Learning from Mechanics

Why MBSE is actually nothing new

Maria Eckey & Andreas Korff | Process Insights |
2025/03/20

MAHLE

MAHLE public (CL1)

How it Started …

MAHLE Production in the early 1920th

… what is **Today**

Automated Electronics Production

Leading International Supplier of Highly Advanced Solutions

… for Electrification, Thermal Management and Sustainable ICE

We Shape Future Mobility

Development partner to all major passenger and commercial vehicle OEM worldwide.

Systems supplier of crucial **powertrain** and **air conditioning** technologies with TOP 3 market position in core products.

Technological diversity for effective and fast climate protection.

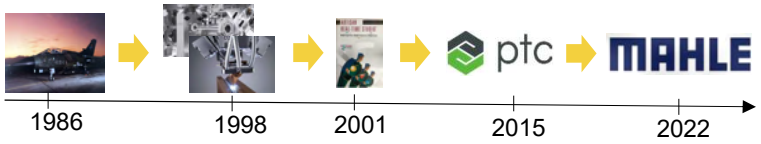
The Presenters

Andreas Korff

The guy always murmuring something about models

- Head of Systems and Software Engineering (SSE) at Mahle
- Dipl.-inform.(univ)
- Biased to MBSE and SSE
  - Working with SA-SD Models since 1991
  - Working with UML since 2001
  - Working with SysML since 2007

- (former) Member of OMG, INCOSE, GI, prostep ivip, DIN
- Bridging process, methods and tools for SSE with project needs
  - Pushing for standards whereever posible and useful
- Book Author on UML for Embedded Systems



Maria Eckey

Loves talking – especially about tools and ASPICE

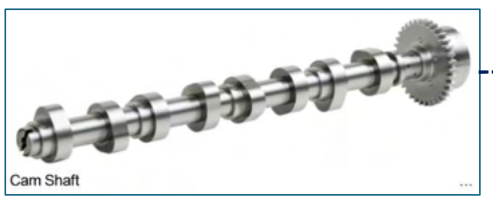
- Lead Consultant and People Lead at UL SIS
- MSc in Industrial Engineering
  - Production Sciences & IT
- ASPICE Competent Assessor & ALM Tool expert
- > 11 years supporting ASPICE compliant Systems Engineering
- Worldwide workshop series on ALM/PLM

UL Solutions

Software Intensive Systems

MAHLE public (CL1)

Heterogeneous Product Portfolio



MAHLE public (CL1)

The Process Challenge @ MAHLE

Heterogeneous Product Portfolio:
Avoid overkill from „One size fits All“, using the most complex process as template

Norms compliance for different domains & product types
Avoid development time overload

Approach
for developing integrated, tailorable engineering processes

Understand & formalize specific ways of working for the different product types (base to understand them as parts of an overall system)

Use Patterns to **find commonalities:**
Avoid „Divide and Conquer“ in the different Domains
Make processes **compatible** (if needed)

MAHLE public (CL1)

Understand & formalize

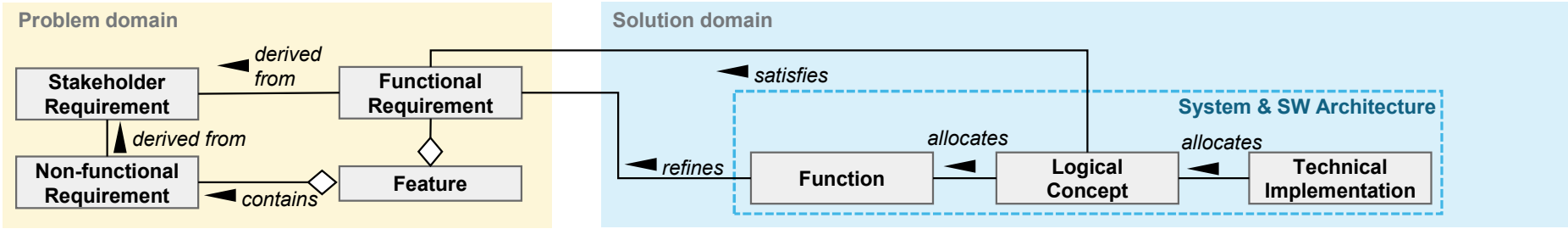
Problem domain

Stakeholder Requirement

Solution domain

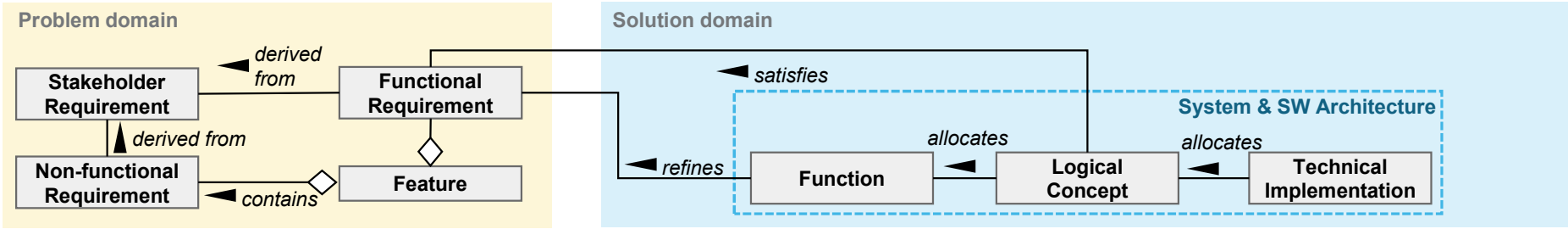
MAHLE public (CL1)

Understand & formalize



MAHLE public (CL1)

Understand & formalize



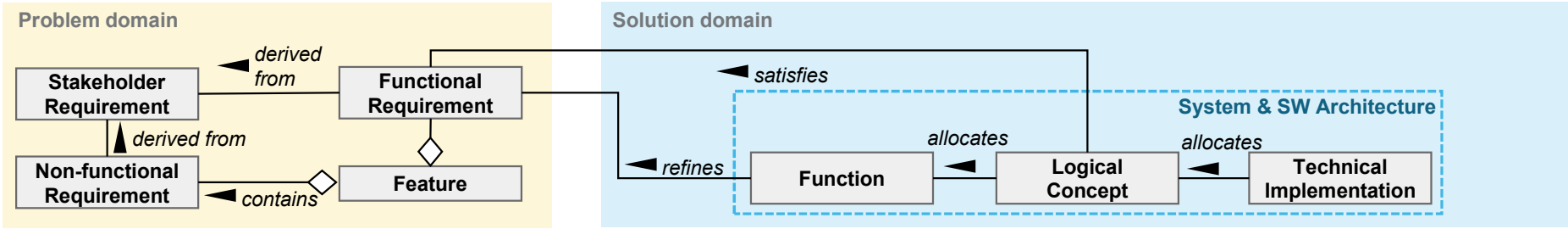
Systems Development

is often

- Project centric
- Anticipating high degree of freedom in solution space
- Starting with pure textual descriptions
- Focussing on Function

MAHLE public (CL1)

Understand & formalize



Systems Development

is often

- Project centric
- Anticipating high degree of freedom in solution space
- Starting with pure textual descriptions
- Focussing on Function

Mechanical Engineering

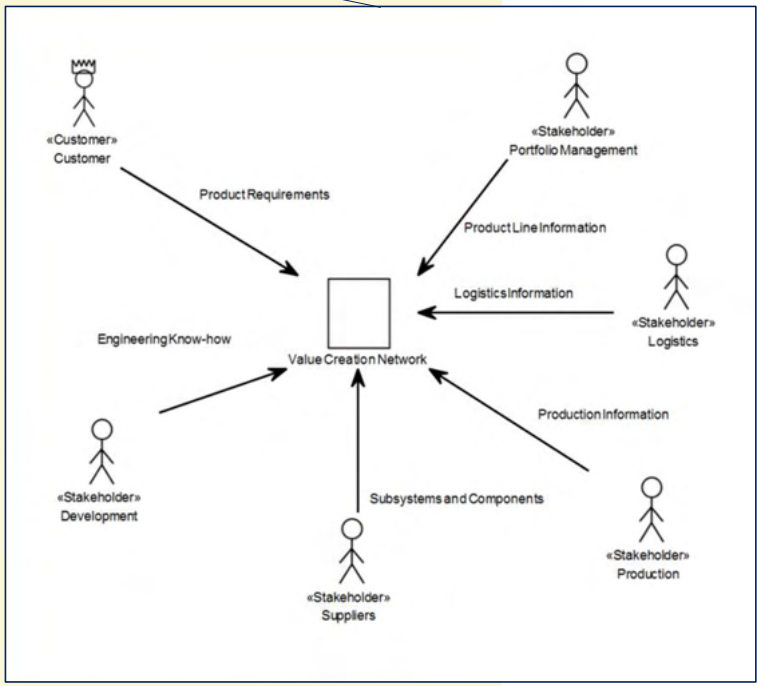
- Product centric

MAHLE public (CL1)

Understand & formalize

Problem domain

Stakeholder Requirement



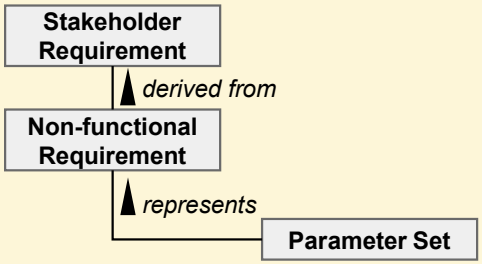
Mechanical Engineering

- Product centric
- Evolving: existing product adjusted, often as simulation model

MAHLE public (CL1)

Understand & formalize

Problem domain



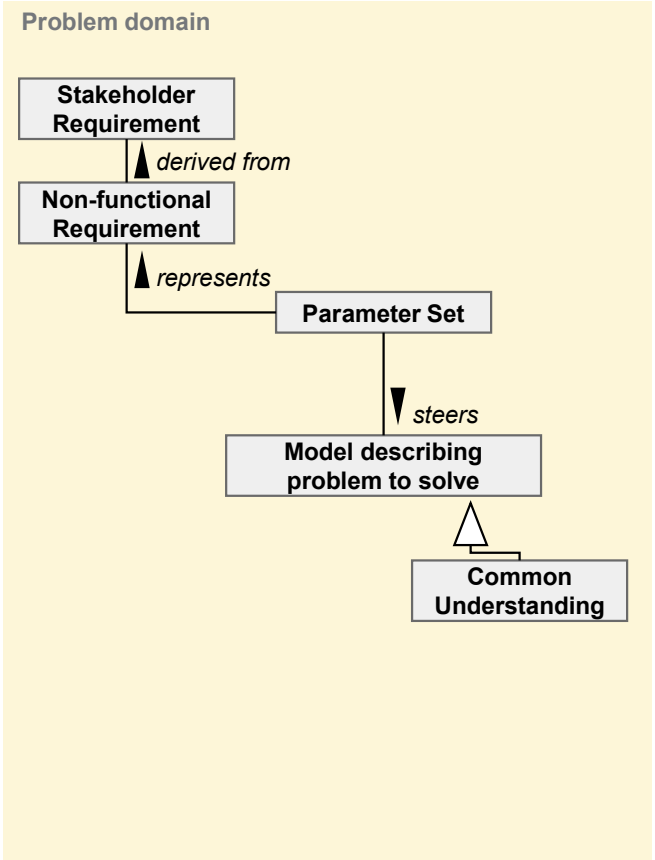
Why express the mechanical requirements with many words, instead of using a well-defined **Parametric exchange**?

Mechanical Engineering

- Product centric
- Evolving: existing product adjusted, often as simulation model
- Focussing on non-functional Constraints (expressed as parameters)

MAHLE public (CL1)

Understand & formalize



Why express the mechanical requirements with many words, instead of using a well-defined **Parametric exchange**?

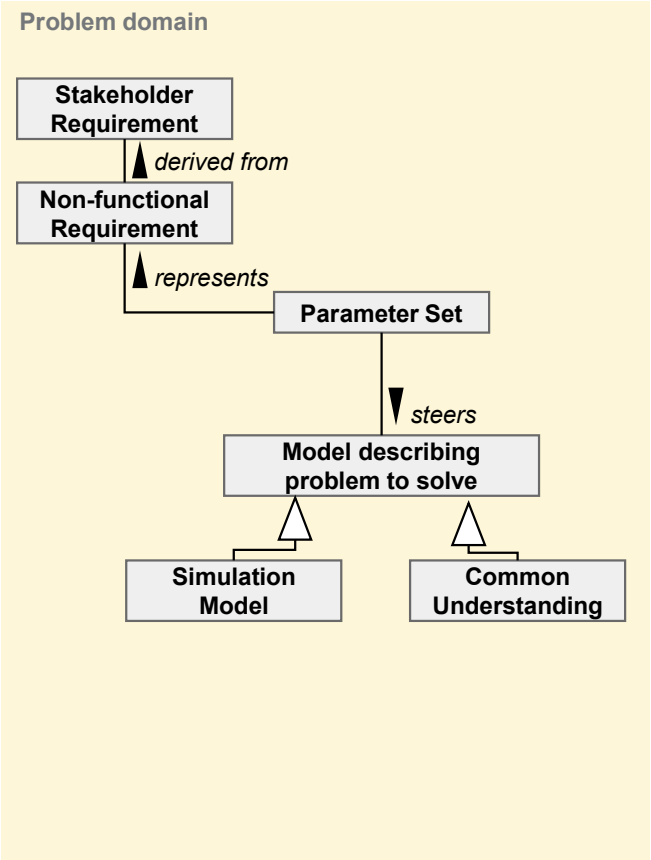
Mechanic Development Customers and Suppliers share a **common understanding** (i.e. a model) what is going to be delevoped

Mechanical Engineering

- Product centric
- Evolving: existing product adjusted, often as simulation model
- Focussing on non-functional Constraints (expressed as parameters)

MAHLE public (CL1)

Understand & formalize



Why express the mechanical requirements with many words, instead of using a well-defined **Parametric exchange**?

Mechanic Development Customers and Suppliers share a **common understanding** (i.e. a model) what is going to be delevoped

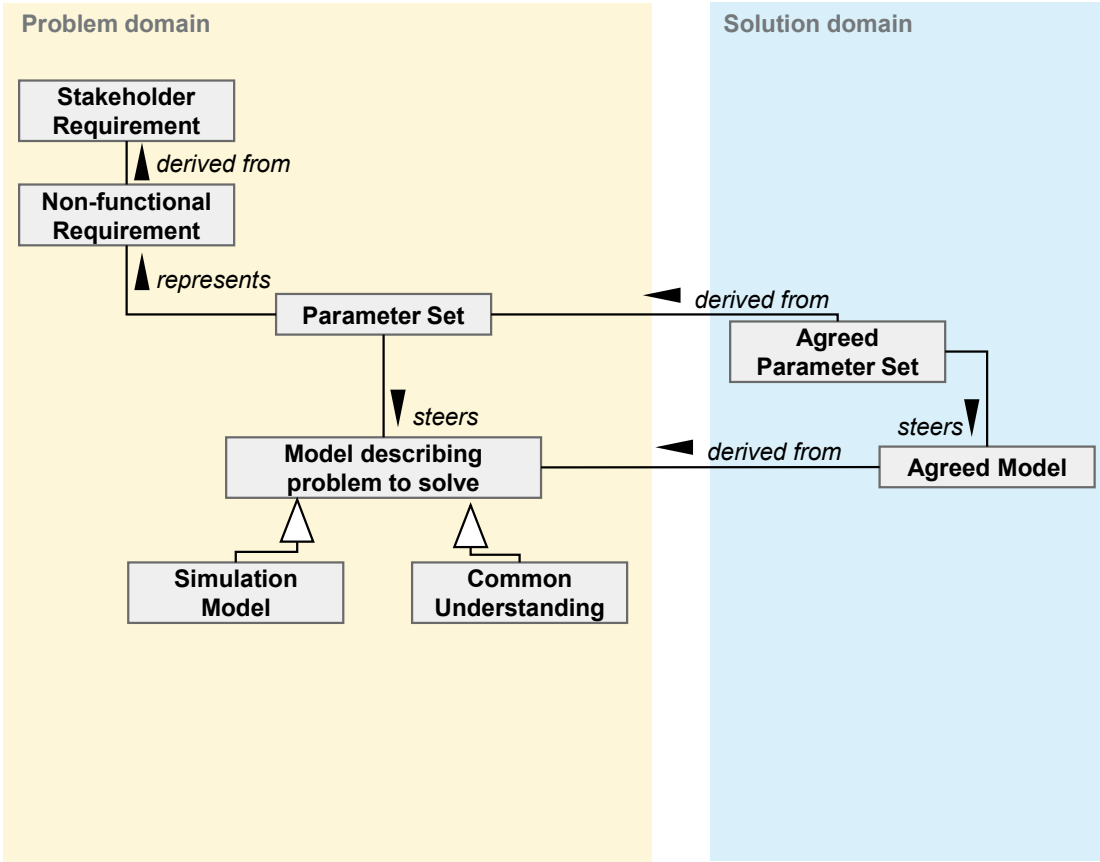
Allows to use **simulation models** as soon as possible with the agreed set of parameters (Validate Early)

Mechanical Engineering

- Product centric
- Evolving: existing product adjusted, often as simulation model
- Focussing on non-functional Constraints (expressed as parameters)

MAHLE public (CL1)

Understand & formalize

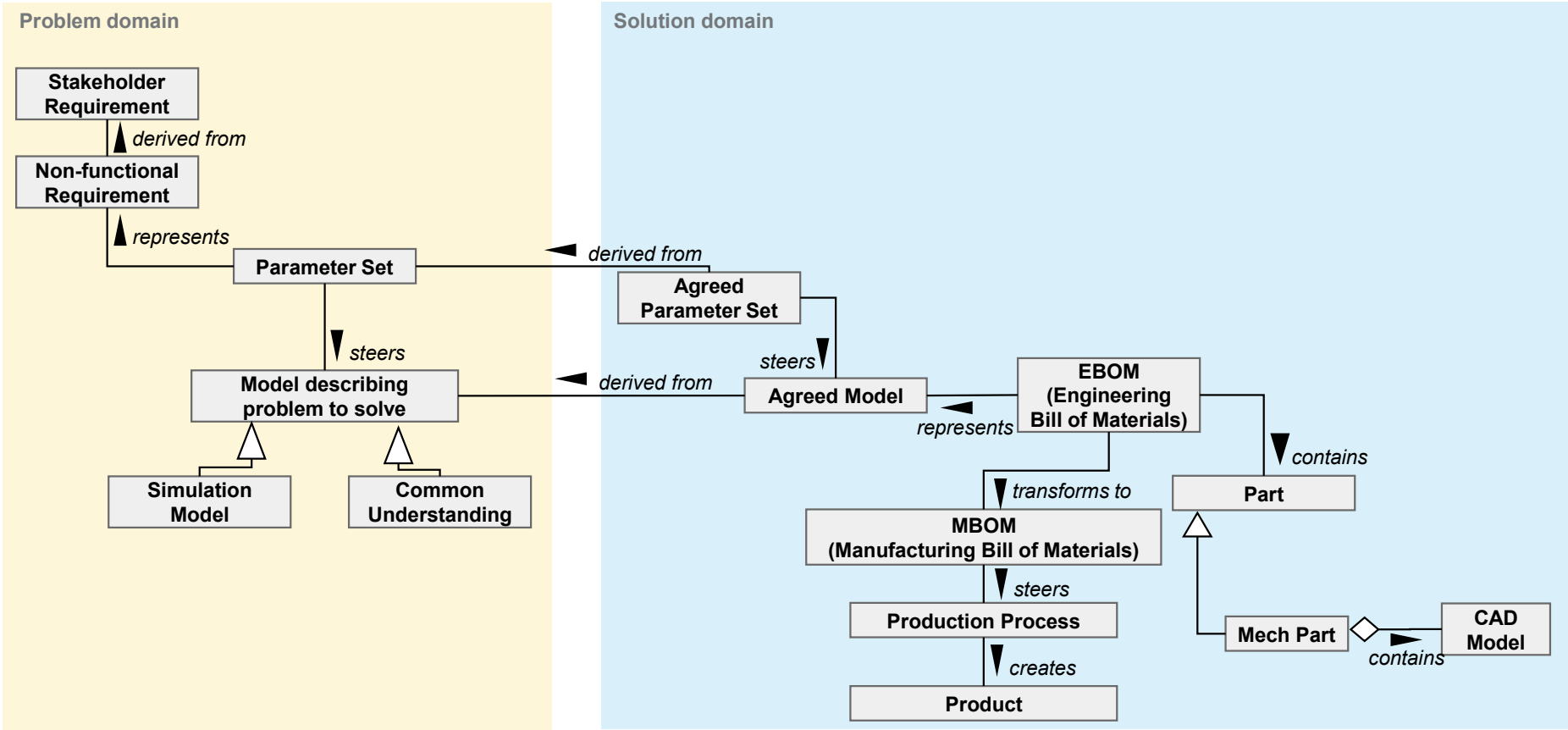


Mechanical Engineering

- Product centric
- Evolving: existing product adjusted, often as simulation model
- Focussing on non-functional Constraints (expressed as parameters)
- System Specification = agreed model & set of parameters

MAHLE public (CL1)

Understand & formalize



MAHLE public (CL1)

Build common processes

Understand & formalize specific ways of working for the different product types (base to undestand them as parts of an overall system)

Use Patterns to **find commonalities:**
Avoid „Divide and Conquer“ in the different Domains
Make processes **compatible** (if needed)

Systems Development

is often
- Project centric
- Anticipating high degree of freedom in solution space
- Starting with pure textual descriptions
- Focussing on Function

Mechanical Engineering

- Product centric
- Evolving: existing product adjusted, often as simulation model
- Focussing on non-functional Constraints (expressed as parameters)
- System Specification = agreed model & set of parameters

MAHLE public (CL1)

Some commonalities

Use Patterns to **find commonalities:**
Avoid „Divide and Conquer“ in the different Domains
Make processes **compatible** (if needed)

MAHLE public (CL1)

Some commonalities

Use Patterns to **find commonalities:**
Avoid „Divide and Conquer“ in the different Domains
Make processes **compatible** (if needed)

Non-functional Requirements define the acceptable **Solution Space**

Need for **abstraction** (i.e. modeling) - Parameters and Parameter ranges:

- The ECU shall start in max. 0.2s
- The piston diameter shall be 85 mm at 50°C

Strict **Configuration Management**

- FFF-Versioning Rules with numbering logic in the eBOM and mBOM
- Engineering Change Requests (ECR)

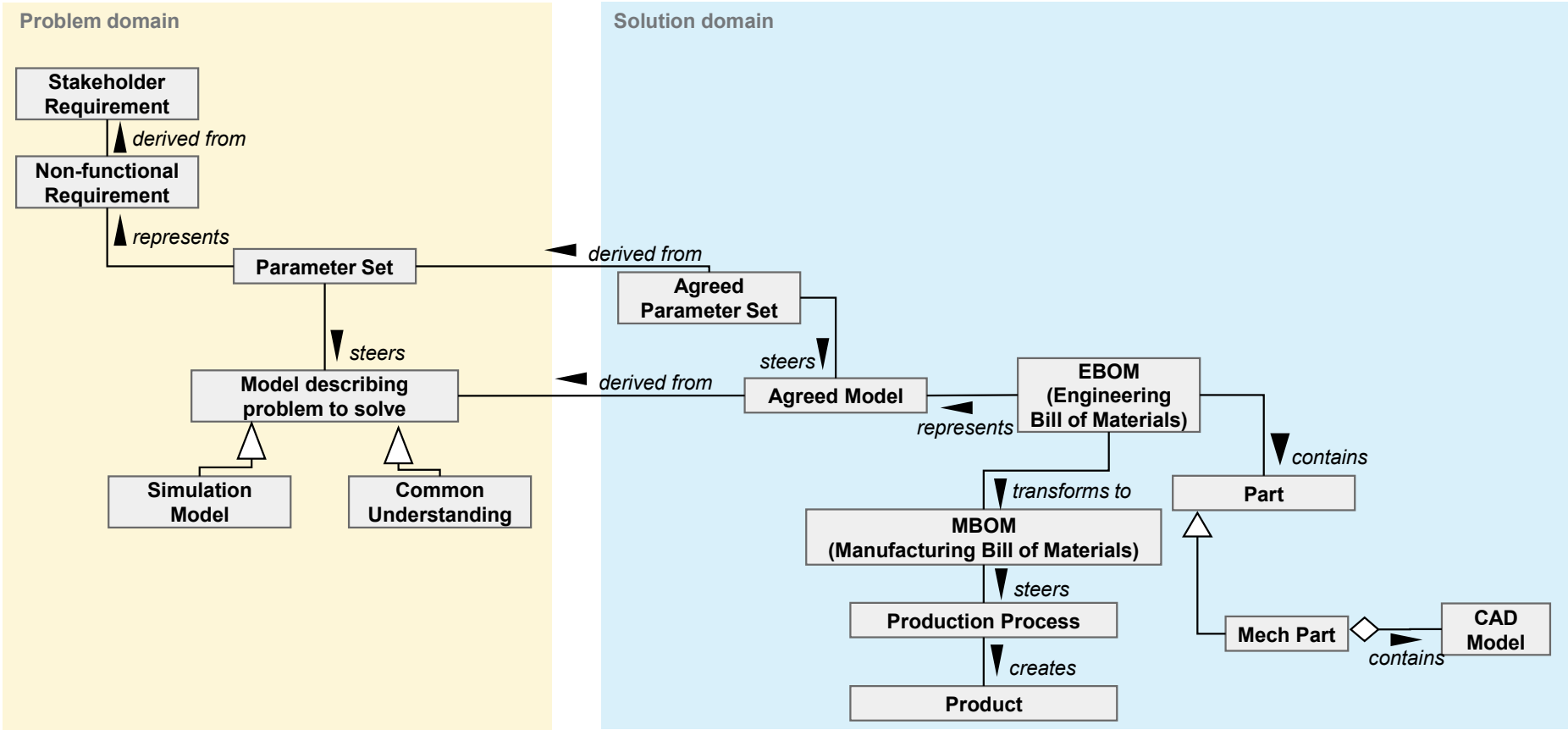
Traceability in Mechanical Engineering
Parameter → Part Parameter → Parametric CAD model

Decomposition

- A mechanical Assembly is a set of Parts, which can also be Assemblies
- A System is a set of subsystems, which can also be systems

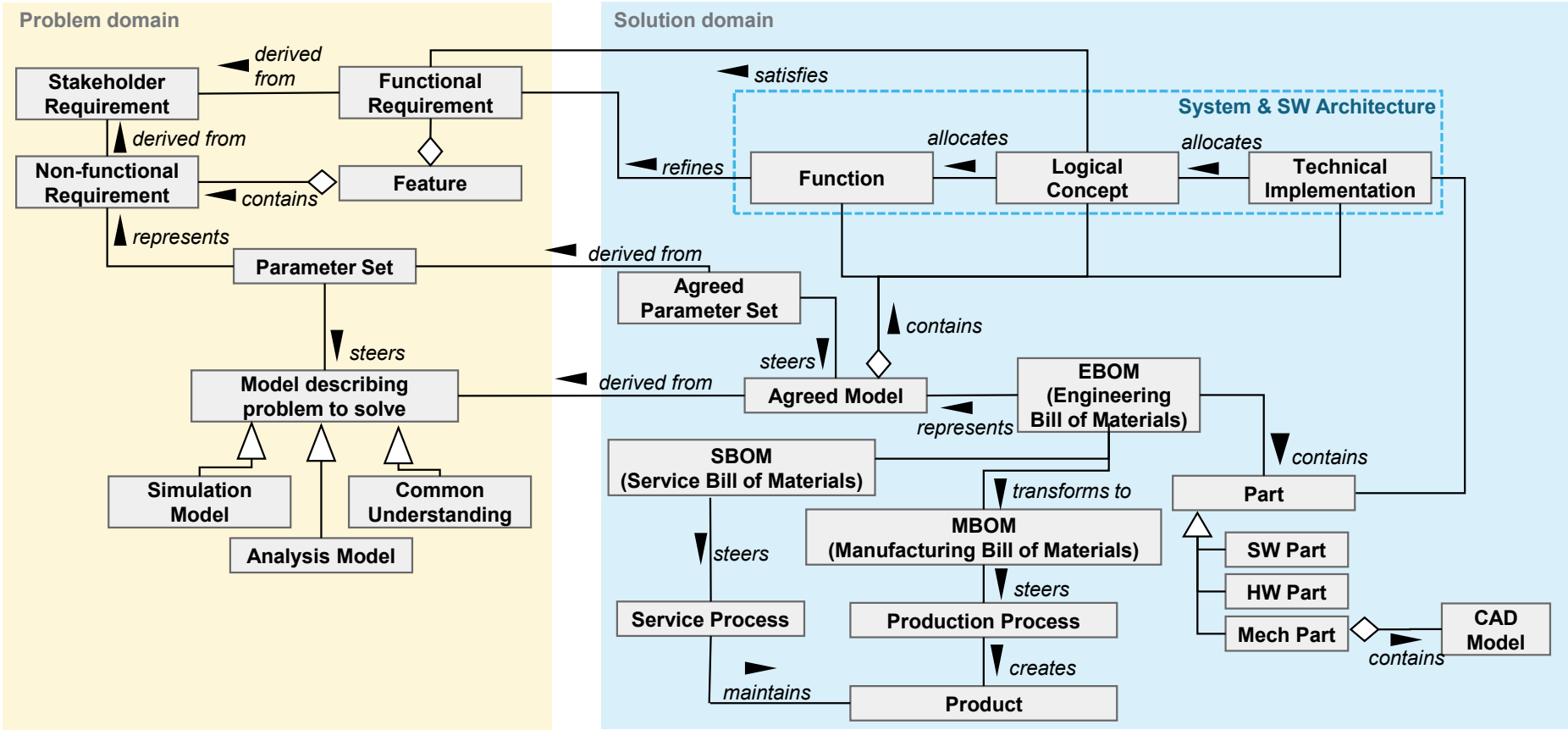
MAHLE public (CL1)

Build common processes



MAHLE public (CL1)

Build common processes



MAHLE public (CL1)

Summary of our Example

Important Aspects in MBSE are already established in Mechanics

Mostly Evolution instead of Revolution
- Innovation is needed, but must be a concious decision (expensive!)

Focus on Product (Lines) instead of a Project

Holistic Perspective long established in Mechanics (a bit different from SE):
→ Value Network Perspective

A viable solution is defined by the **non-functional constraints**

Add the **functional View** to cope with **Complexity and „Software-driven“ Systems**
→ Exist implicitely or as Simulation Models in the mechanical World

MAHLE public (CL1)

Process Learnings

Avoid Divide and Conquer as a basic principle
Commonalities are crucial for a viable solution

Avoid Process Overload
The „One Size fits all“ Process does not exist
Tailoring with one switch „SE involved y/n“ doesn‘t play the game

A Bridge between Mechanical and Systems Engineering
Harvest from experience
Things might look different, but the underlying concepts are very similar

Stakeholder Requirements Elicitation
must be complete
Mechanic Projects know them very well
They all show up, sometimes too late

MAHLE public (CL1)

Thank you – questions?

MAHLE