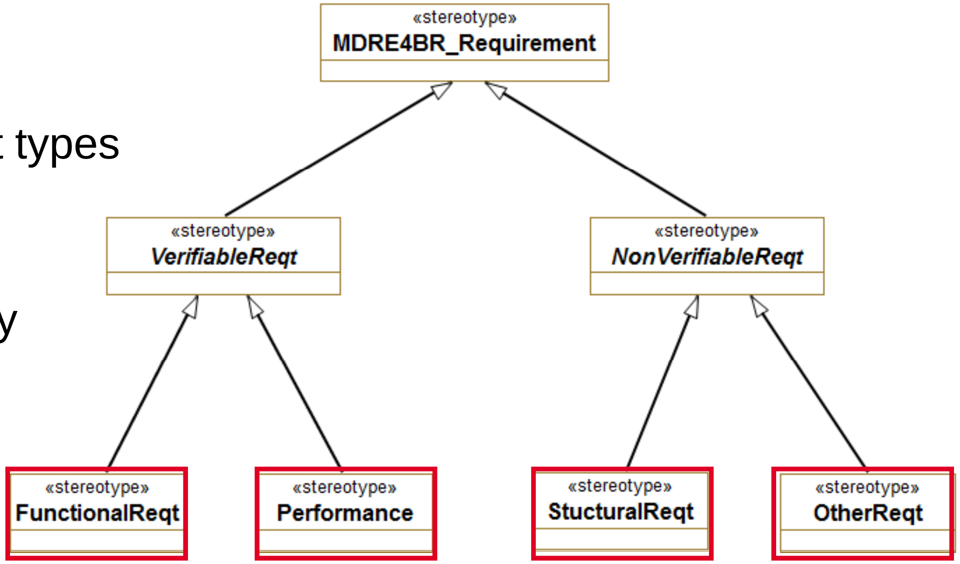


Rexroth
Bosch Group

Classification of Requirements

- Classification
  - A *functional requirement* is a requirement that should produce an expected reaction to a given stimuli.
  - A *performance* is a requirement to check whether a system variable such as timing, speed, volume or throughput is in a desired range.
  - A *structural requirement* is a requirement which describes the structural demand of the stakeholder.
  - A *other requirement* is all the other types of not-analytically-verifiable requirements

- Reasons for classification
  - Different attributes for different types
  - Processed differently in the verification stage
  - Basis for the further traceability concept



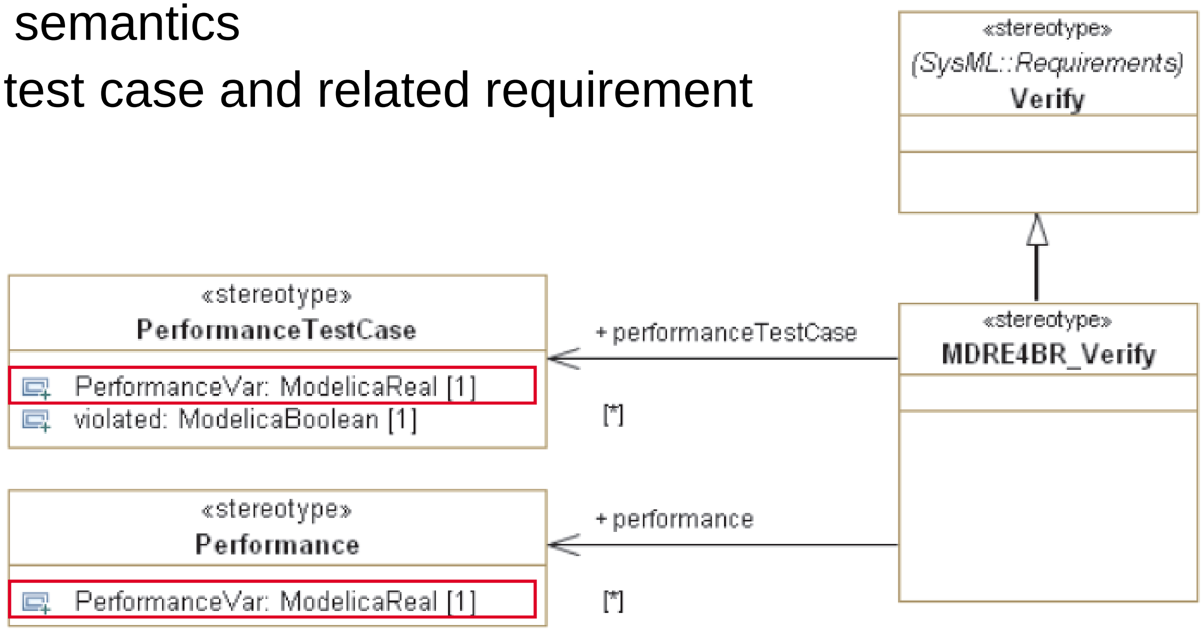
[Glinz, 2007]

Traceability of Requirements

- Traceability in SysML
  - Requirement to requirement: (copy, containment, deriveReqt)
  - Requirement to design element: (**satisfy**, trace, refine)
  - Requirement to test case: (**verify**)

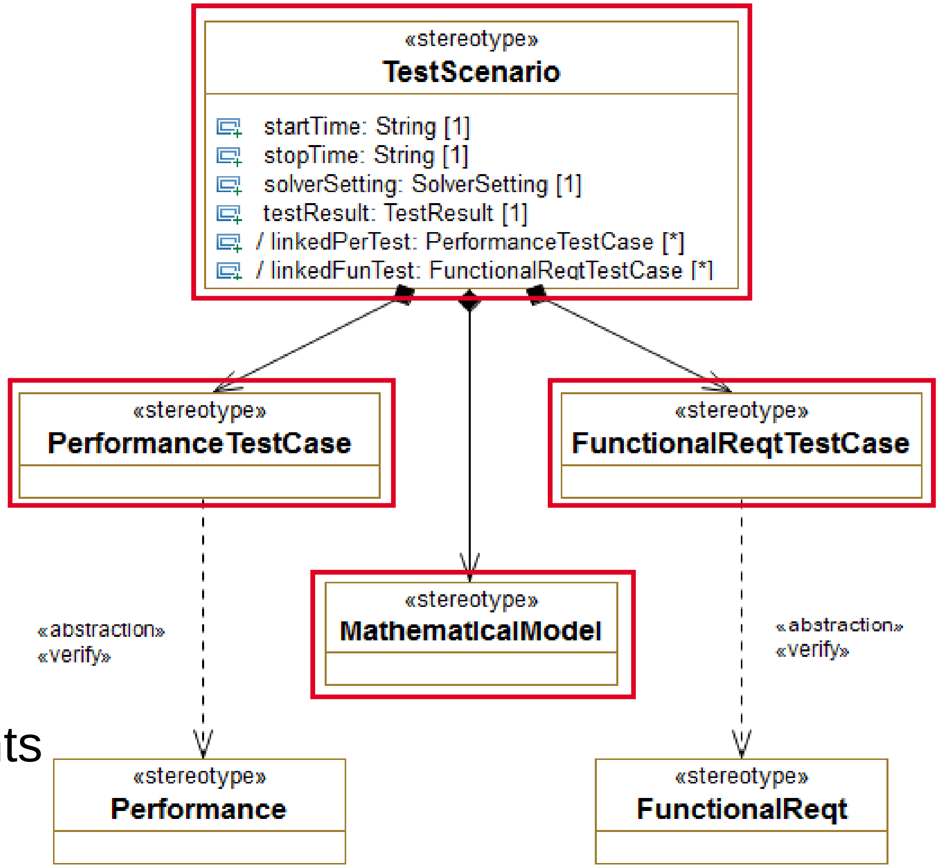
- Concept to extend the **satisfy** and **verify** relationships
  - More precisely defined semantics
  - Binding of variables of test case and related requirement with verify relation

- Open issue
  - Trace back the verification results to requirement automatically



Verification of Requirements

- New stereotypes
  - Performance Test Case
  - Functional Requirement Test Case
  - Test Scenario
  - Mathematical Model
- Verification process
  - Define a test scenario
  - Select a mathematical model
  - Select the associated test cases
  - Run the simulation
  - Update the *Verify State* in requirements
- Open issue
  - Define a traceability link between test cases and mathematical model to automate the variables binding



Content

3

- Introduction
- Model Driven Requirements Engineering
- **Case Study: Hydrostatic Press System**
- Conclusion and Outlook

Rexroth
Bosch Group

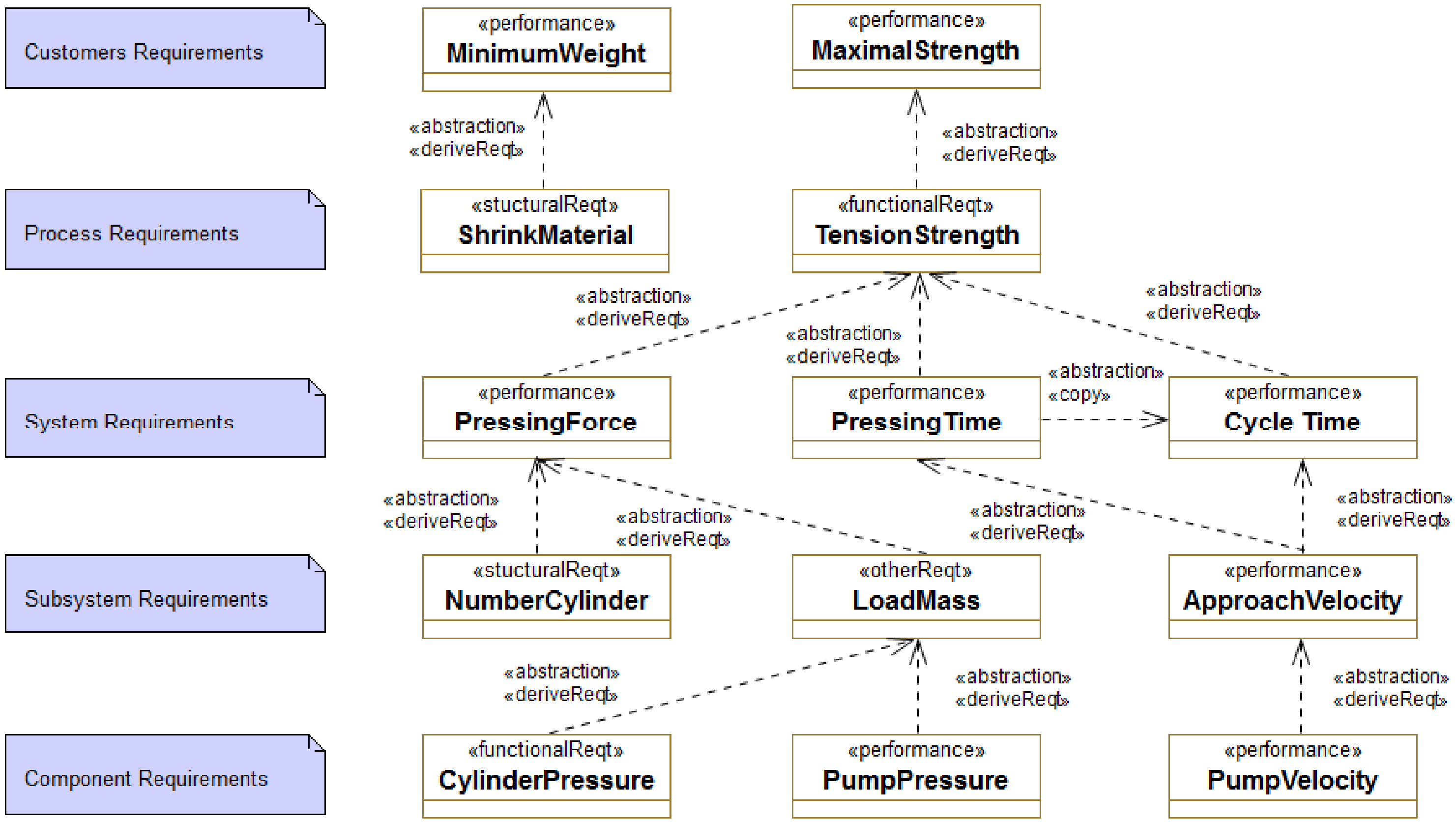
Case Study: Hydrostatic Press System

- The tasks
  - Analysis of customer requirements
  - Modeling of customer requirements
  - Modeling of system design
    - System structures
    - System behaviors
    - Domain-specific design
  - Modeling of the analytical models
  - Verification of systems design against requirements

Rexroth
Bosch Group

Step 1: Capture and Refine Requirements

- Capture the customer requirements as stereotyped requirements according to the classification
- Refine the higher level requirements to lower level requirements



Rexroth
Bosch Group

Step 1: Capture and Refine Requirements

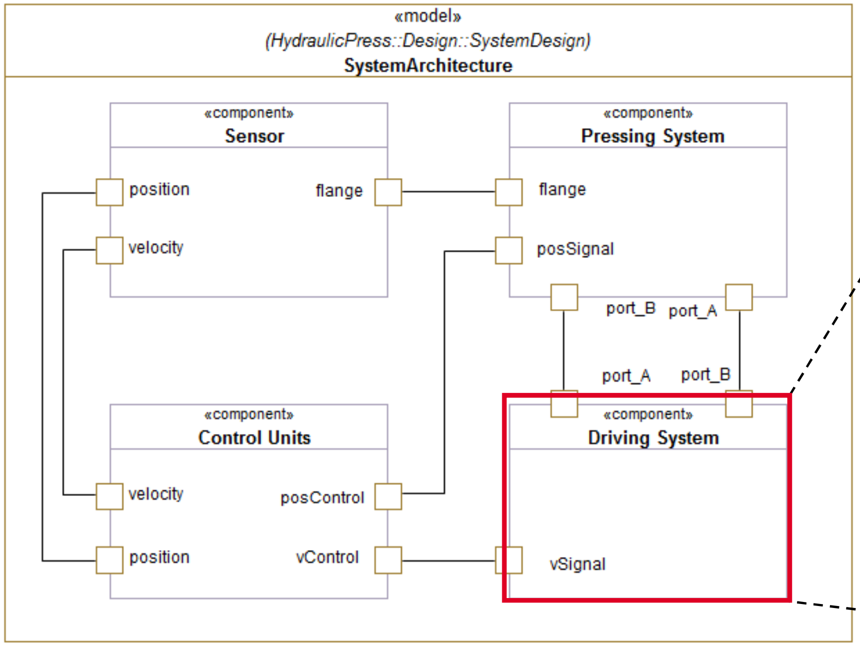
- Capture the customer requirements as stereotyped requirements according to the classification
- Refine the higher level requirements to lower level requirements

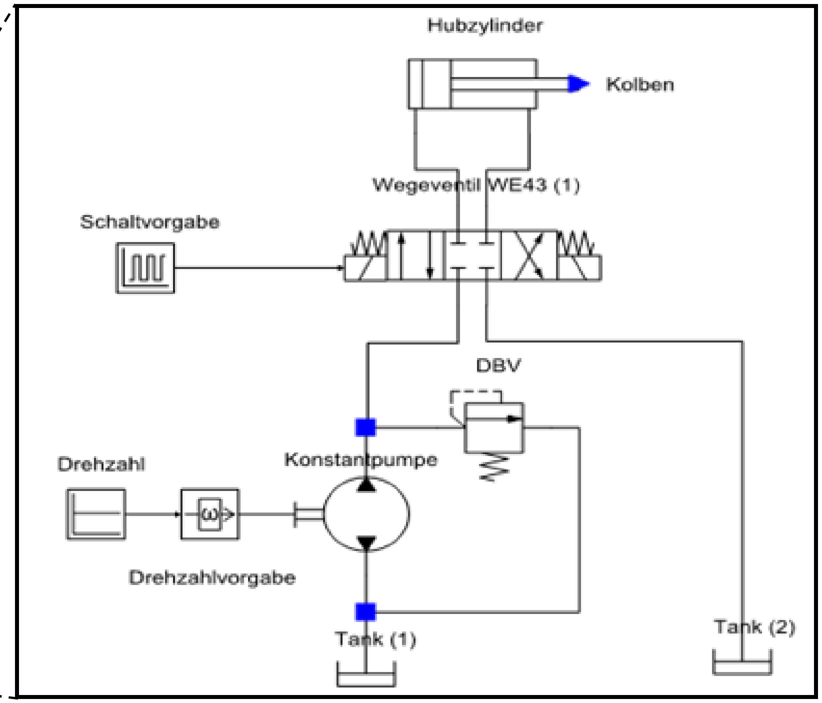
Remarks:

- The requirements can be structured systematically using different abstraction levels and classifications

Step 2: Create System Design Models

- System design: create the structural and behavioral model with SysML diagrams
- Domain-specific design: create the detailed design model in a domain-specific drawing tool





Rexroth
Bosch Group

Step 2: Create System Design Models

- System design: model the structural and behavioral model with SysML diagrams
- Domain-specific design: model the detailed design model in a domain-specific drawing tool

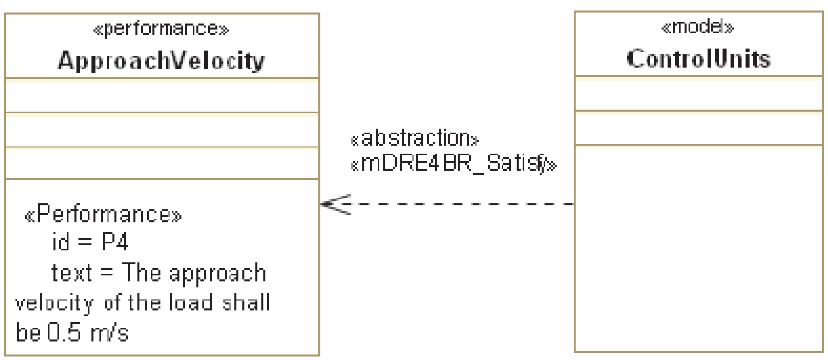
Remarks

- The system design process is well supported by SysML
- SysML is not suited for the domain-specific design
- Tool support for the domain-specific design with the extension that it can also reflect SysML constructs (stored as SysML model) is desired

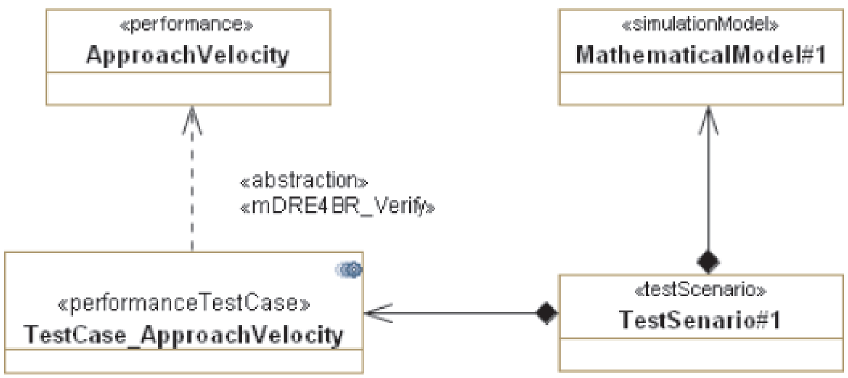
Rexroth
Bosch Group

Step 3: Establish Traceability Links

- Satisfy
  - Select the design object supposed to satisfy the requirement
  - Check and update the *Satisfy State*
  - Check the coverage of requirements



- Verify
  - Define test case for each analytically-verifiable requirement
  - Create the verify relation
  - Binding the variables

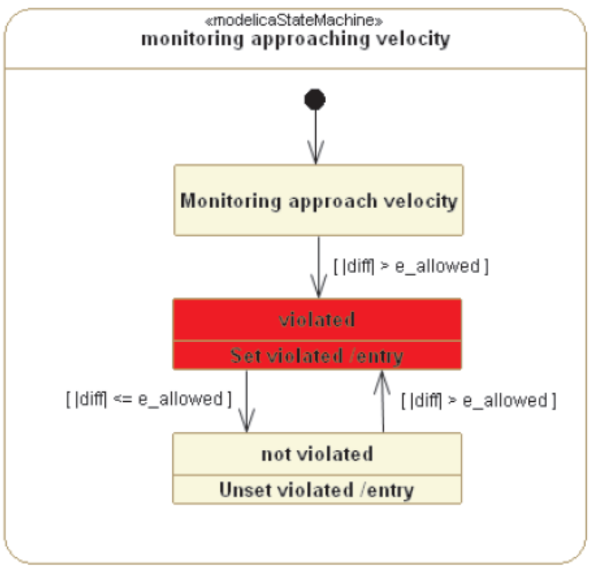


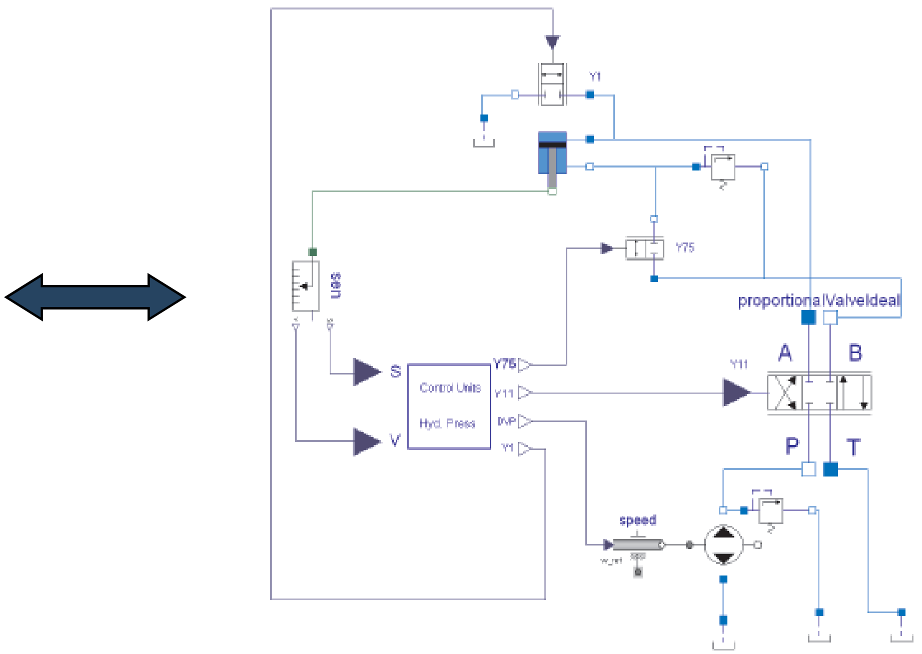
Remarks

- Improved impact analysis by using the extended traceability links

Step 4: Create Analytical Models

- Modeling test case including violation monitor (vVDR, [Schamai, 2010])
- Creating related mathematical models
- Link the mathematical model and test cases in the test scenario





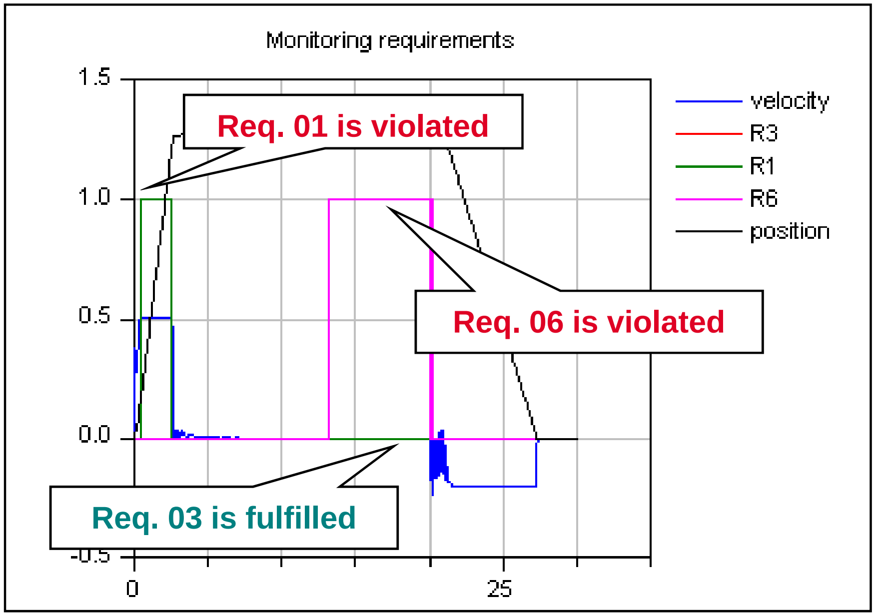
Remarks

- The vVDR method is applicable to our MDRE process

Rexroth
Bosch Group

Step 5: Update Results of Verification

- Run the simulation
- Update the *Verify State* in each requirement



ID	Description	?
R1	The approaching velocity of the load shall be 0.5 m/s.	
R2	The maximum force of cylinder can not exceed 200 kN	
R3	The pressing velocity must not exceed 10 mm/s	
R4	The cycle time shall not exceed 27.5 s	
R5	The returning velocity shall be 0.2m/s	
R6	The pressing position shall be 1300 mm, the allowable error is 1mm.	

Remarks

- Automatic update the *Verify State* is desired

Content

4

- Introduction
- Model Driven Requirements Engineering
- Case Study: Hydrostatic Press System
- **Conclusion and Outlook**

Conclusion and Outlook

Conclusions

- Engineering of industrial automation systems benefits from the MDRE
  - Formal requirements modeling
  - Standard modeling languages covering the whole development process
  - Automatic verification process
- The enhanced SysML requirement constructs of the MDRE4BR profile have been demonstrated by the case study

Outlook

- Finishing the implementation of the Modelica code generator considering traceability links
- Integrated tool support for the whole process

Thanks for your attentions!

Hongchao Ji

Bosch Rexroth AG
+ 49 (0) 9352 18 – 4764
Hongchao.Ji@boschrexroth.de

Rexroth
Bosch Group