Collaborative Embedded Systems

CrESt

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

CrESt Projektabschluss

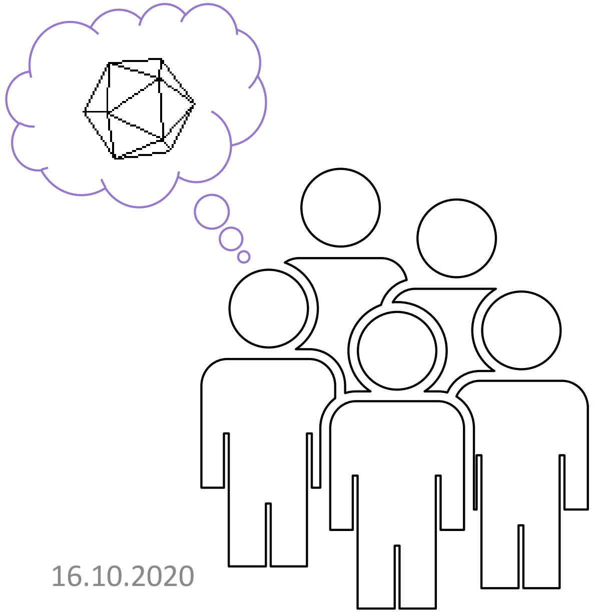
Context-sensitive Reconfiguration

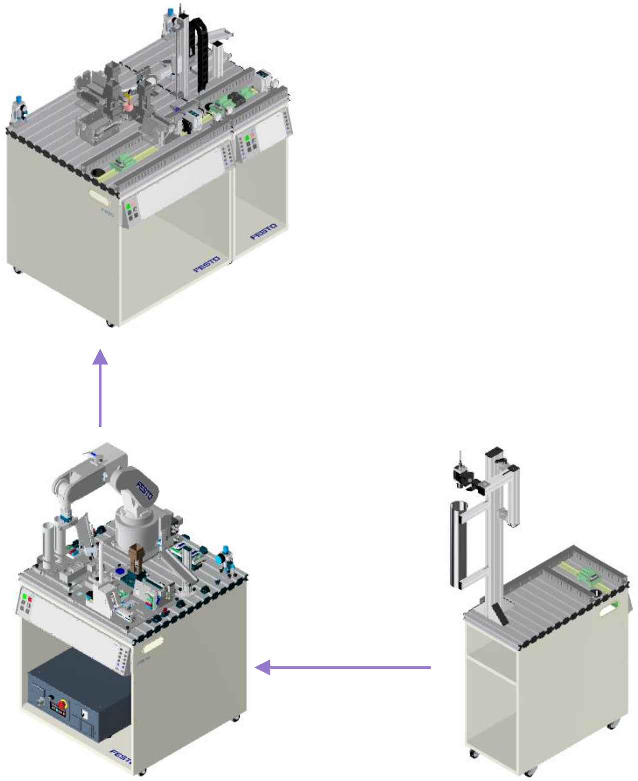
Birte Caesar / Helmut-Schmidt Universität

Ina Schaefer / Technische Universität Braunschweig

Motivation

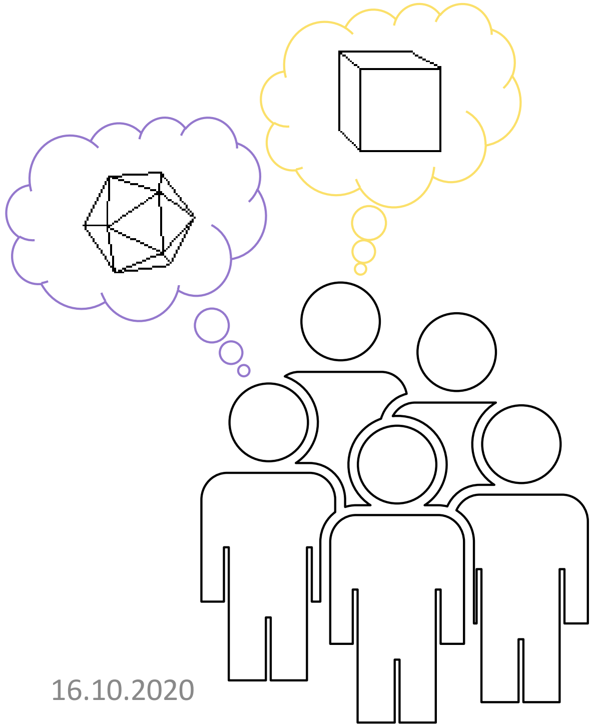
- Varying production requests → different manufacturing system configurations

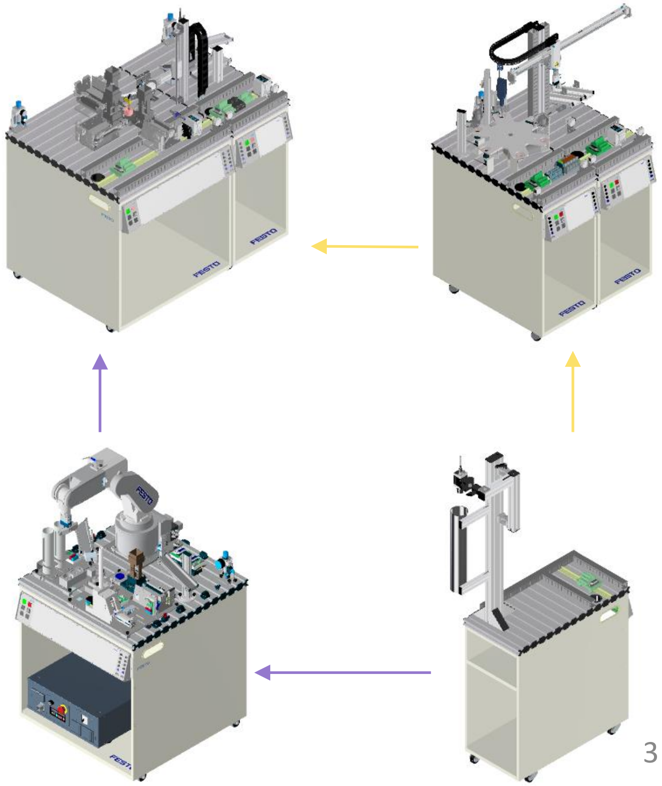




Motivation

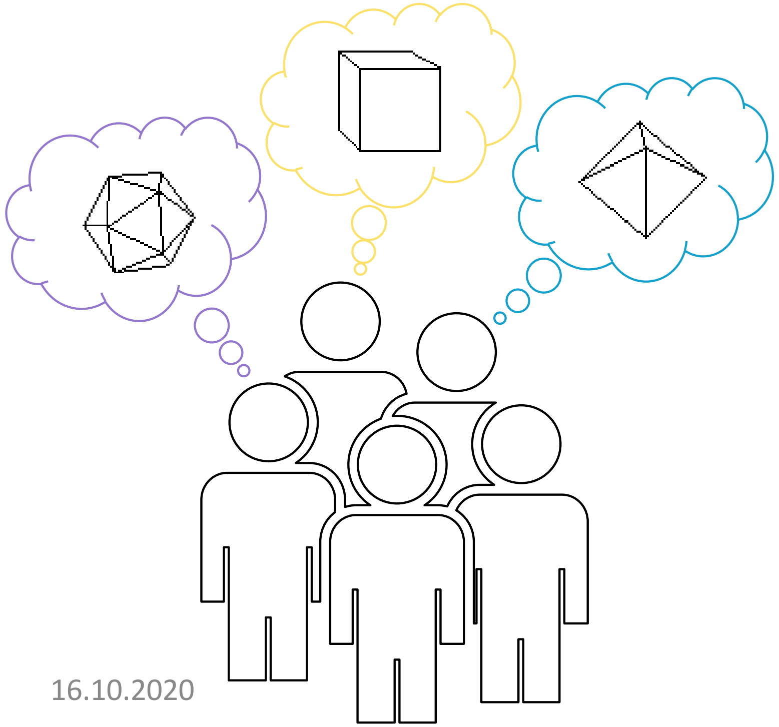
- Varying production requests → different manufacturing system configurations

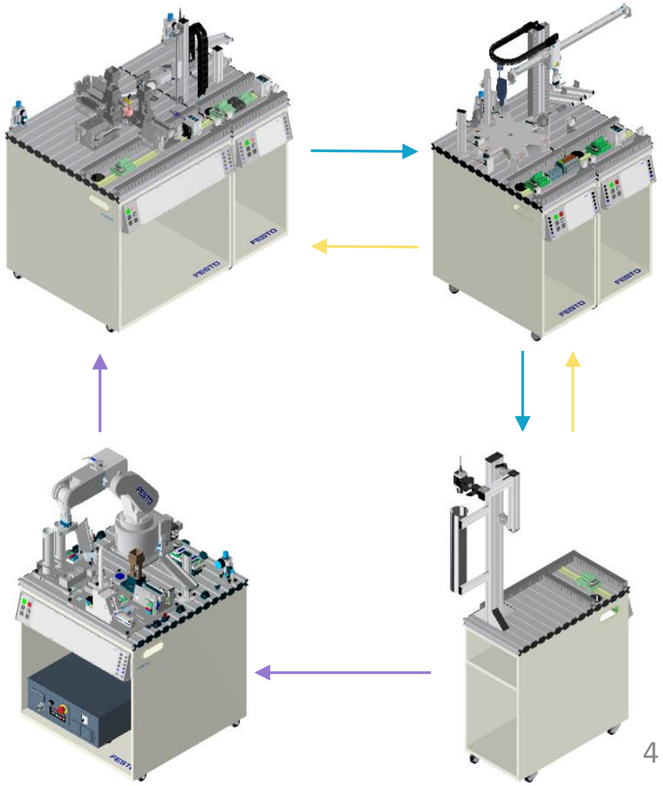




Motivation

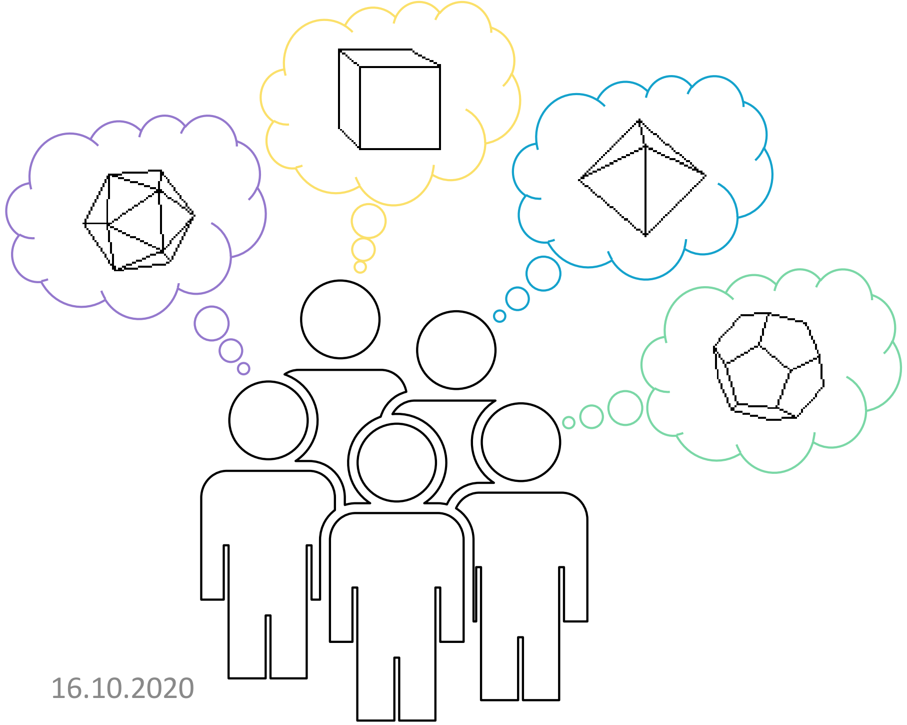
- Varying production requests → different manufacturing system configurations

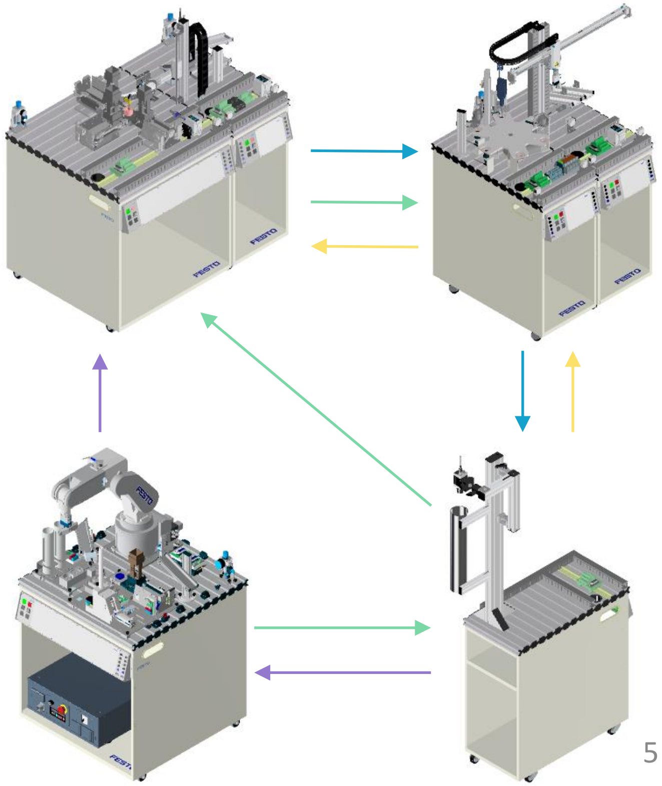




Motivation

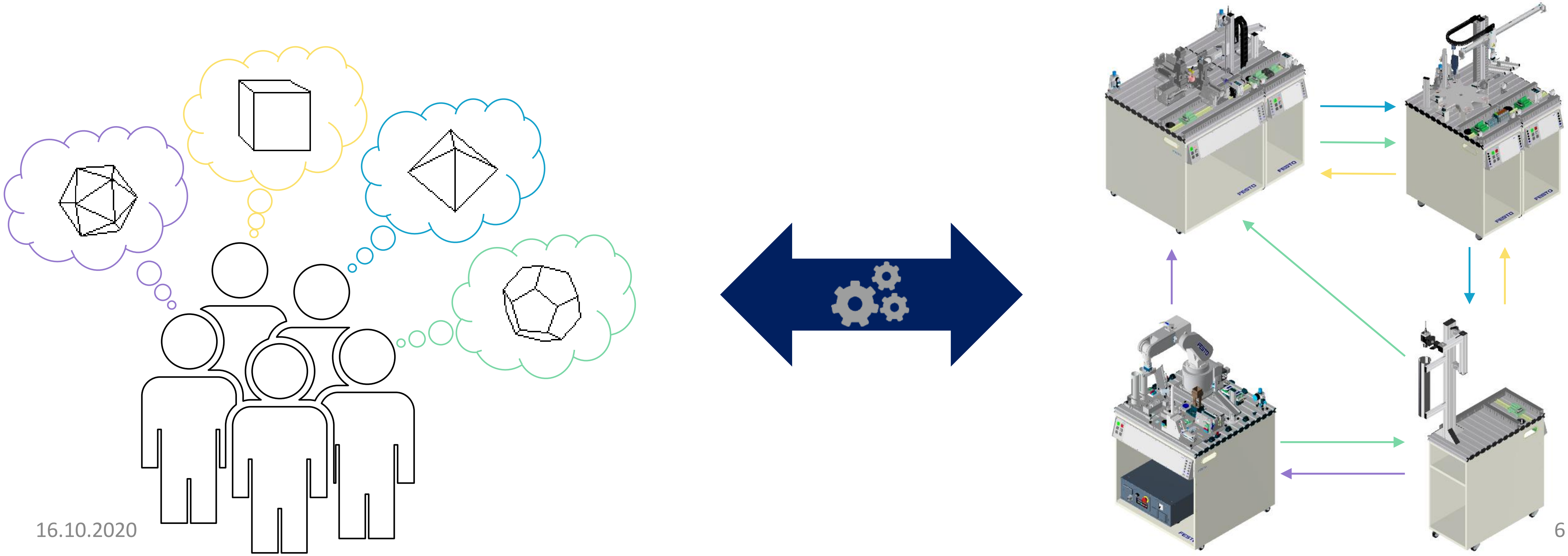
- Varying production requests → different manufacturing system configurations





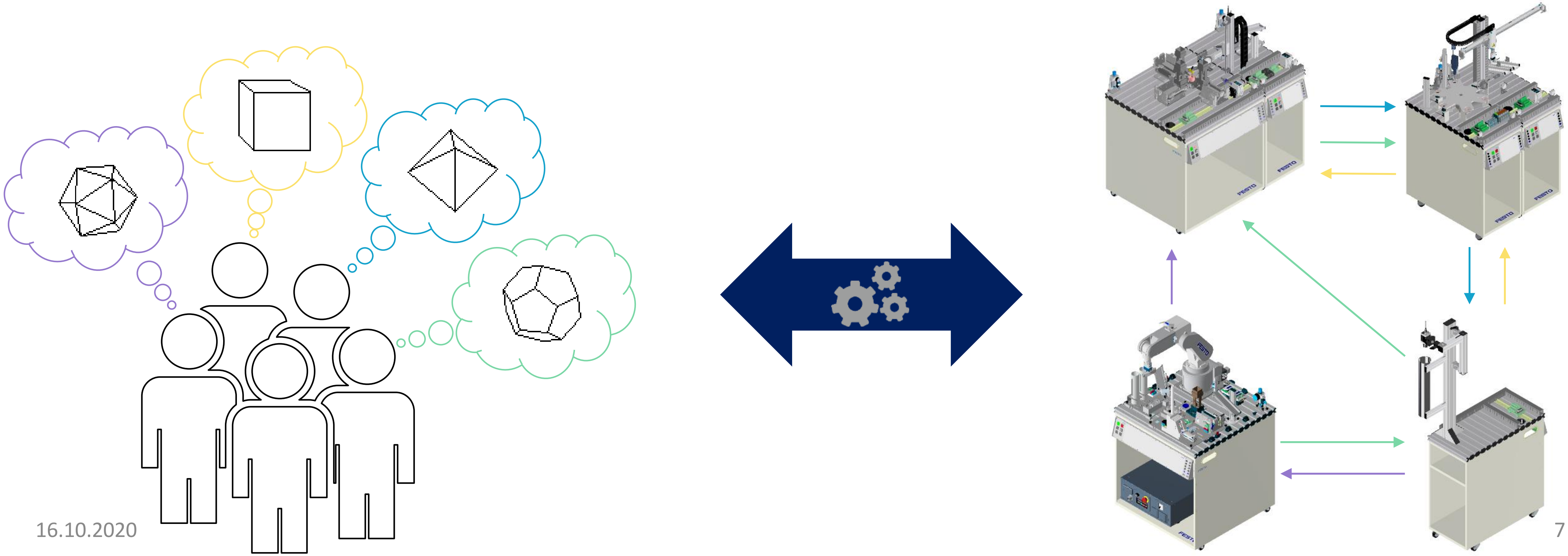
Motivation

- Varying production requests → different manufacturing system configurations
- Extract requirements for new production requests



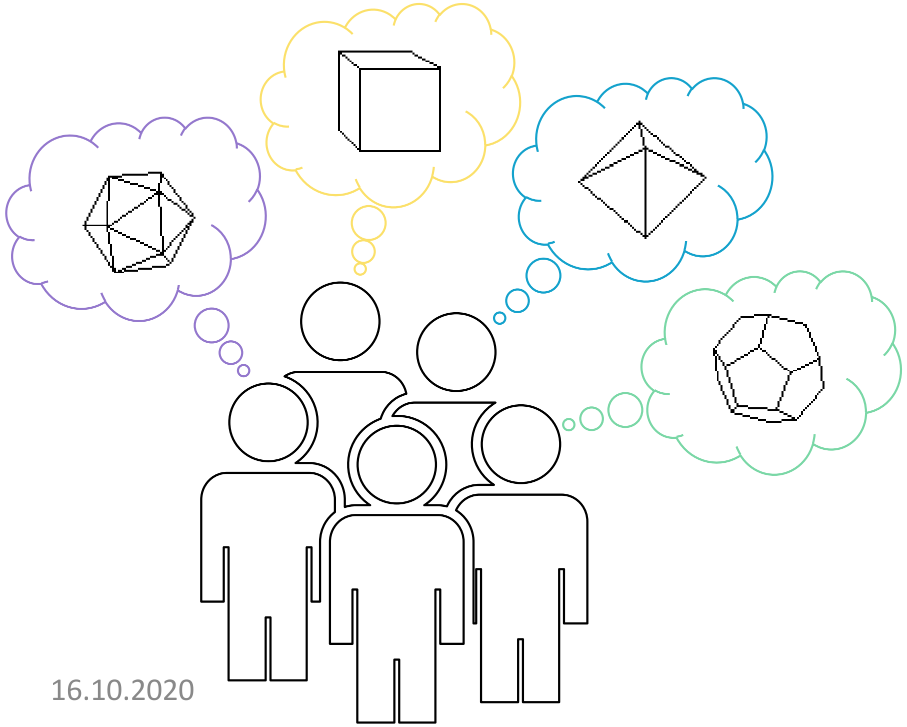
Motivation

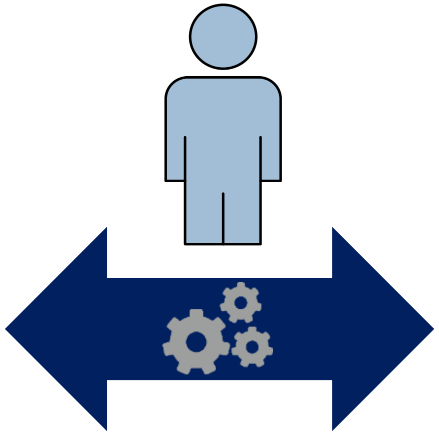
- Varying production requests → different manufacturing system configurations
- Extract requirements for new production requests
- Evaluate whether the manufacturing system can fulfill requirements

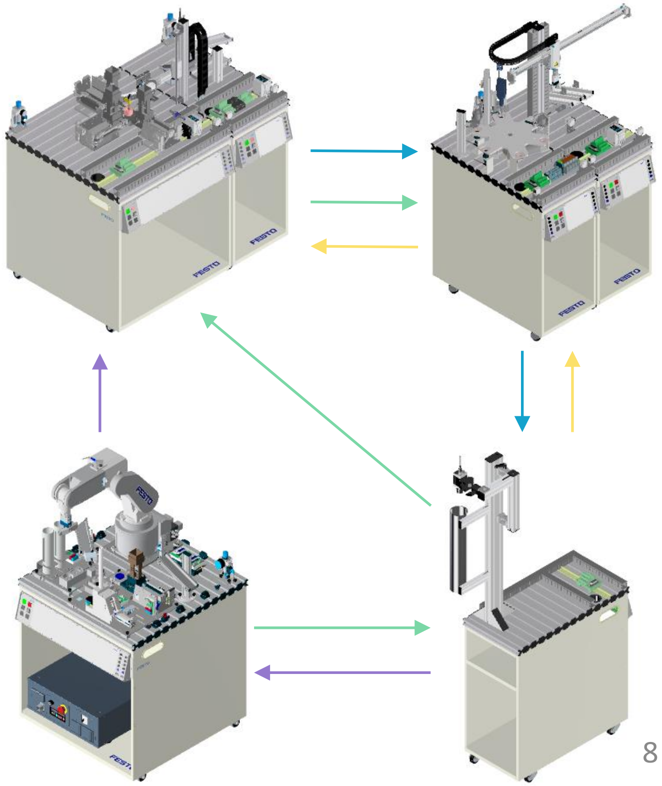


Motivation

- Varying production requests → different manufacturing system configurations
- Extract requirements for new production requests
- Evaluate whether the manufacturing system can fulfill requirements

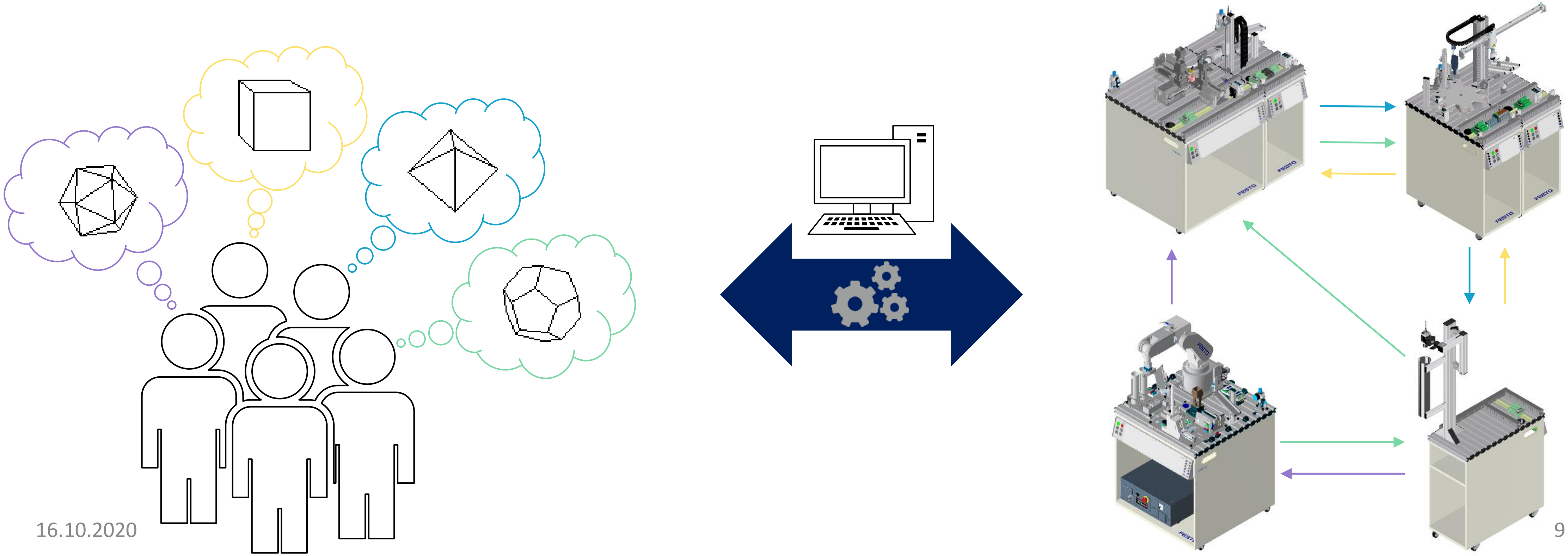






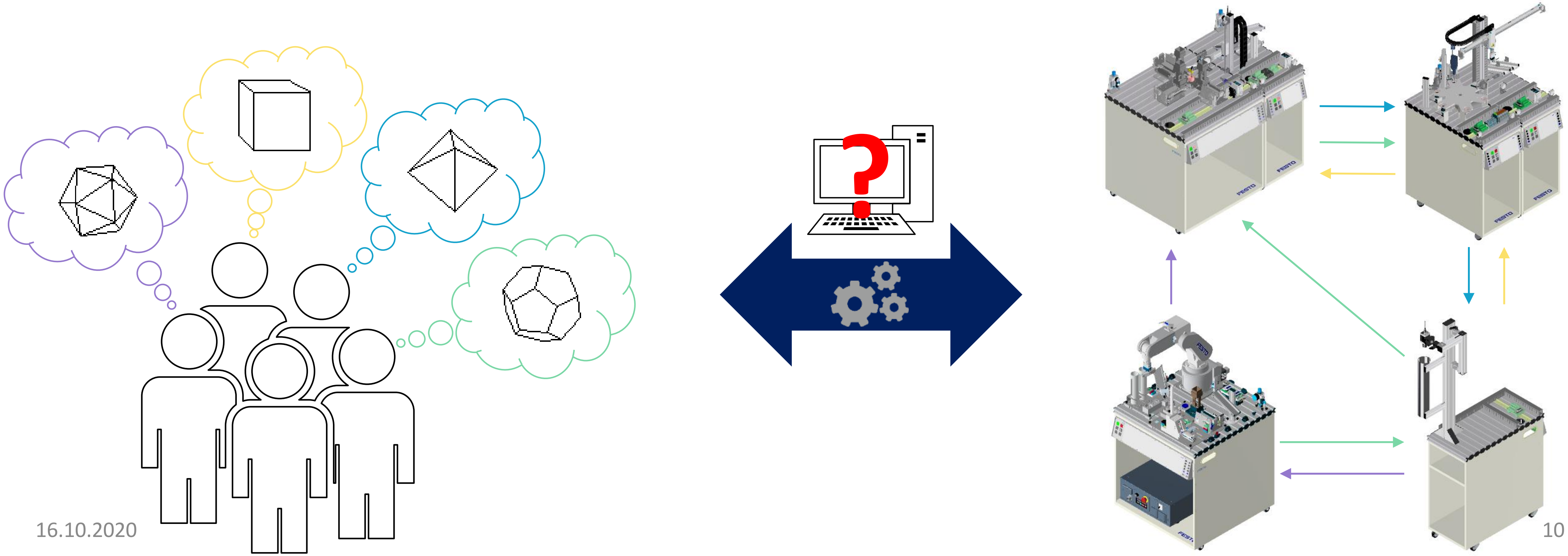
Motivation

- Varying production requests → different manufacturing system configurations
- Extract requirements for new production requests
- Evaluate whether the manufacturing system can fulfill requirements



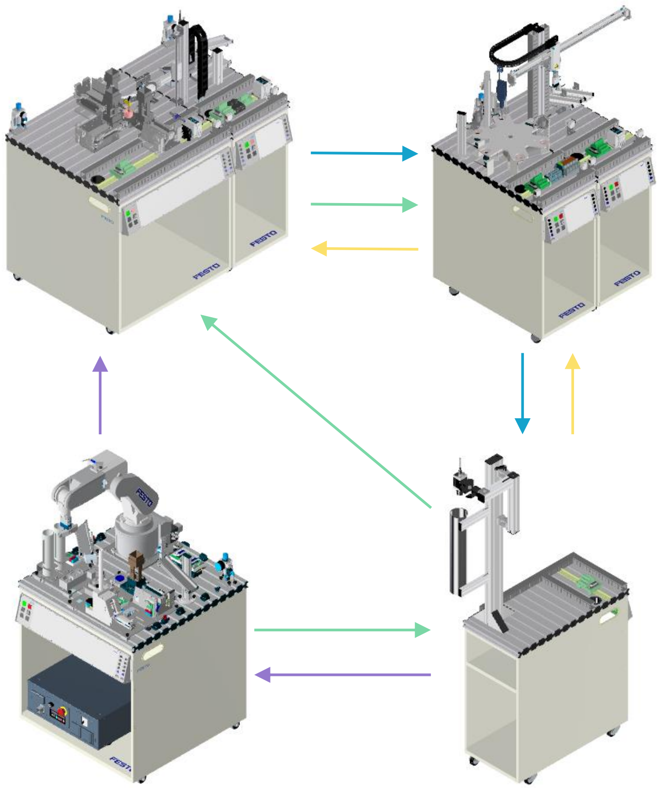
Motivation

- Varying production requests → different manufacturing system configurations
- Extract requirements for new production requests
- Evaluate whether the manufacturing system can fulfill requirements



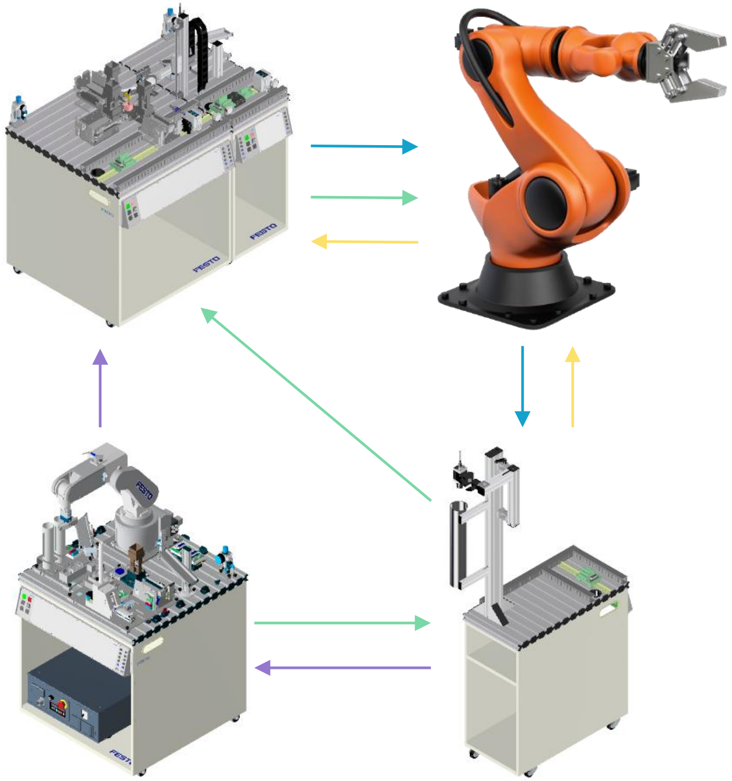
Motivation

- A Manufacturing system is composed of modules



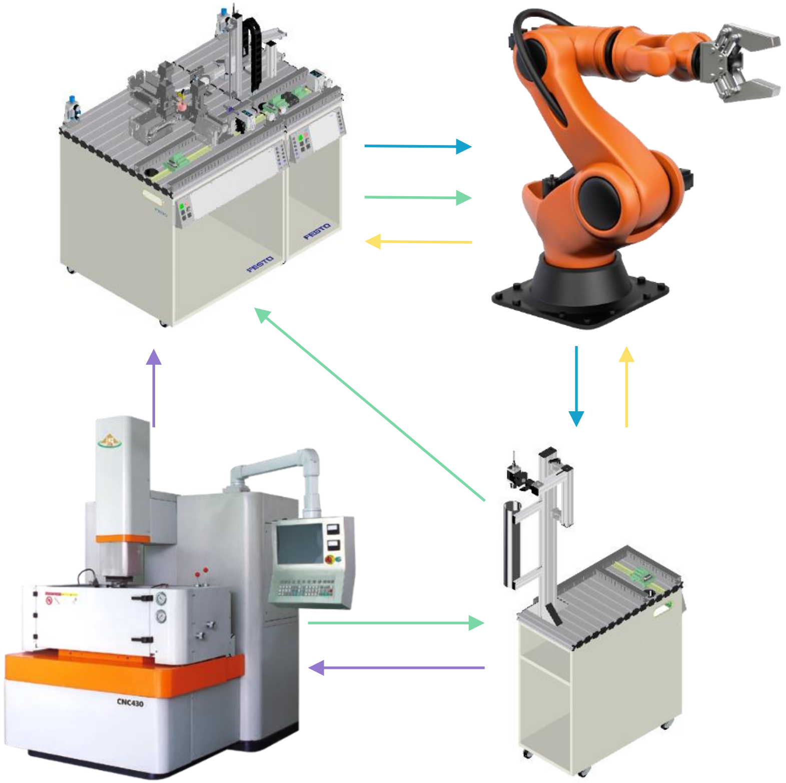
Motivation

- A Manufacturing system is composed of modules
- Modules are exchangeable



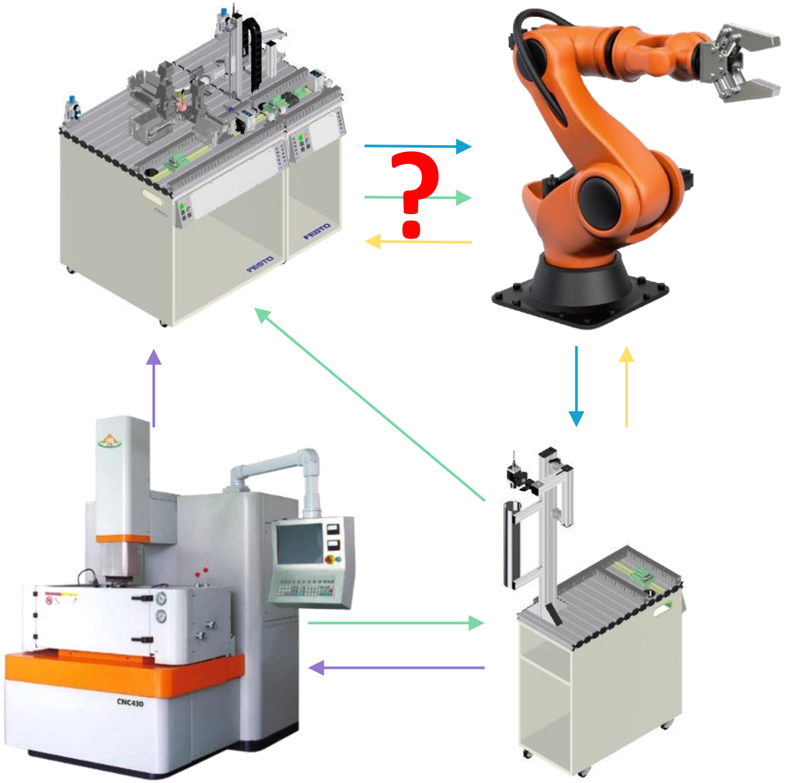
Motivation

- A Manufacturing system is composed of modules
- Modules are exchangeable



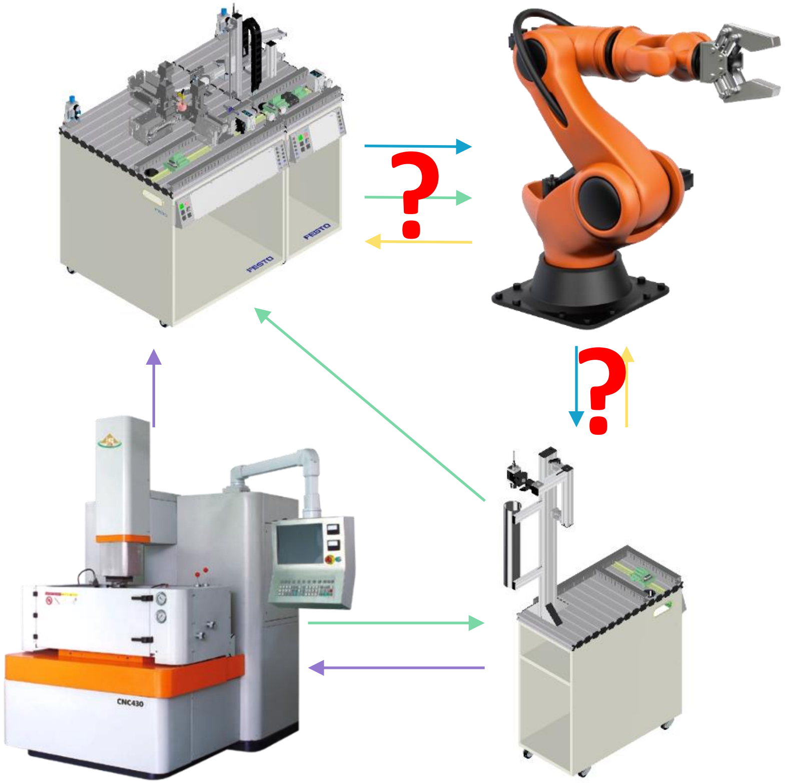
Motivation

- A Manufacturing system is composed of modules
- Modules are exchangeable



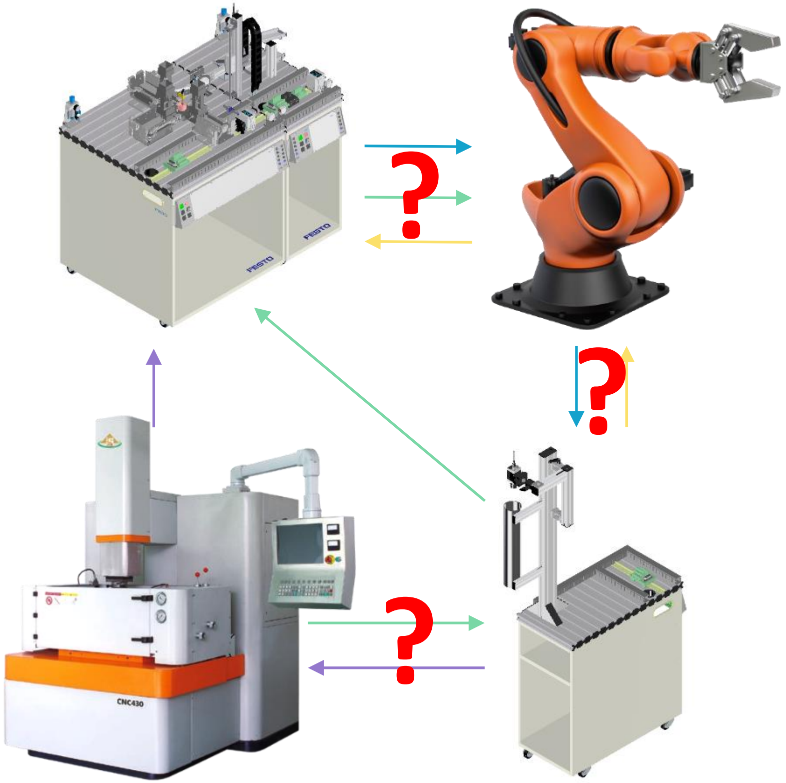
Motivation

- A Manufacturing system is composed of modules
- Modules are exchangeable



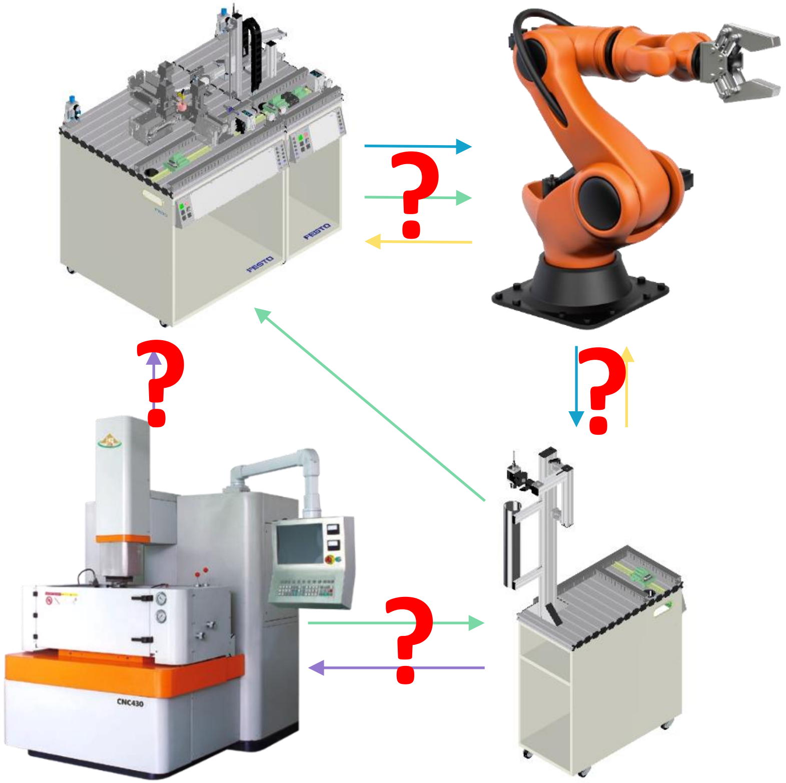
Motivation

- A Manufacturing system is composed of modules
- Modules are exchangeable



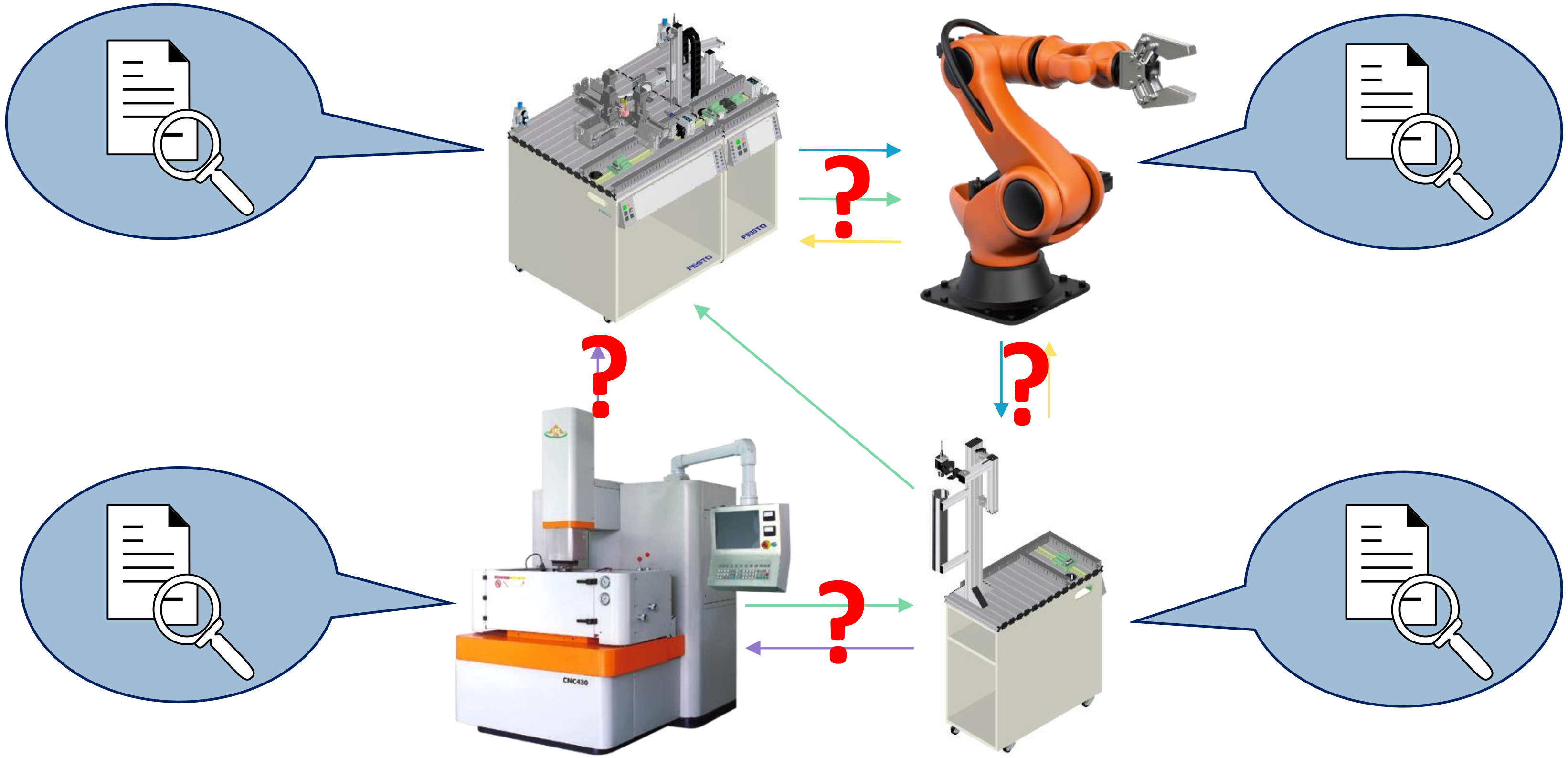
Motivation

- A Manufacturing system is composed of modules
- Modules are exchangeable



Motivation

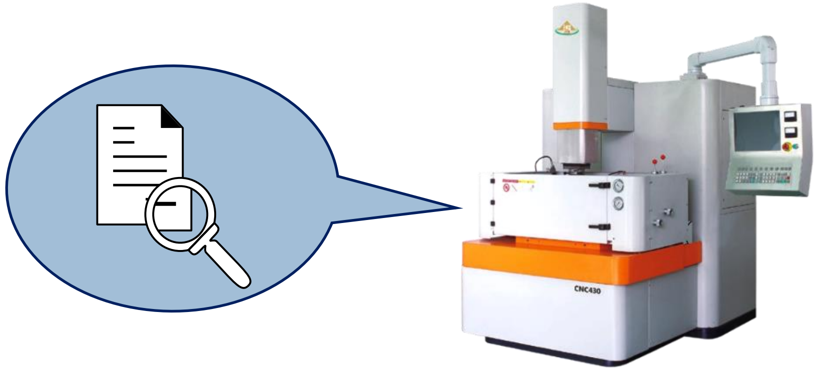
- A Manufacturing system is composed of modules
- Modules are exchangeable
- Module capabilities must be communicated to evaluate producability



Motivation

- Manufacturing system is composed of modules
- Modules are exchangeable
- Module capabilities must be communicated to evaluate producability

⇨ How to model the capability description of each module?

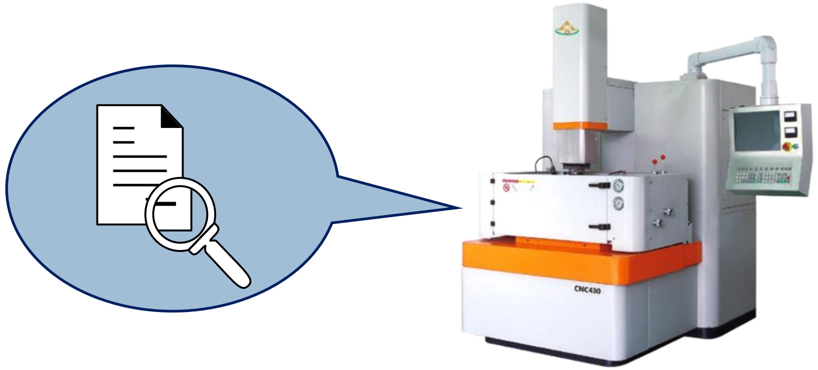


Motivation

- Manufacturing system is composed of modules
- Modules are exchangeable
- Module capabilities must be communicated to evaluate producability

⇨ How to model the capability description of each module?

- Modules can be configured → changing capabilities



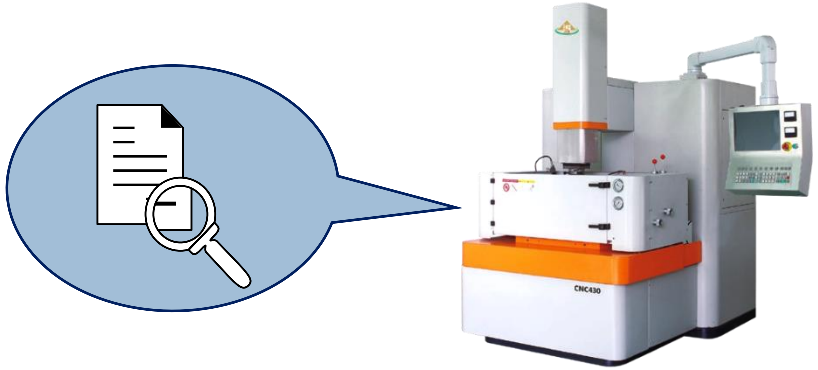
Motivation

- Manufacturing system is composed of modules
- Modules are exchangeable
- Module capabilities must be communicated to evaluate producability

➡ How to model the capability description of each module?

- Modules can be configured → changing capabilities

➡ How to select the right configuration of a module?



Method - Overview

How to model the capability description of each module?

- Ontology Usage for Capability Description

How to select the right configuration of a module?

- Context-sensitive-Variability Models

Introduction to Ontology

“An ontology may take a variety of forms, but necessarily it will include a ***vocabulary of terms,*** and some ***specification of their meaning***. This includes definitions and an **indication of how concepts are inter-related** which collectively impose a structure on the domain and constrain the possible interpretations of terms”
(R. Jasper & M. Uschold, 1999)

- Ontologies are graphs
- The terms and relations (including attributes) form the so-called terminology box (T-Box)
- The axioms (i.e., individuals of the ontology) form the assertional box (A-Box) which conform to the T-Box

Ontology Usage for Capability Description

Why use ontologies?

- High number of interacting systems?
- Open context?
- Complexity and dynamicity of exchanged data?
- Long-term usage?

Ontology Usage for Capability Description

Why use ontologies?

- ✓ High number of interacting systems?
- Open context?
- Complexity and dynamicity of exchanged data?
- Long-term usage?

Ontology Usage for Capability Description

Why use ontologies?

✓ High number of interacting systems?

✓ Open context?

- Complexity and dynamicity of exchanged data?
- Long-term usage?

Ontology Usage for Capability Description

Why use ontologies?

✓ High number of interacting systems?

✓ Open context?

✓ Complexity and dynamicity of exchanged data?

- Long-term usage?

Ontology Usage for Capability Description

Why use ontologies?

✓ High number of interacting systems?

✓ Open context?

✓ Complexity and dynamicity of exchanged data?

✓ Long-term usage?

Ontology Usage for Capability Description

Why use ontologies?

- ✓ High number of interacting systems?
- ✓ Open context?
- ✓ Complexity and dynamicity of exchanged data?
- ✓ Long-term usage?

Terminological Box (T-Box) for capability description

Ontology Usage for Capability Description

Why use ontologies?

✓ High number of interacting systems?

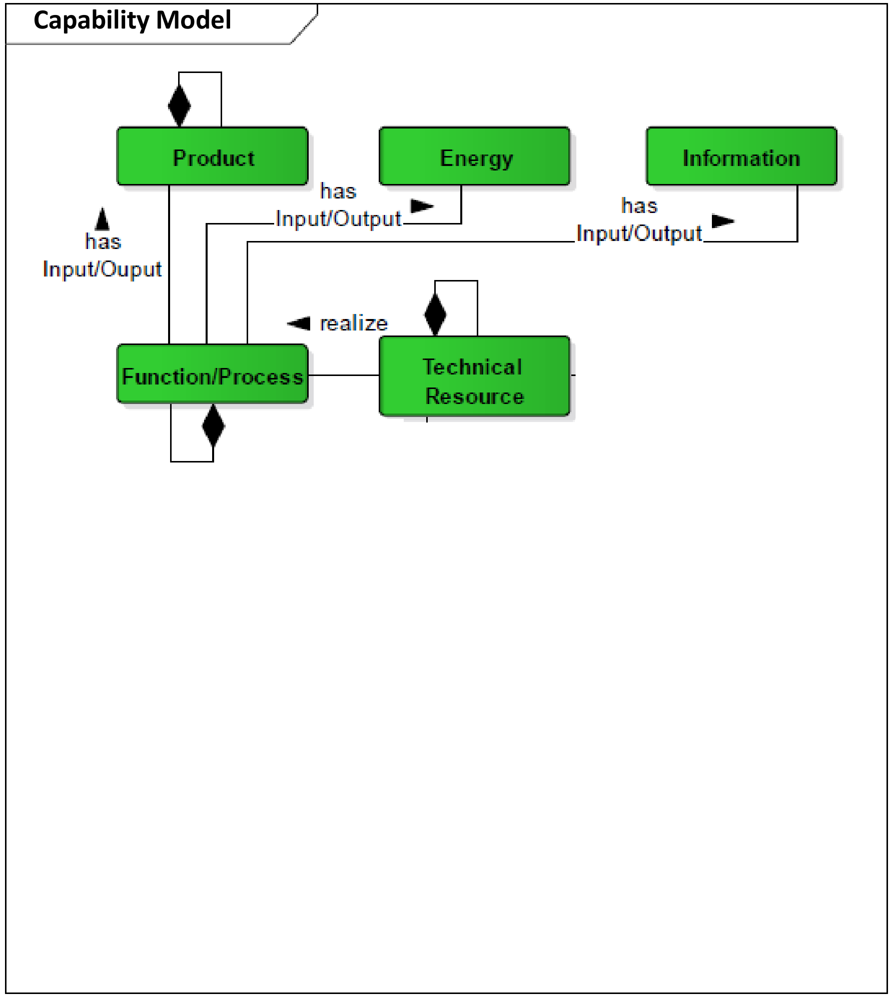
✓ Open context?

✓ Complexity and dynamicity of exchanged data?

✓ Long-term usage?

Terminological Box (T-Box) for capability description

- Function (VDI/VDE 3682-2): Transformation of input to output



Ontology Usage for Capability Description

Why use ontologies?

✓ High number of interacting systems?

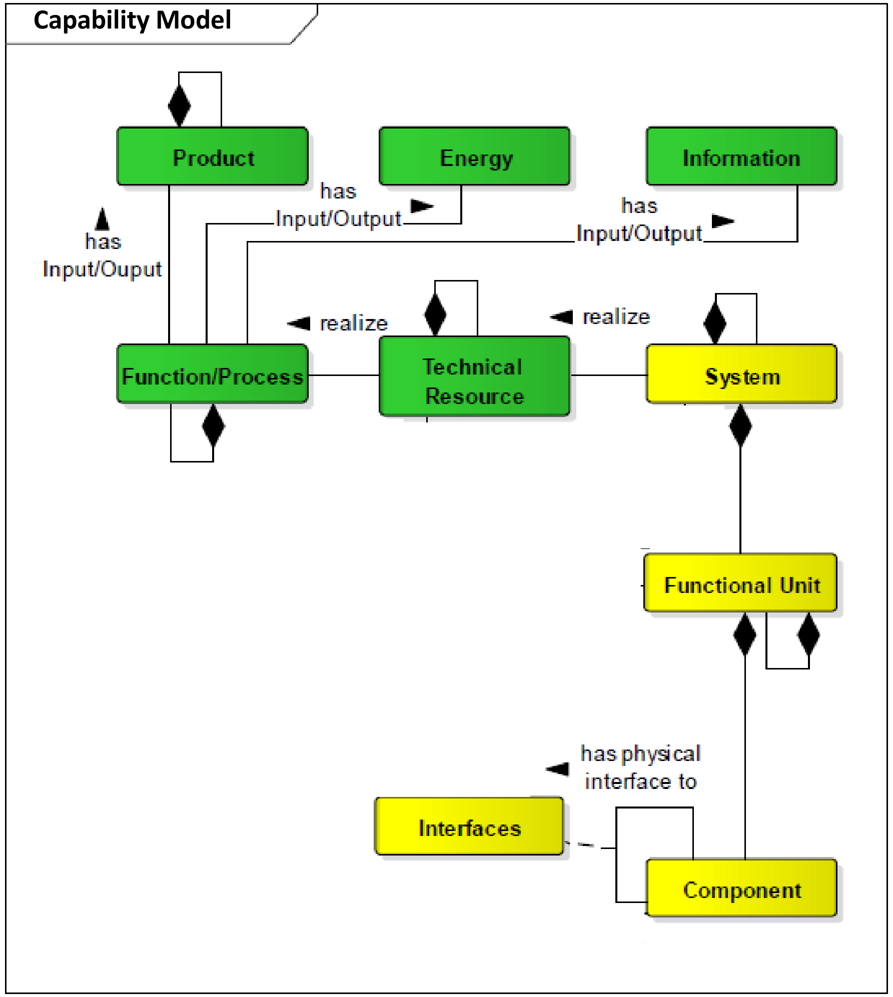
✓ Open context?

✓ Complexity and dynamicity of exchanged data?

✓ Long-term usage?

Terminological Box (T-Box) for capability description

- Function (VDI/VDE 3682-2): Transformation of input to output
- Structure (IEC 62264-1): Components as well as their relations to other components



Ontology Usage for Capability Description

Why use ontologies?

✓ High number of interacting systems?

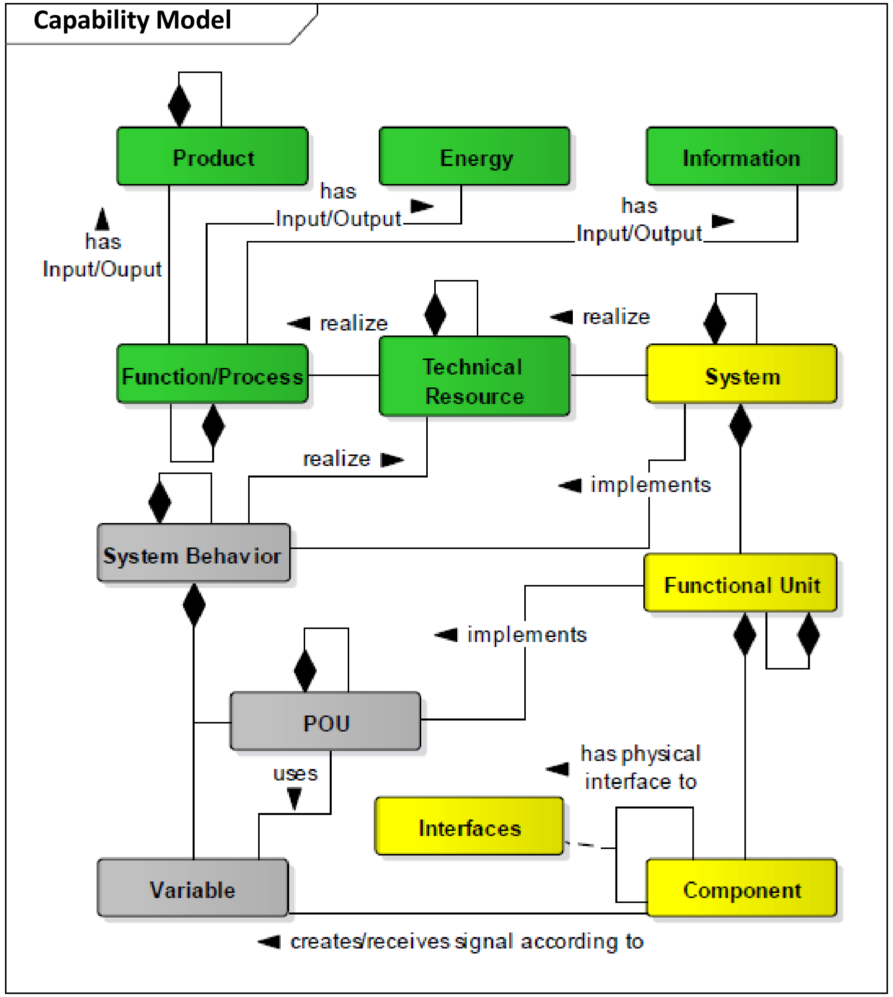
✓ Open context?

✓ Complexity and dynamicity of exchanged data?

✓ Long-term usage?

Terminological Box (T-Box) for capability description

- Function (VDI/VDE 3682-2): Transformation of input to output
- Structure (IEC 62264-1): Components as well as their relations to other components
- Behavior (IEC 61131-3): Procedures, states or variables for information processing



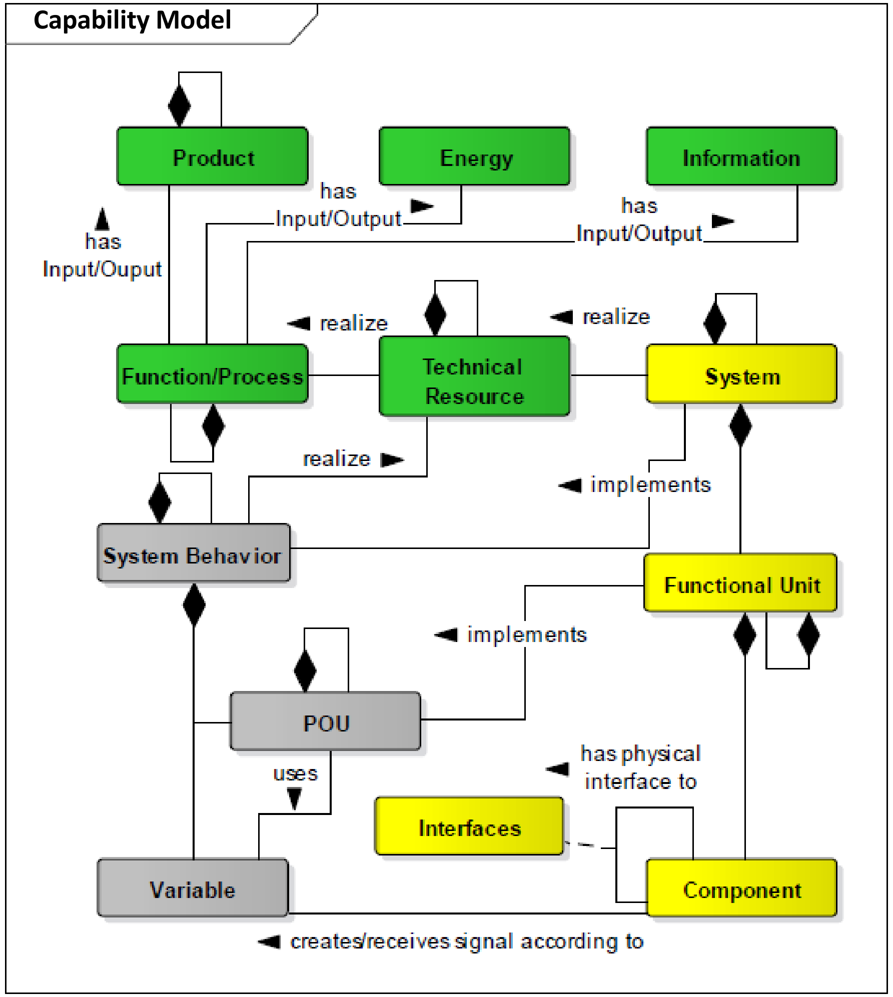
Ontology Usage for Capability Description

Why use ontologies?

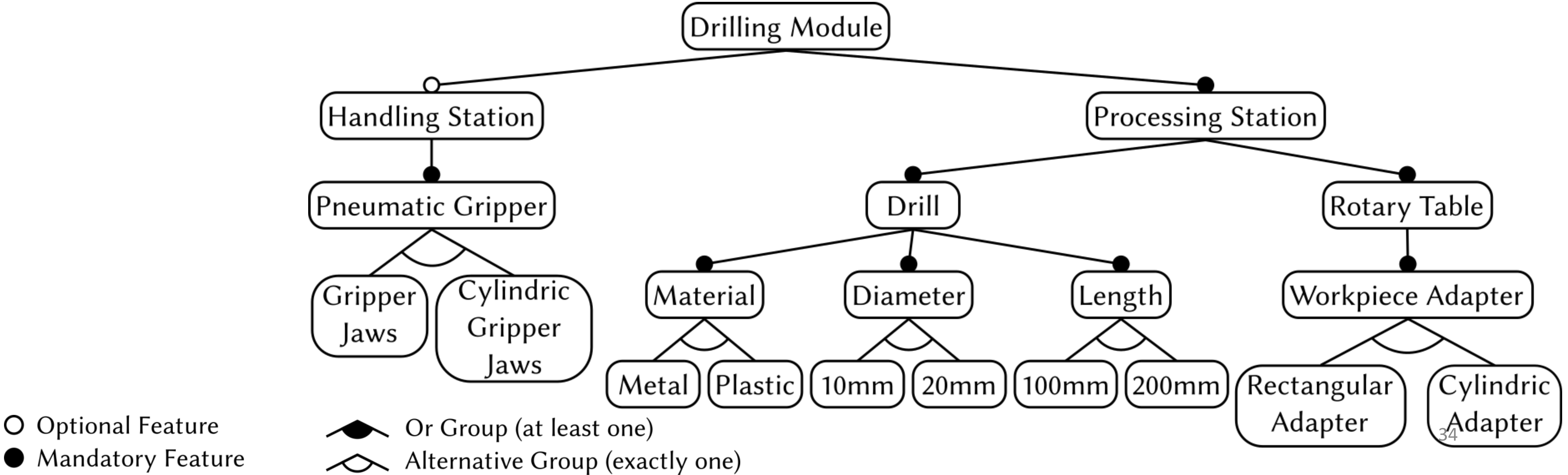
- ✓ High number of interacting systems?
- ✓ Open context?
- ✓ Complexity and dynamicity of exchanged data?
- ✓ Long-term usage?

Terminological Box (T-Box) for capability description

- Function (VDI/VDE 3682-2): Transformation of input to output
- Structure (IEC 62264-1): Components as well as their relations to other components
- Behavior (IEC 61131-3): Procedures, states or variables for information processing
- Characteristics (DIN/EN 61360-1): Simple or complex data types

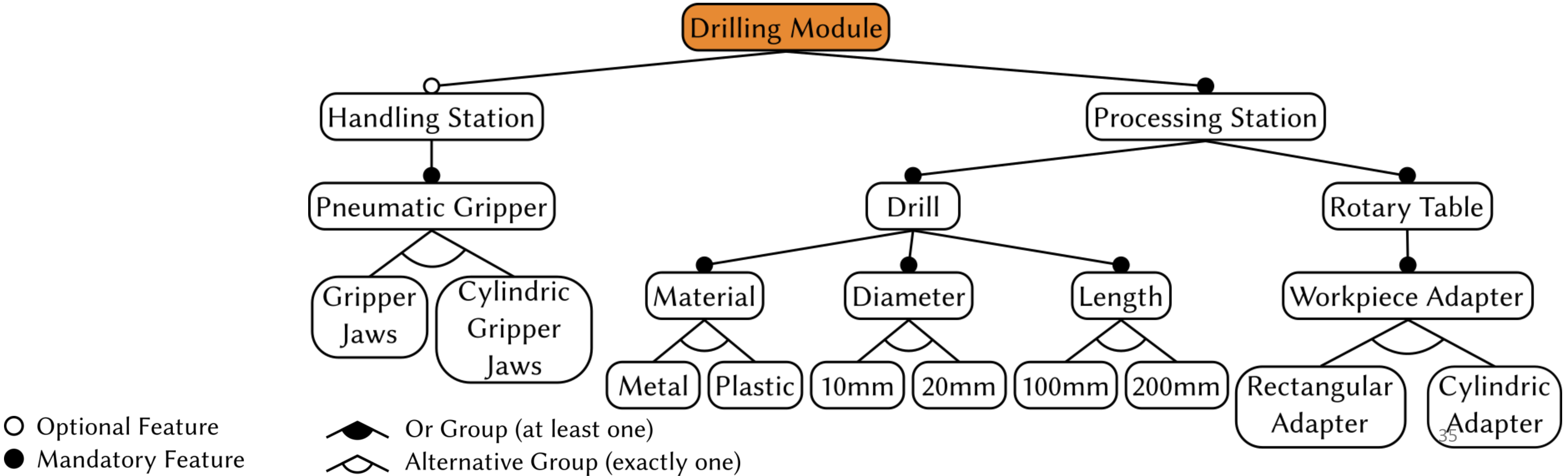


Feature Model



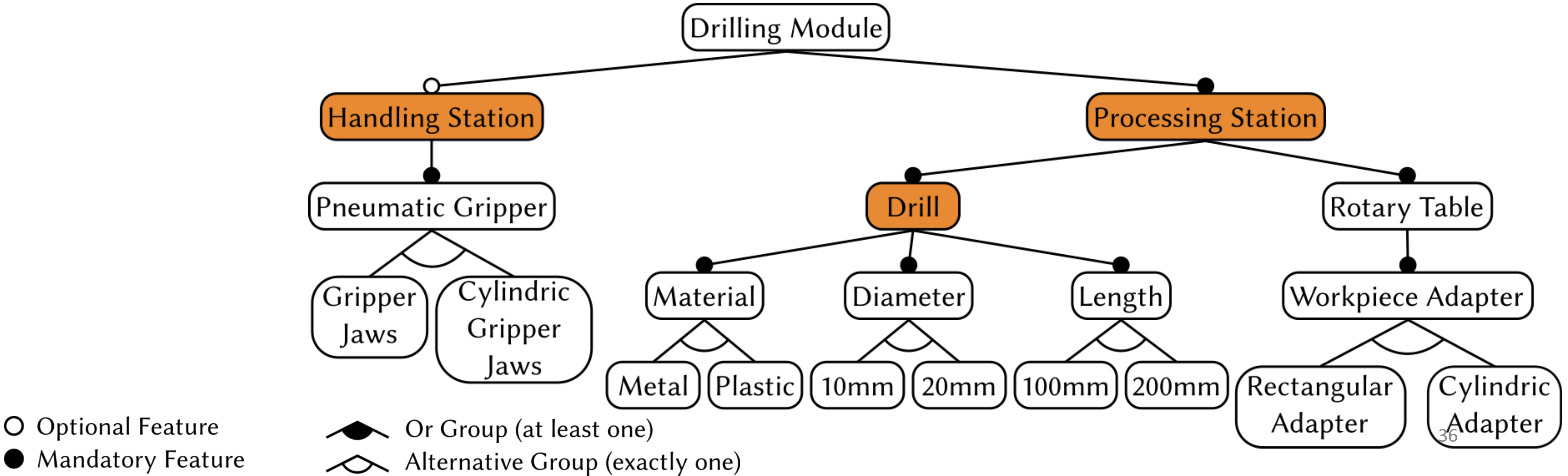
Feature Model

- Root feature represents the manufacturing module



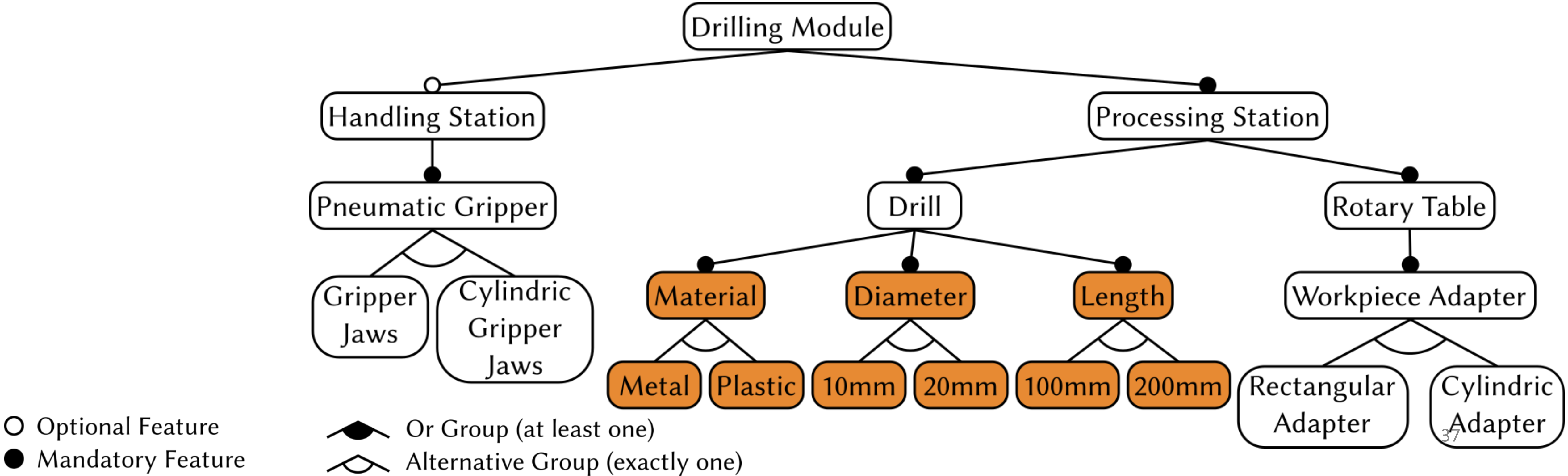
Feature Model

- Root feature represents the manufacturing module
- Adaptable components are modelled below



Feature Model

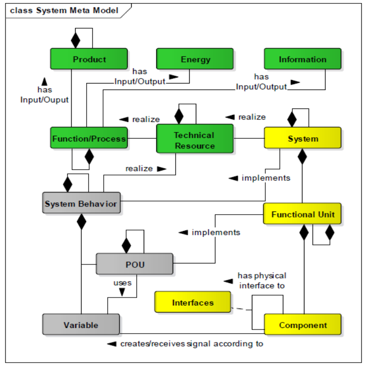
- Root feature represents the manufacturing module
- Adaptable components are modelled below
- Resource parameters are subordinated



Approach Overview

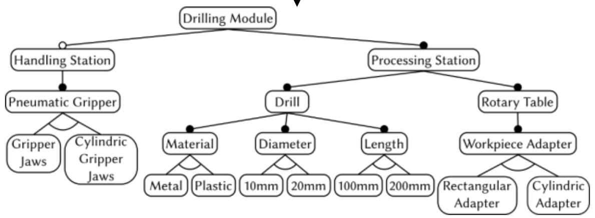
Collaborative Embedded Systems

CrESt



System Ontology

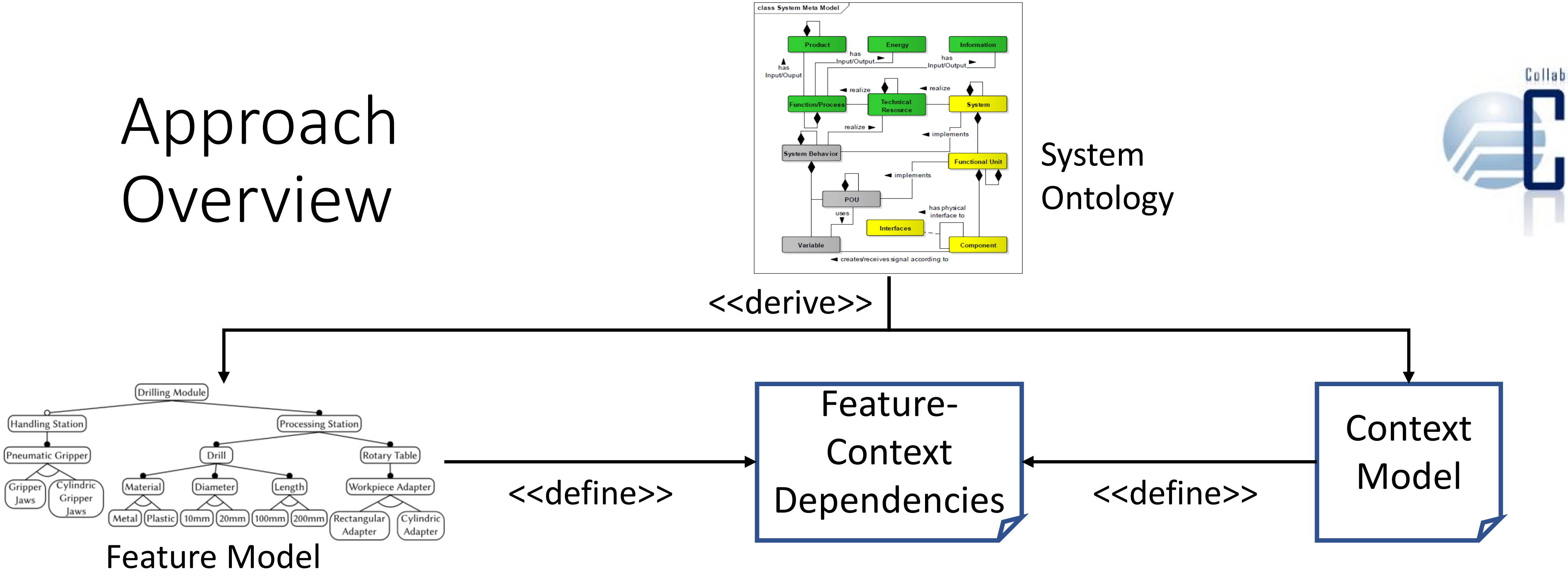
<<derive>>



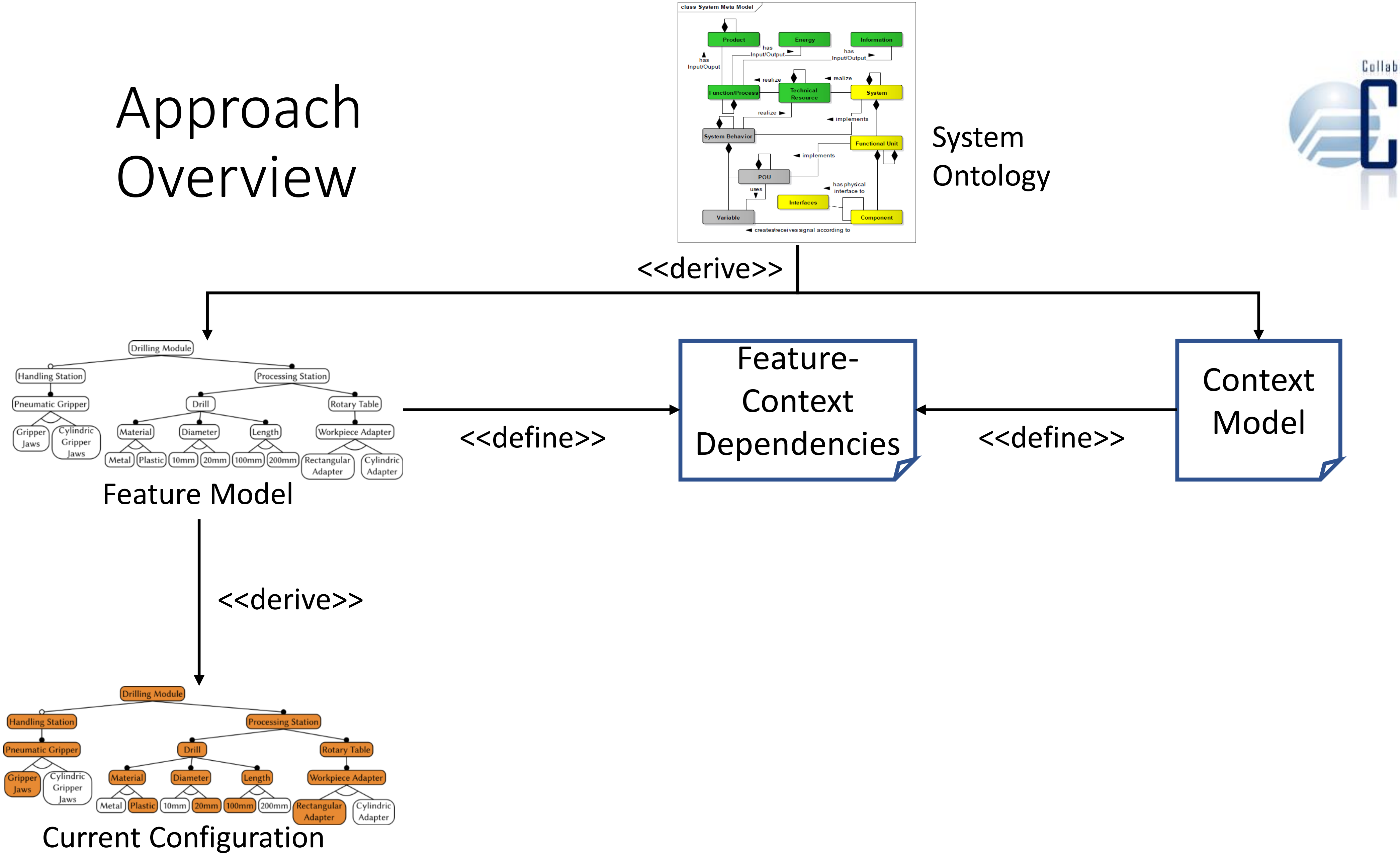
Feature Model

Context Model

Approach Overview



Approach Overview



System Ontology

<<derive>>

Feature Model

<<define>>

Feature-Context Dependencies

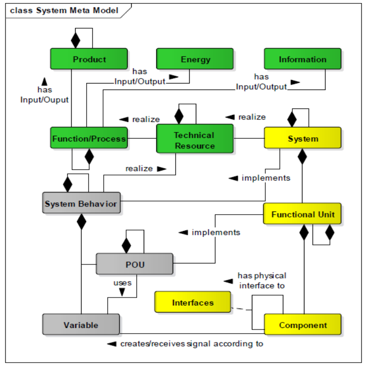
<<define>>

Context Model

<<derive>>

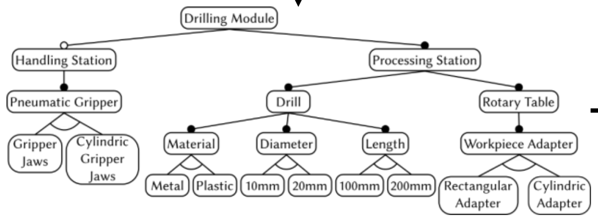
Current Configuration

Approach Overview



System Ontology

<<derive>>



Feature Model

<<define>>

Feature-Context Dependencies

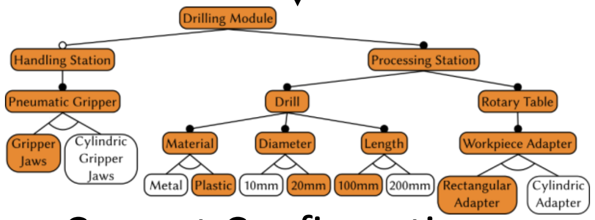
<<define>>

Context Model

New Production Request

New Context Values

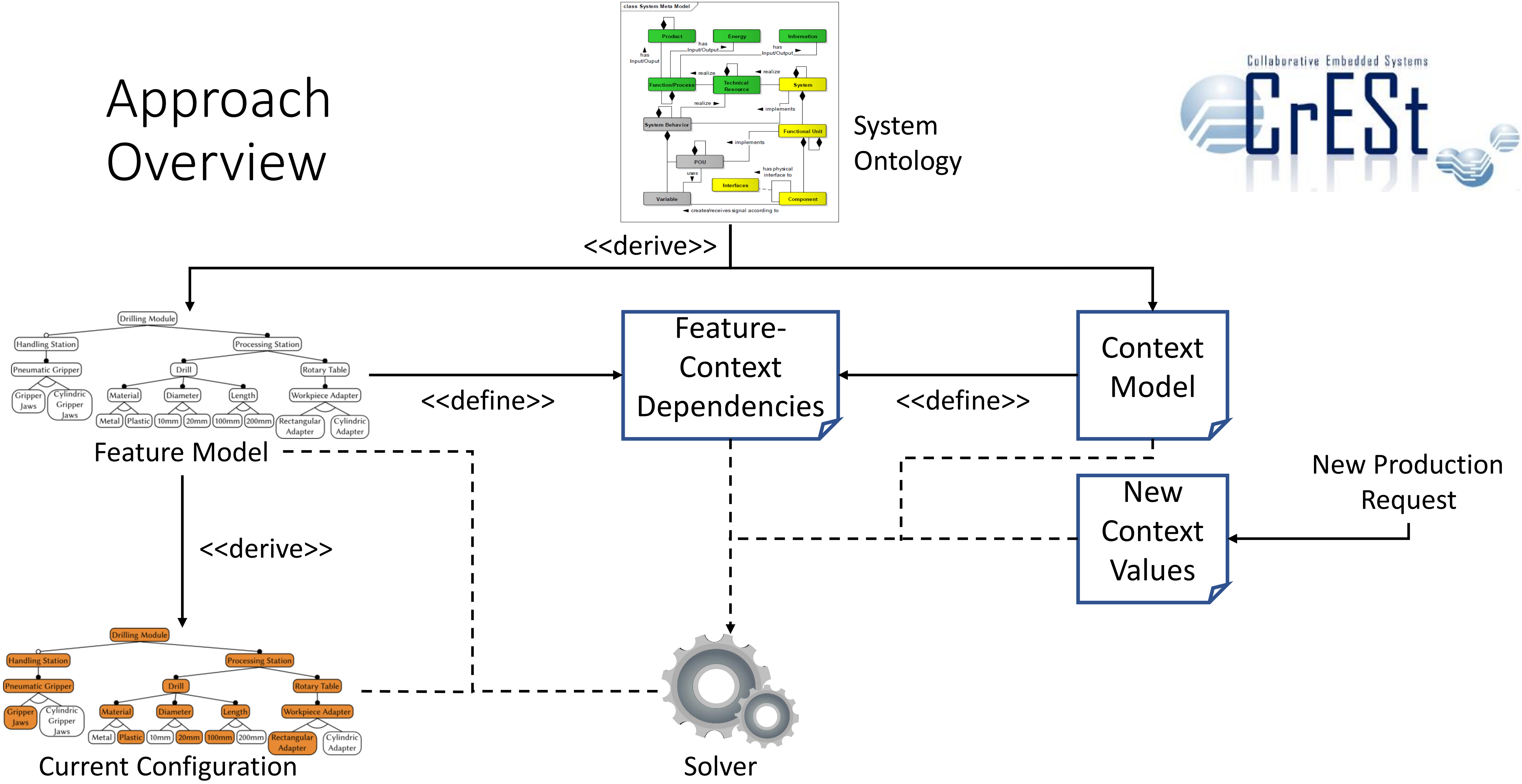
<<derive>>



Current Configuration

16.10.2020

Approach Overview



System Ontology

<<derive>>

Feature Model

<<define>>

Feature-Context Dependencies

<<define>>

Context Model

<<derive>>

New Context Values

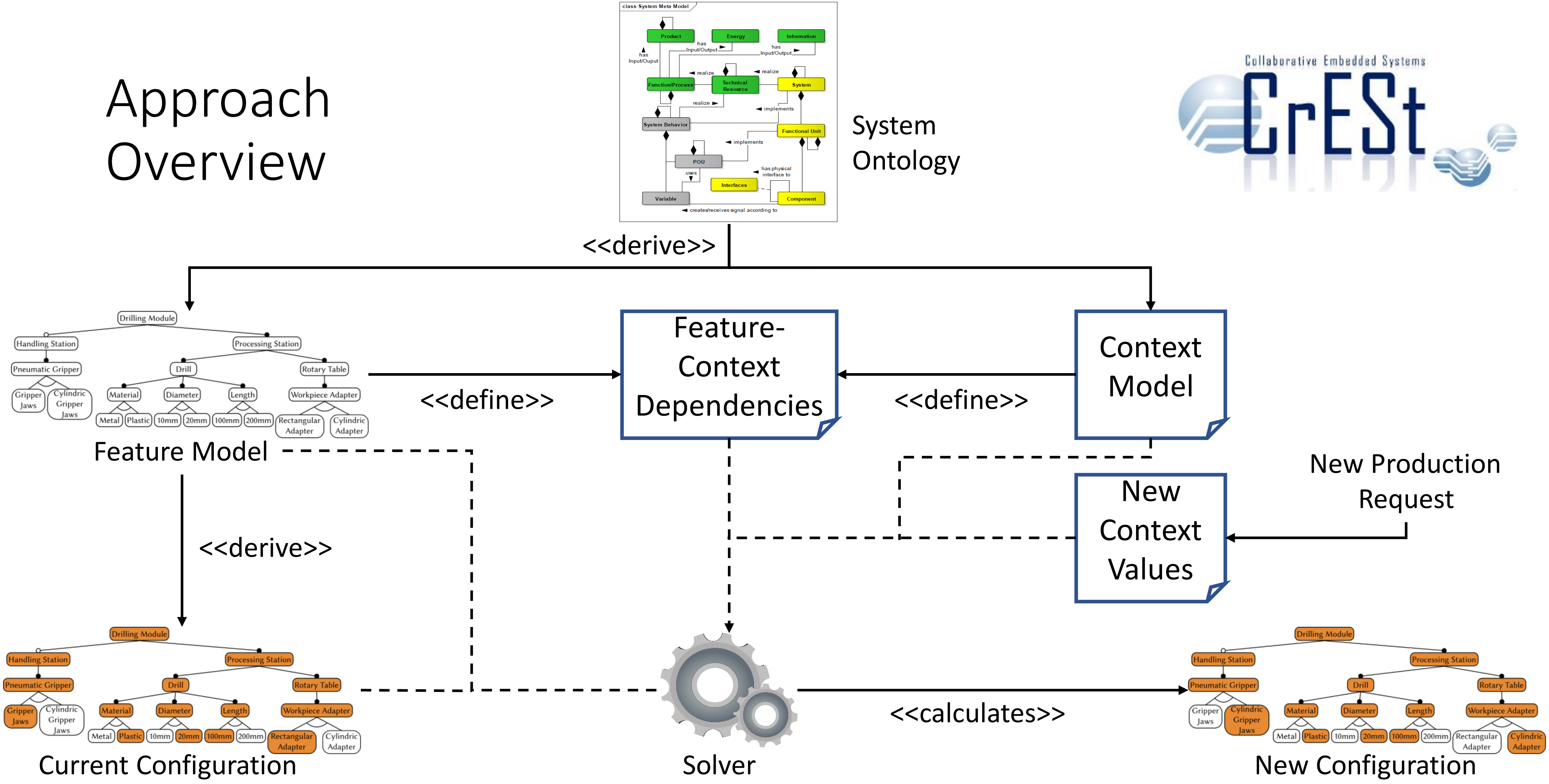
New Production Request

Current Configuration

Solver

Approach Overview

Collaborative Embedded Systems CrESt



System Ontology

<<derive>>

Feature-Context Dependencies

Context Model

<<define>>

<<define>>

Feature Model

<<derive>>

New Context Values

New Production Request

Solver

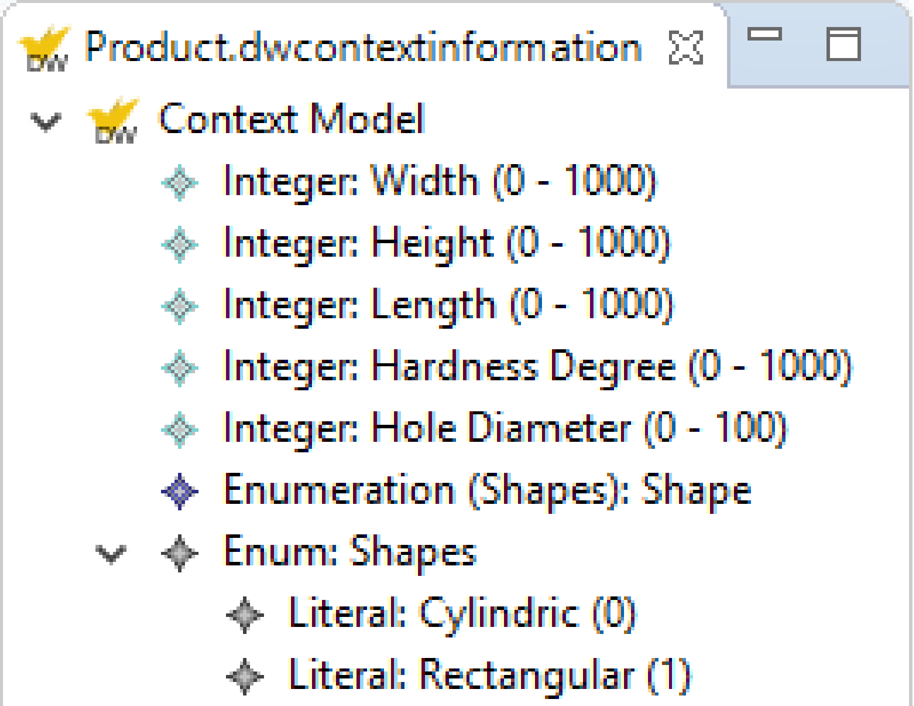
<<calculates>>

Current Configuration

New Configuration

Context-sensitive Variability Model

- Context model contains
  all relevant parameters
  that can trigger a reconfiguration
- Resource parameters are translated into product parameters with parameter range

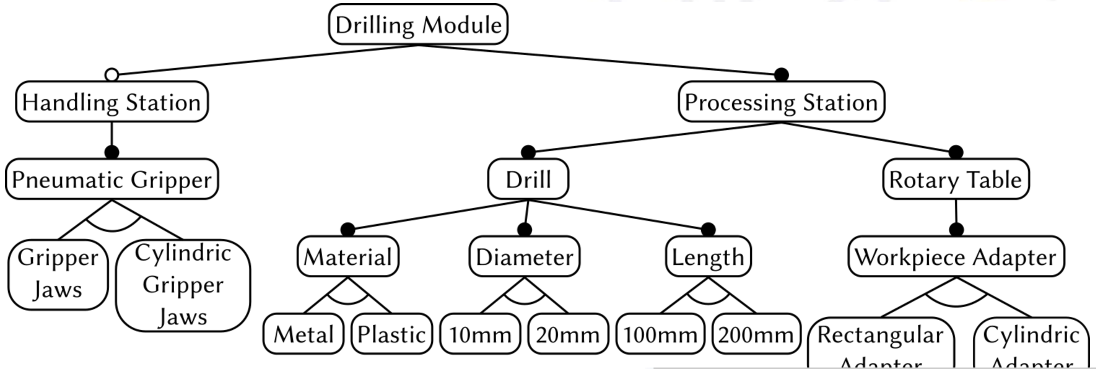


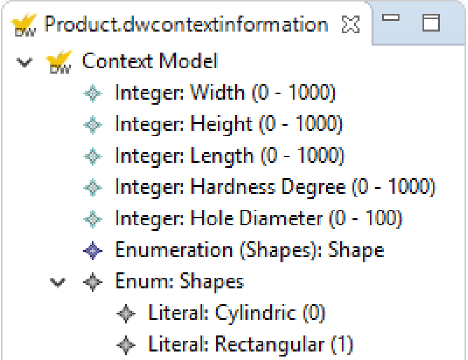
Product.dwcontextinformation

- Context Model
  - Integer: Width (0 - 1000)
  - Integer: Height (0 - 1000)
  - Integer: Length (0 - 1000)
  - Integer: Hardness Degree (0 - 1000)
  - Integer: Hole Diameter (0 - 100)
  - Enumeration (Shapes): Shape
  - Enum: Shapes
    - Literal: Cylindric (0)
    - Literal: Rectangular (1)

Context-sensitive Variability Model

- Dependencies are modeled as **cross-tree constraints**
- Dependencies between:
  - components
  - resource parameters
  - product parameters
  - context information

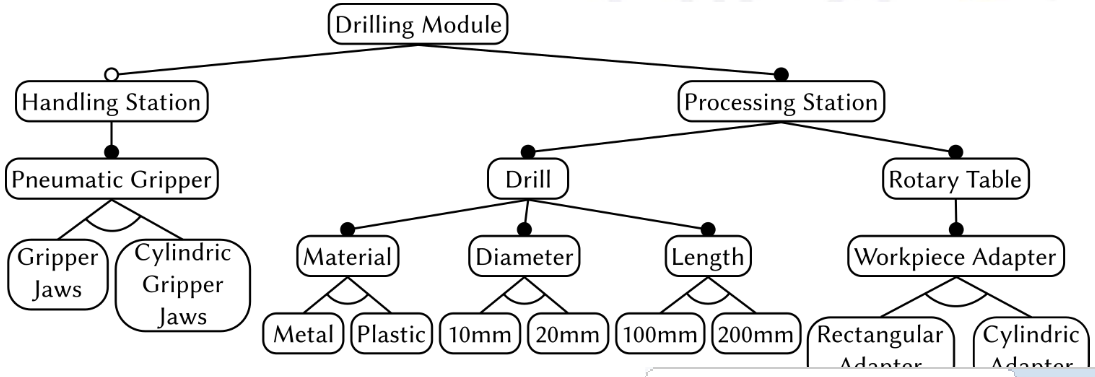


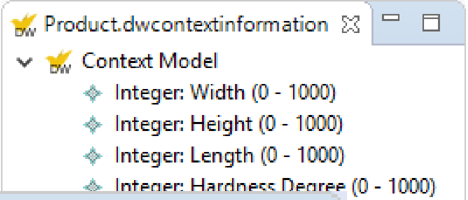

Product.dwcontextinformation

- Context Model
  - Integer: Width (0 - 1000)
  - Integer: Height (0 - 1000)
  - Integer: Length (0 - 1000)
  - Integer: Hardness Degree (0 - 1000)
  - Integer: Hole Diameter (0 - 100)
  - Enumeration (Shapes): Shape
  - Enum: Shapes
    - Literal: Cylindric (0)
    - Literal: Rectangular (1)

Context-sensitive Variability Model

- Dependencies are modeled as **cross-tree constraints**
- Dependencies between:
    - components
    - resource parameters
    - product parameters
    - context information

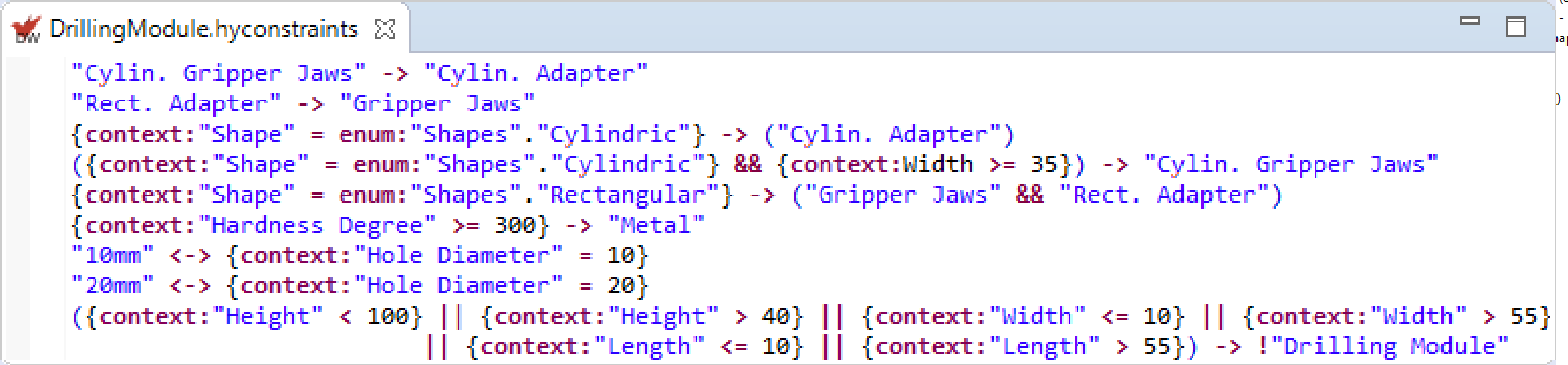



Product.dwcontextinformation

- Context Model
    - Integer: Width (0 - 1000)
    - Integer: Height (0 - 1000)
    - Integer: Length (0 - 1000)
    - Integer: Hardness Degree (0 - 1000)

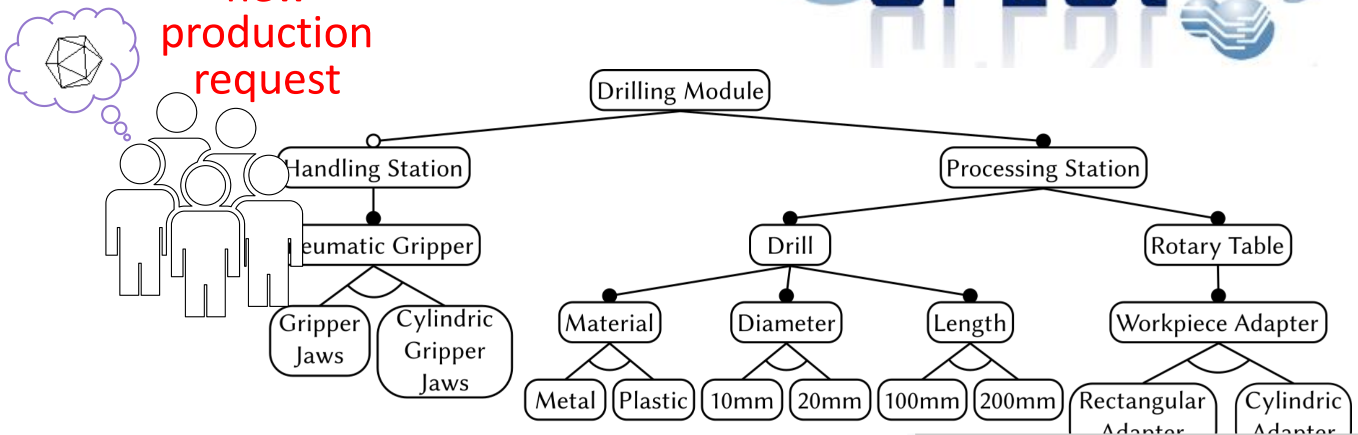
- 100)

ape

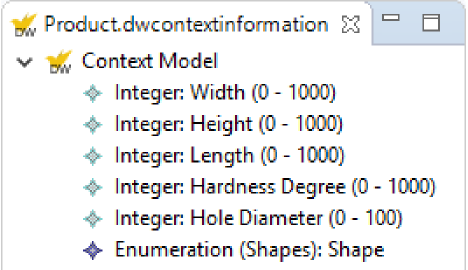

DrillingModule.hyconstraints

```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                        || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

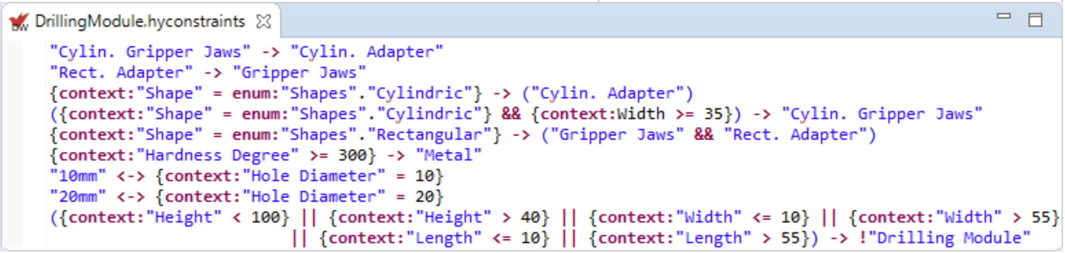
Context-sensitive Variability Model Reconfiguration



new production request



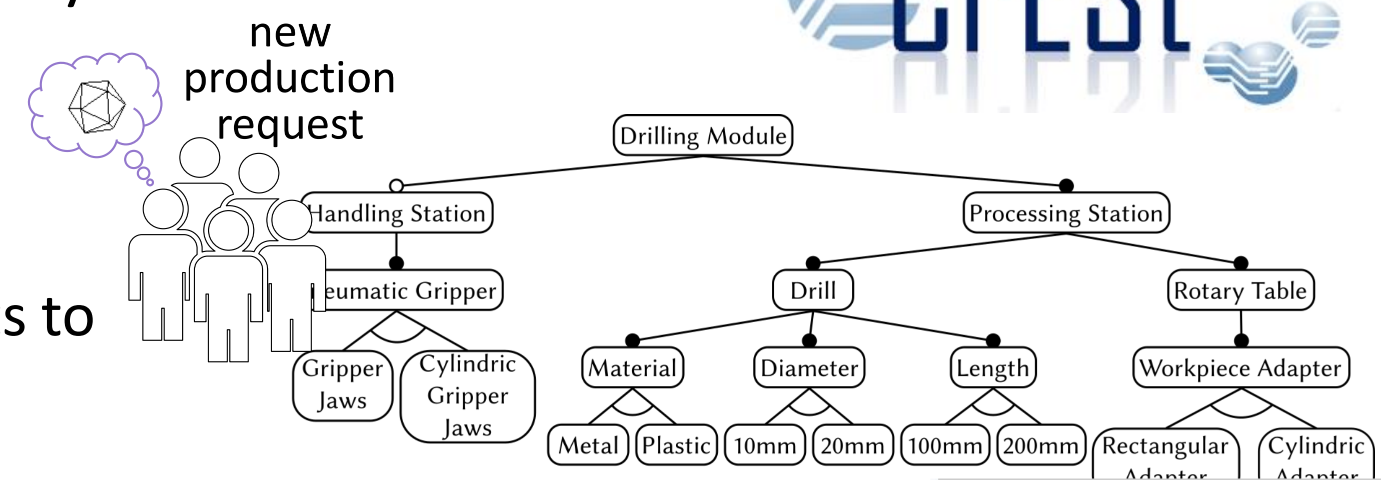
Context Model
- Integer: Width (0 - 1000)
- Integer: Height (0 - 1000)
- Integer: Length (0 - 1000)
- Integer: Hardness Degree (0 - 1000)
- Integer: Hole Diameter (0 - 100)
- Enumeration (Shapes): Shape

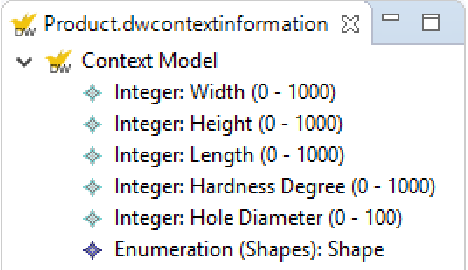


```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                         || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

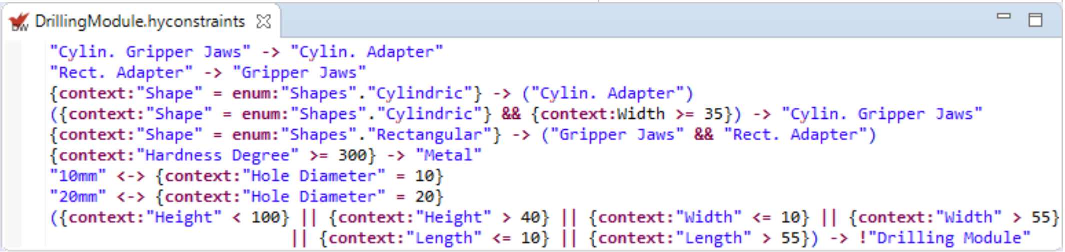
Context-sensitive Variability Model Reconfiguration

1. Translate feature model & constraints to Boolean formulas



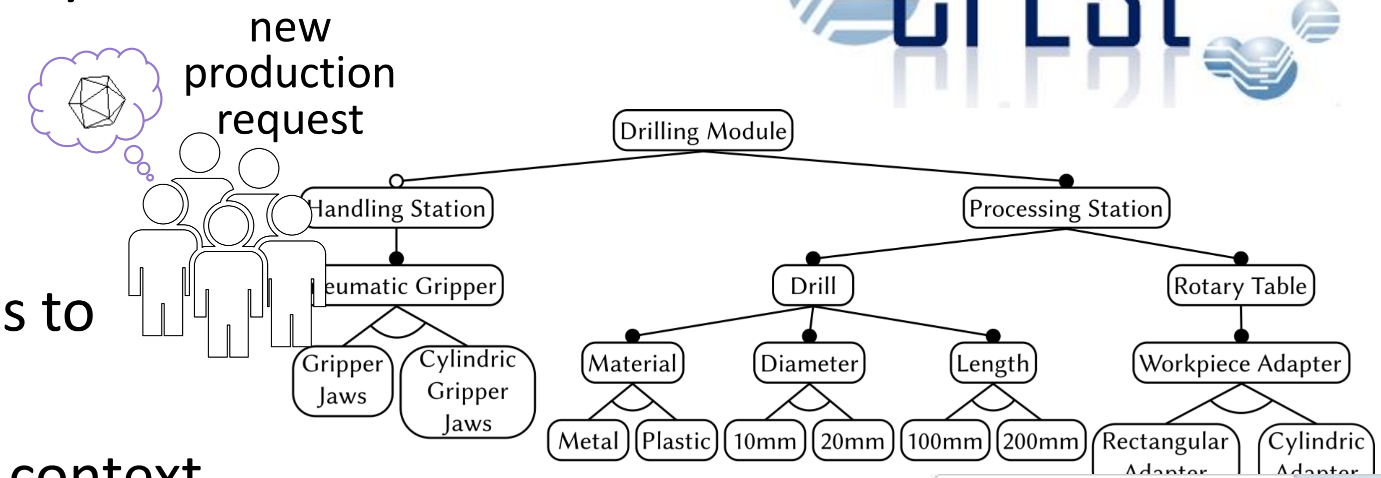


Context Model
- Integer: Width (0 - 1000)
- Integer: Height (0 - 1000)
- Integer: Length (0 - 1000)
- Integer: Hardness Degree (0 - 1000)
- Integer: Hole Diameter (0 - 100)
- Enumeration (Shapes): Shape



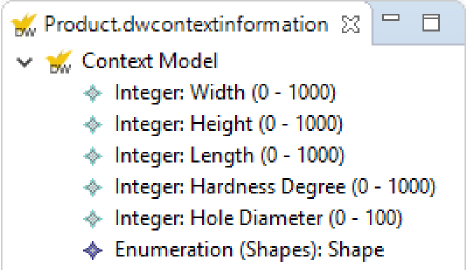
```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                        || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

Context-sensitive Variability Model Reconfiguration

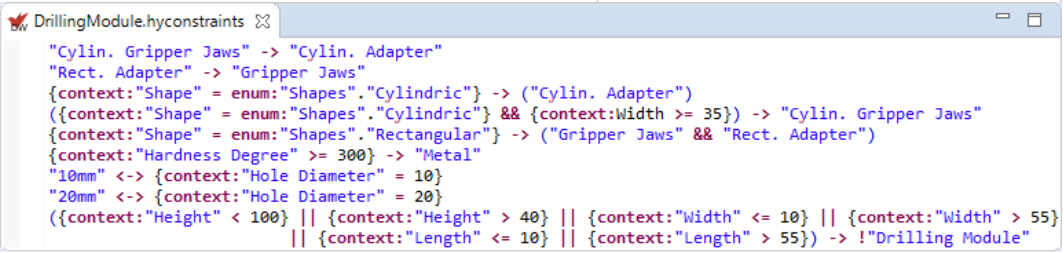


new production request

1. Translate feature model & constraints to Boolean formulas
2. Translate new production request to context values in formula
   - e.g., "Shape = Cylindric"

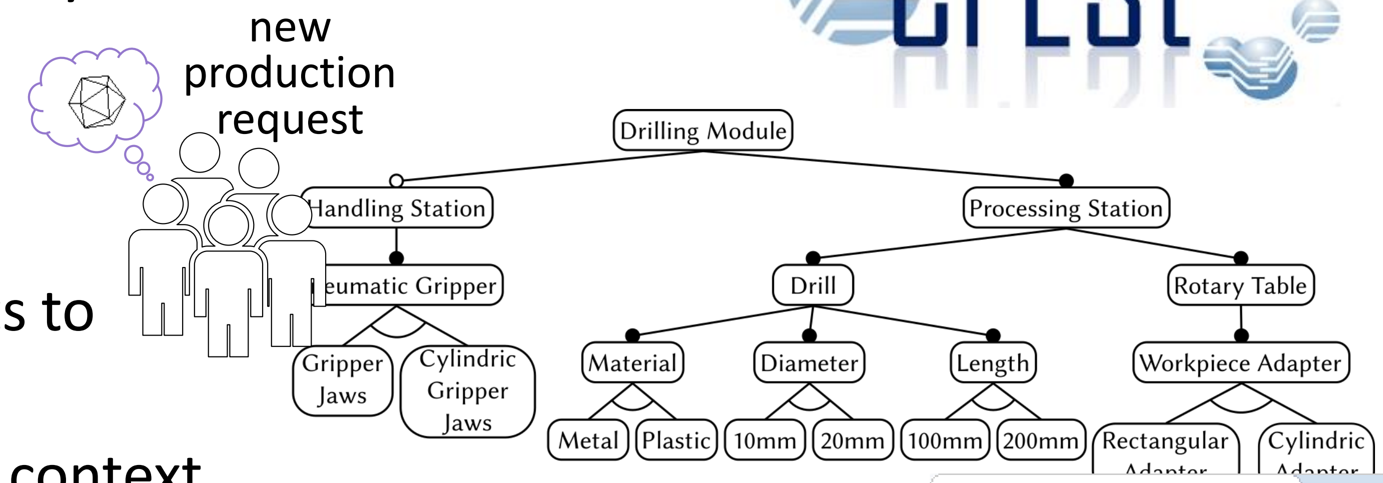


- Context Model
  - Integer: Width (0 - 1000)
  - Integer: Height (0 - 1000)
  - Integer: Length (0 - 1000)
  - Integer: Hardness Degree (0 - 1000)
  - Integer: Hole Diameter (0 - 100)
  - Enumeration (Shapes): Shape

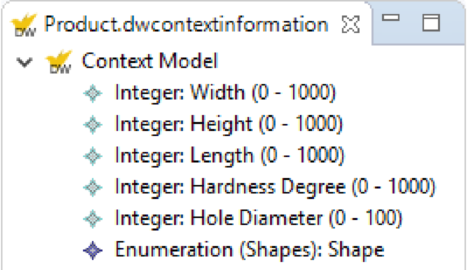


```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                         || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

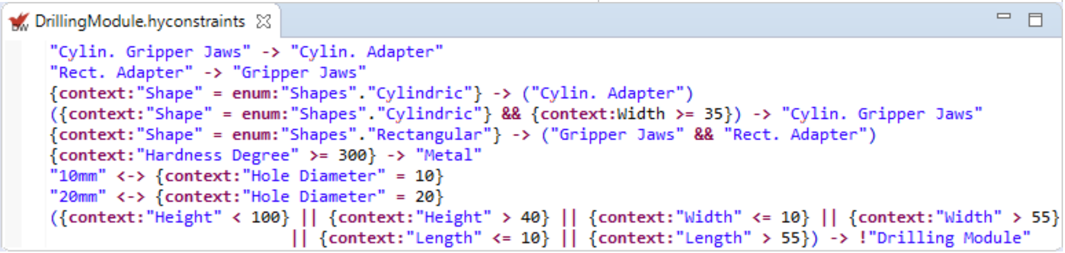
Context-sensitive Variability Model Reconfiguration



1. Translate feature model & constraints to Boolean formulas
2. Translate new production request to context values in formula
   - e.g., “Shape = Cylindric”
3. Evaluate whether old feature configuration satisfies all formulas.
   a. Yes → done
   b. No → Find a new solution for entire formula
      - Optimization problem: new solution similar to old one

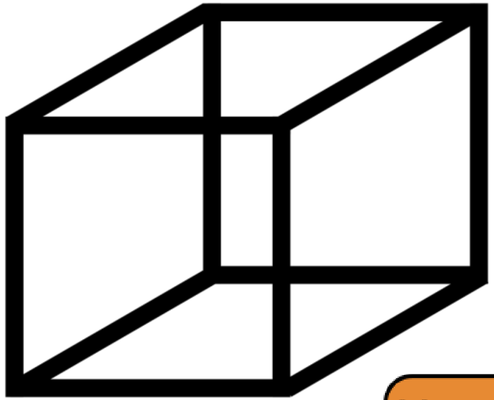

Product.dwcontextinformation

- Context Model
  - Integer: Width (0 - 1000)
  - Integer: Height (0 - 1000)
  - Integer: Length (0 - 1000)
  - Integer: Hardness Degree (0 - 1000)
  - Integer: Hole Diameter (0 - 100)
  - Enumeration (Shapes): Shape

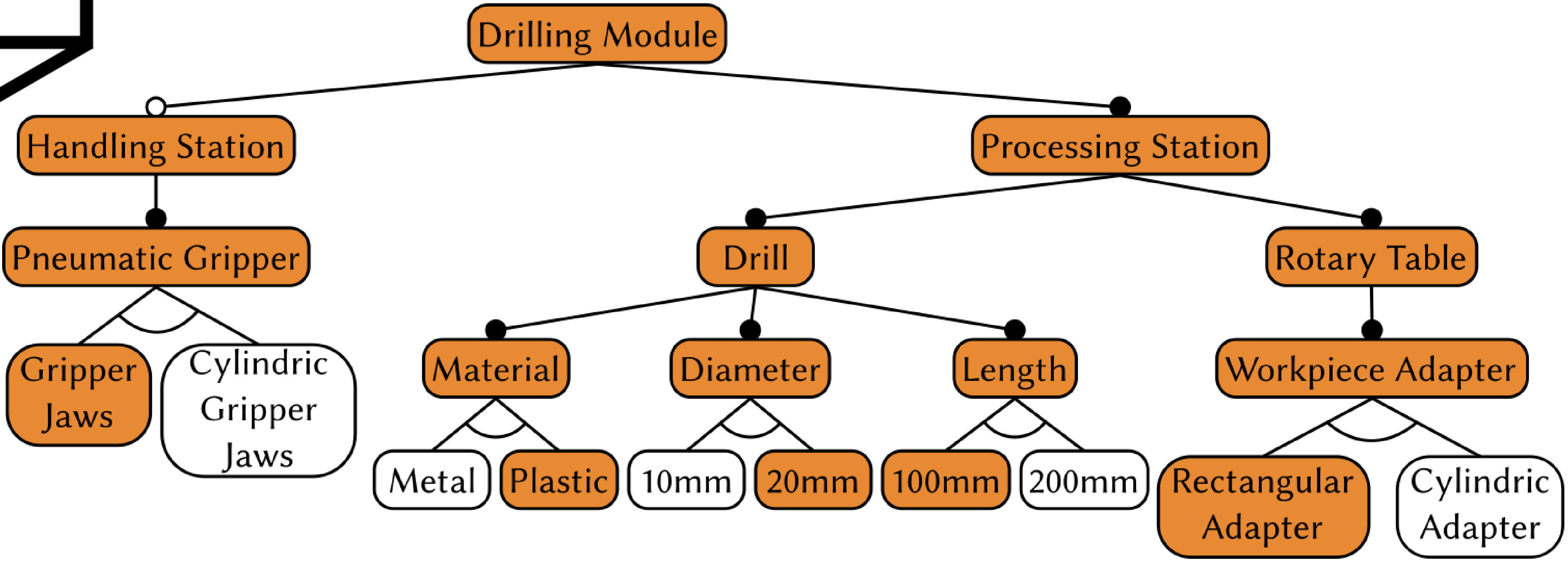

DrillingModule.hyconstraints

```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                         || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

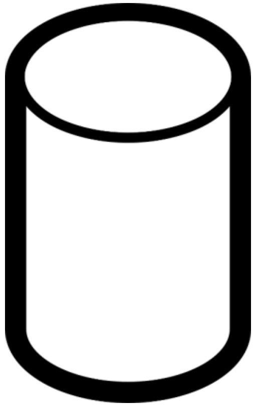
Context-sensitive Variability Model Reconfiguration – Example



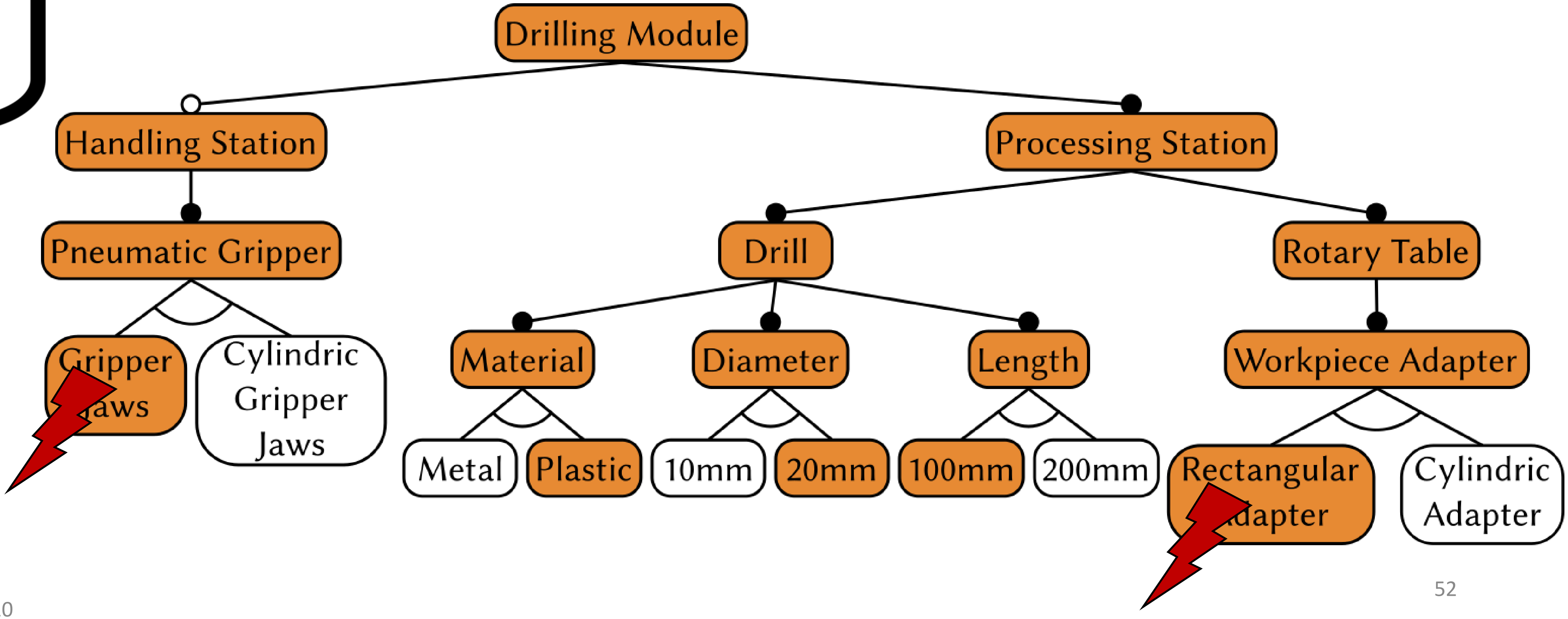
Context:
Shape = Rectangular



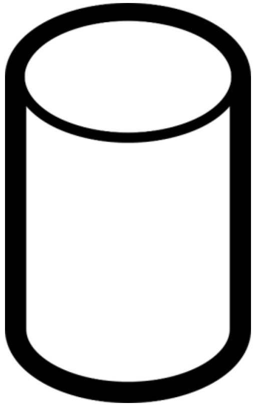
Context-sensitive Varaiability Model Reconfiguration – Example



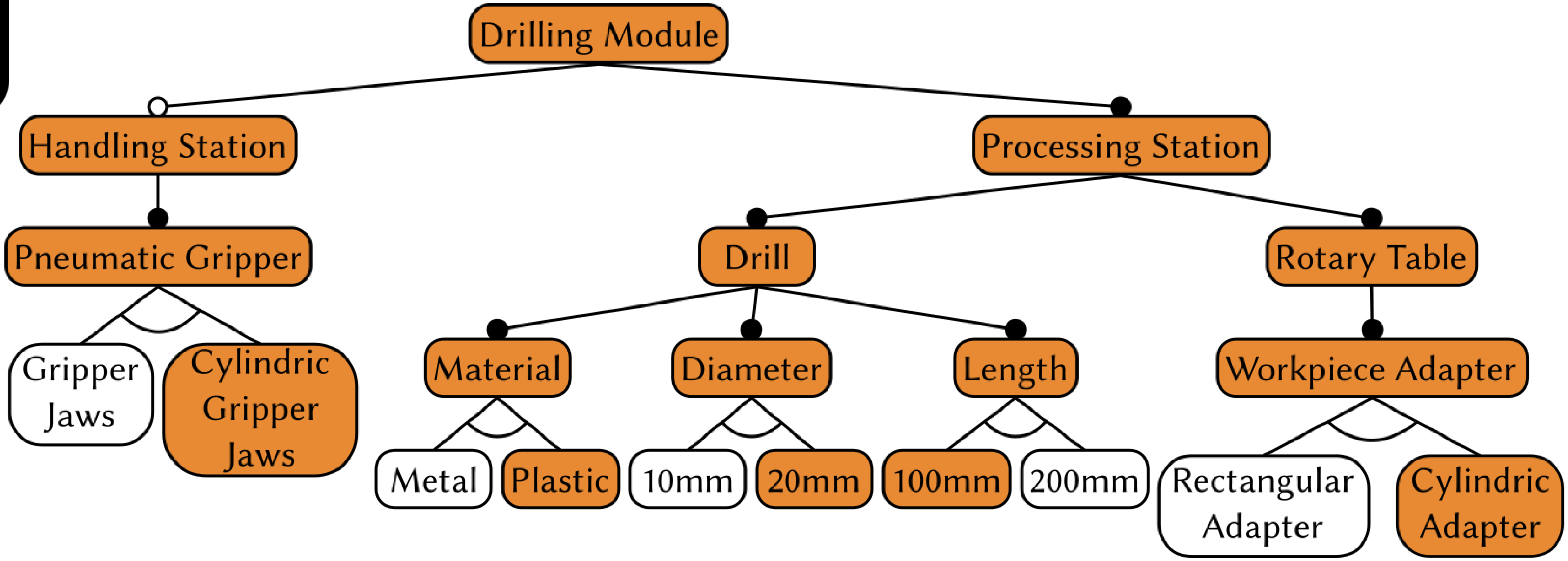
Context:
Shape = Cylindric



Context-sensitive Varaiability Model Reconfiguration – Example



Context:
Shape = Cylindric



Drilling Module

Handling Station

Processing Station

Pneumatic Gripper

Drill

Rotary Table

Gripper Jaws

Cylindric Gripper Jaws

Material

Diameter

Length

Workpiece Adapter

Metal

Plastic

10mm

20mm

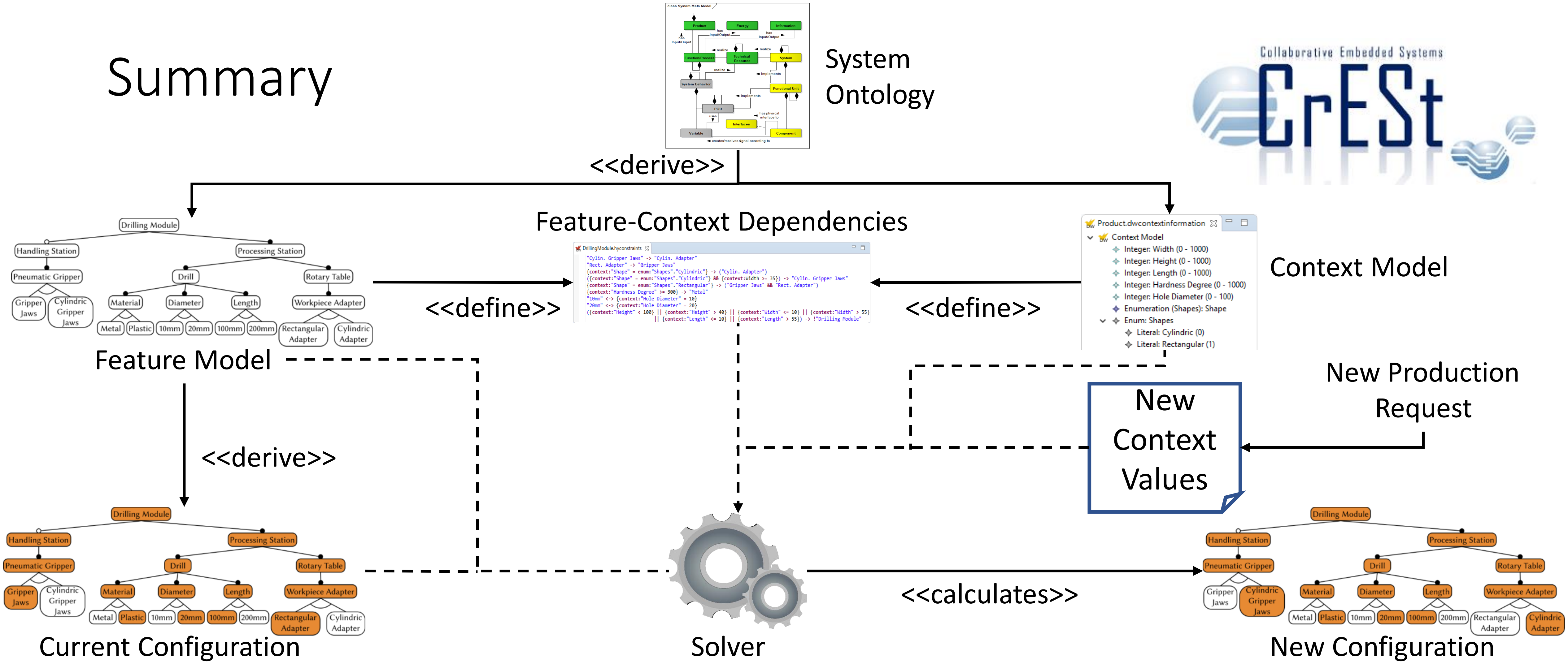
100mm

200mm

Rectangular Adapter

Cylindric Adapter

Summary



DrillingModule.hyconstraints

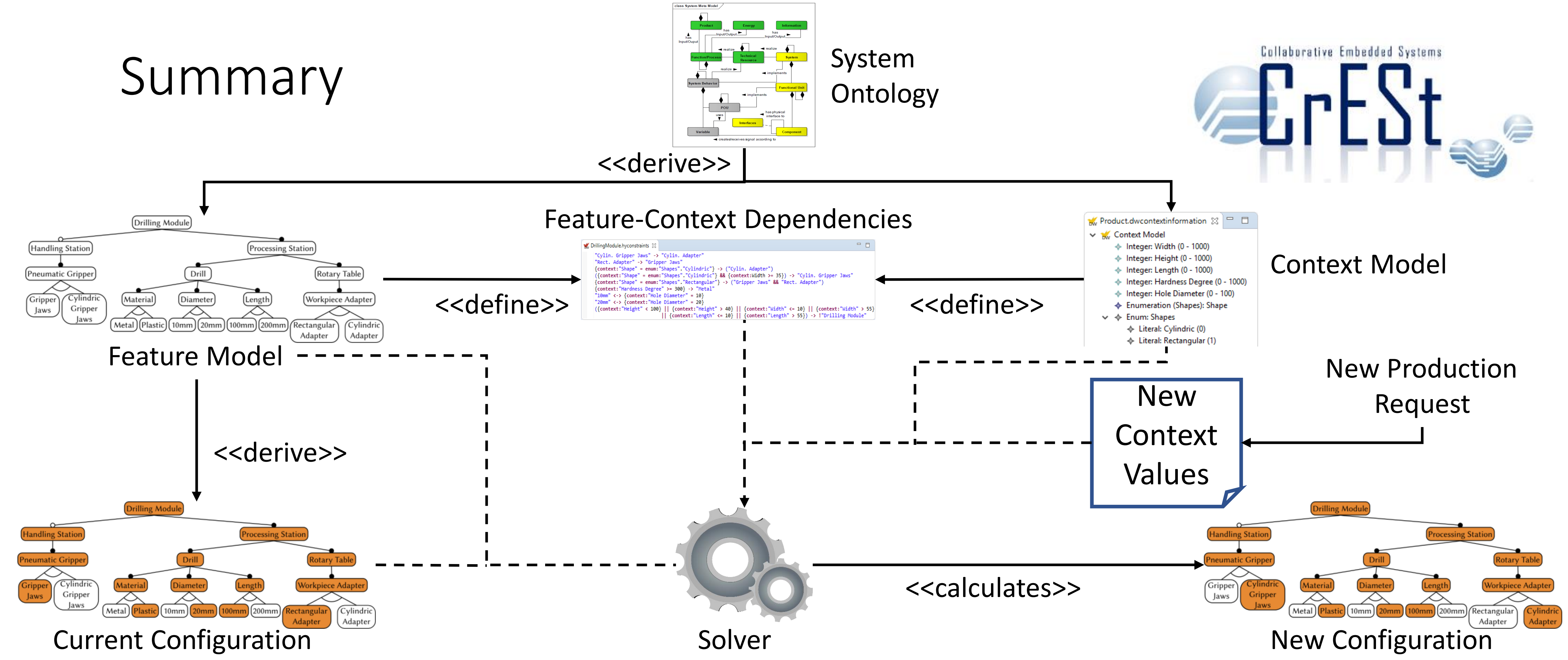
```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                              || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

Product.dwcontextinformation

- Context Model
  - Integer: Width (0 - 1000)
  - Integer: Height (0 - 1000)
  - Integer: Length (0 - 1000)
  - Integer: Hardness Degree (0 - 1000)
  - Integer: Hole Diameter (0 - 100)
  - Enumeration (Shapes): Shape
  - Enum: Shapes
    - Literal: Cylindric (0)
    - Literal: Rectangular (1)

Collaborative Embedded Systems
CrESt

Summary



System Ontology

<<derive>>

Feature-Context Dependencies

DrillingModule.hyconstraints

```
"Cylin. Gripper Jaws" -> "Cylin. Adapter"
"Rect. Adapter" -> "Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Cylindric"} -> ("Cylin. Adapter")
({context:"Shape" = enum:"Shapes"."Cylindric"} && {context:Width >= 35}) -> "Cylin. Gripper Jaws"
{context:"Shape" = enum:"Shapes"."Rectangular"} -> ("Gripper Jaws" && "Rect. Adapter")
{context:"Hardness Degree" >= 300} -> "Metal"
"10mm" <-> {context:"Hole Diameter" = 10}
"20mm" <-> {context:"Hole Diameter" = 20}
({context:"Height" < 100} || {context:"Height" > 40} || {context:"Width" <= 10} || {context:"Width" > 55}
                          || {context:"Length" <= 10} || {context:"Length" > 55}) -> !"Drilling Module"
```

Product.dwcontextinformation

- Context Model
  - Integer: Width (0 - 1000)
  - Integer: Height (0 - 1000)
  - Integer: Length (0 - 1000)
  - Integer: Hardness Degree (0 - 1000)
  - Integer: Hole Diameter (0 - 100)
  - Enumeration (Shapes): Shape
  - Enum: Shapes
    - Literal: Cylindric (0)
    - Literal: Rectangular (1)

Context Model

<<define>>

<<define>>

Feature Model

<<derive>>

New Production Request

New Context Values

Current Configuration

Solver

<<calculates>>

New Configuration

- Demo Video on the Marketplace!
- Tooling open source: https://gitlab.com/DarwinSPL/DarwinSPL
- Publications: Context-sensitive reconfiguration of collaborative manufacturing systems, B. Caesar et al.
  DarwinSPL: an integrated tool suite for modeling evolving context-aware software product lines, M. Nieke et al.

DLR Projektträger

Collaborative Embedded Systems

CrESt

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

CAROLO-WILHELMINA BRAUNSCHWEIG

Technische
Universität
Braunschweig

Vielen Dank!

HELMUT SCHMIDT
UNIVERSITÄT
Universität der Bundeswehr Hamburg

Fragen ?

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Backup

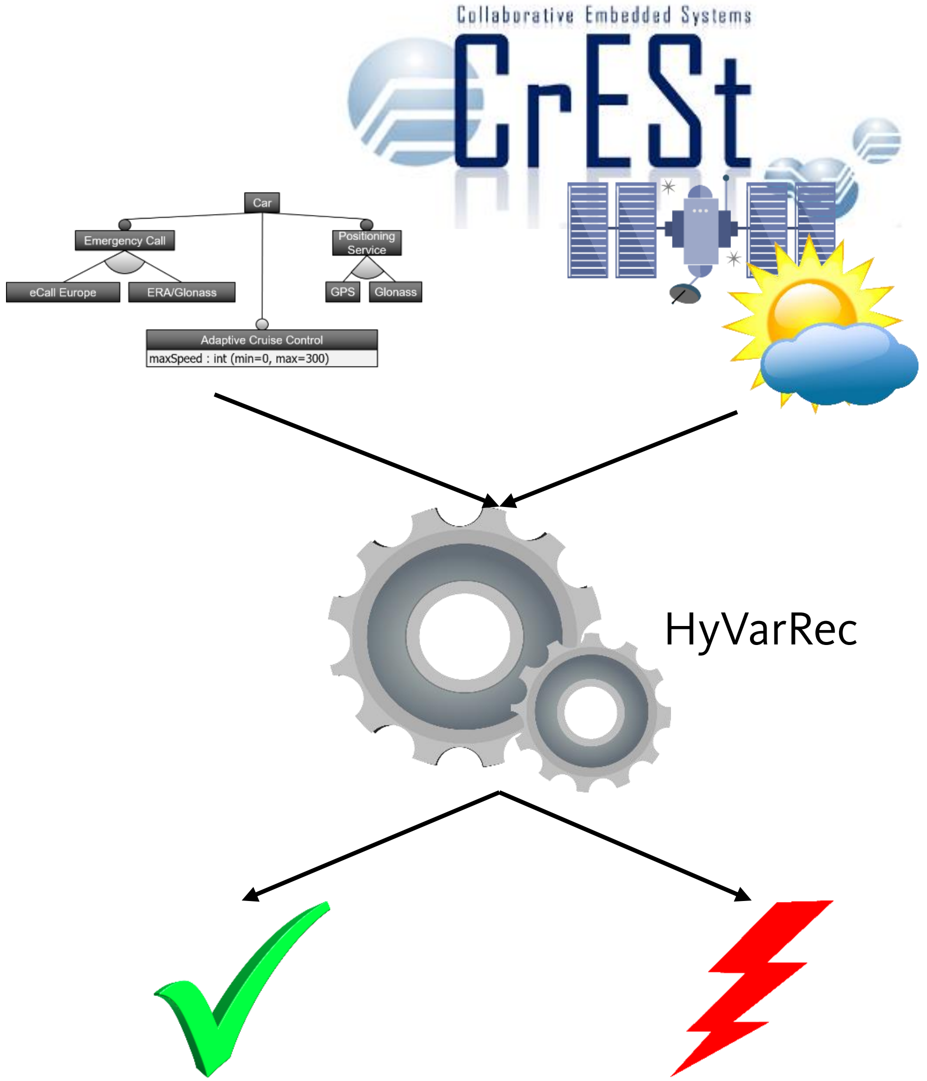
- Create a satisfiability request

- Variables:
- Context
- Features
- Feature attributes
- Variable domains

- Constraints:
- Feature model structure
- Cross-tree constraints
- Validity formulas

$$\varphi_F = \forall c_1, \dots, c_n \cdot \exists f_1, \dots, f_m \cdot \exists a_1, \dots, a_k \cdot \varphi_{dom} \wedge \varphi_{F,struct} \wedge \varphi_{F,ctc}$$

- Negate formula to find anomalies
- A solution of $\neg\varphi_F$ represents context values for which no solution exists


Collaborative Embedded Systems
CrESt
Car
Emergency Call
Positioning Service
eCall Europe
ERA/Glonass
GPS
Glonass
Adaptive Cruise Control
maxSpeed : int (min=0, max=300)
HyVarRec