



# <AutomationML/>

The Glue for Seamless  
Automation Engineering

# **Collaboration of Tools for Production System Planning and PLC Programming by Using AutomationML**

**AutomationML User Conference 2018**

**Mario Thron, ifak e.V.**

**Klaus Hanisch, tarakos GmbH**

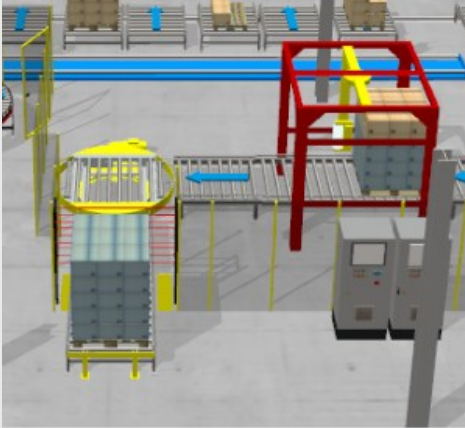
**Matthias Mueller, Mitsubishi Electric Europe B.V.**



## **Agenda**

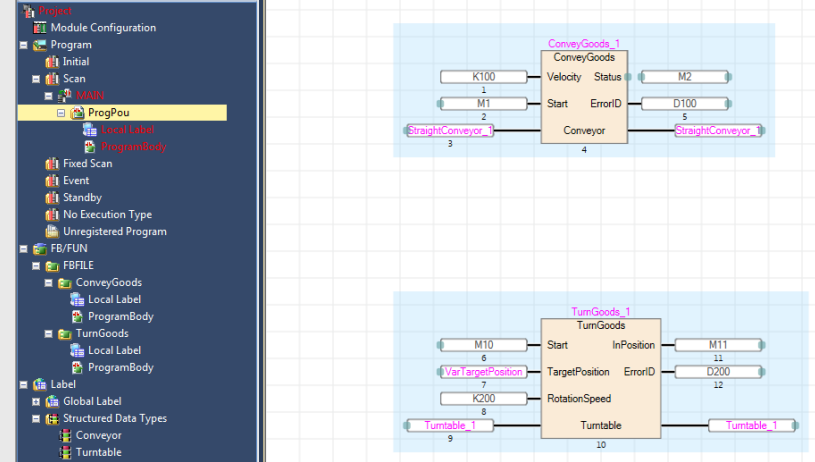
**Motivation – PLC programming**  
**AutomationML – PPR Connector**  
**Mapping AML to PLC program**  
**Data Exchange**  
**Requirements/Milestones**

## Production System Planning Tool



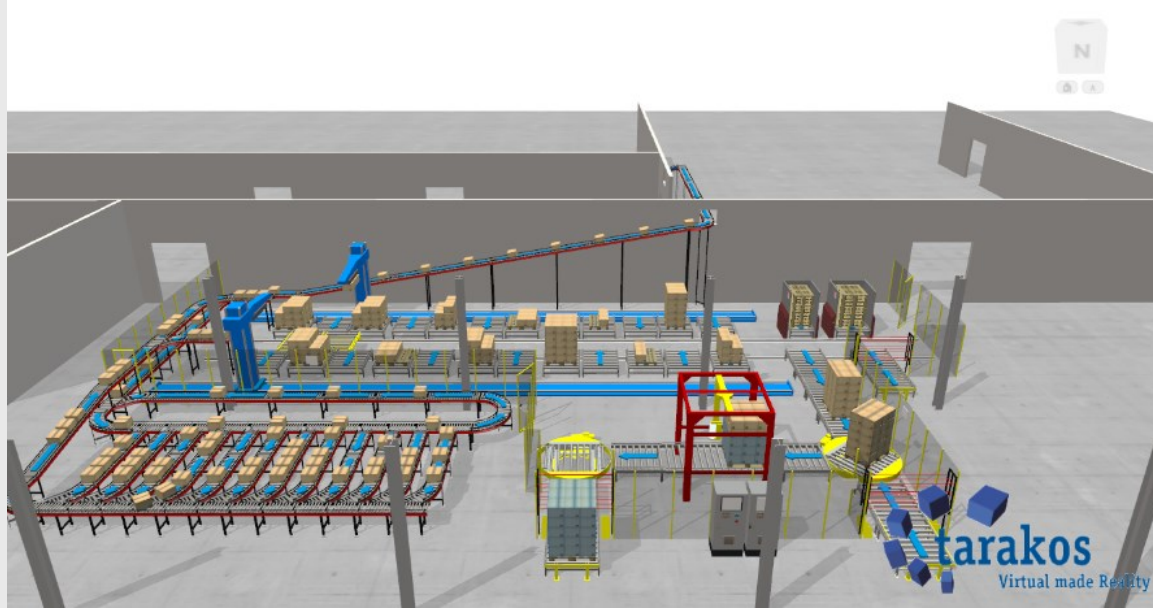
- Select components from libraries
- Arrange and combine them
- Define the tasks of those components
- Confirm/Validate the production system design

## PLC Programming



- Import existing libraries of SDTs and FBs
- Instantiate SDTs and FBs from those libraries
- Assign values to the SDT members and FB inputs based on specification of the production planning system
- Program logic between those FBS
- Create SDTs and FBs if no library exists

# Improving the PLC programming



The PLC programming for a single conveyor might not be so complicated and time consuming, but this changes when looking at huge production facilities.

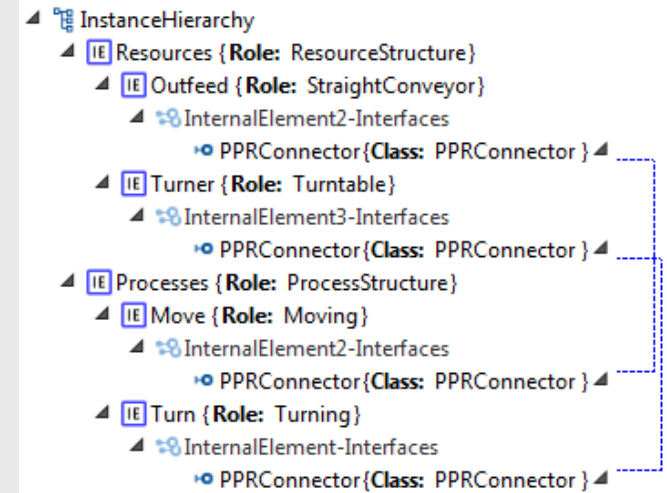
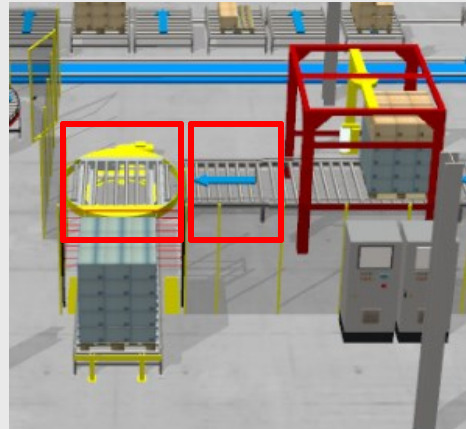
Such facility consists of large numbers of conveyors, turntables, palletizers, etc.

In the PLC program the same SDTs and FBs are used to control the same types of conveyors. Only the values of SDT members and FB inputs are different for each instance.

If a digital description of the production system would exist and if a mapping to SDTs and FBs, between the component types and the tasks they are executing, would exist, the PLC program could be partly generated automatically.

- AutomationML introduces the Product-Process-Resource concept in Whitepaper Part 1
- In order to structure complex plant engineering data, trisection of data into resources, processes and products has delivered proven performance in practice.
- Resources, Processes and Products are connected by the PPR-Connector
- This concept can be applied to the example below

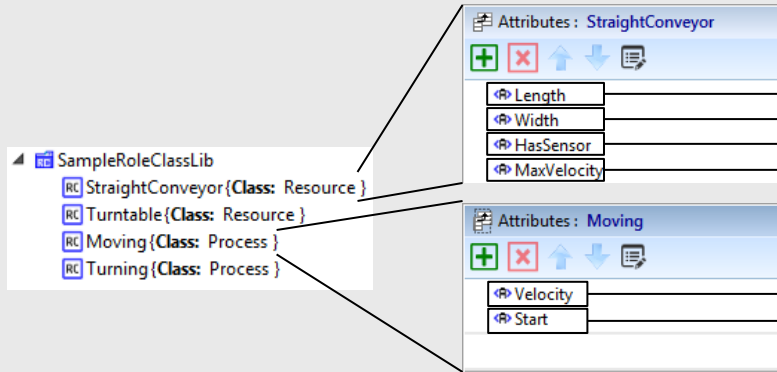
- Resources:**
  - Straight Conveyor
  - Turntable
- Processes**
  - Moving
  - Turning



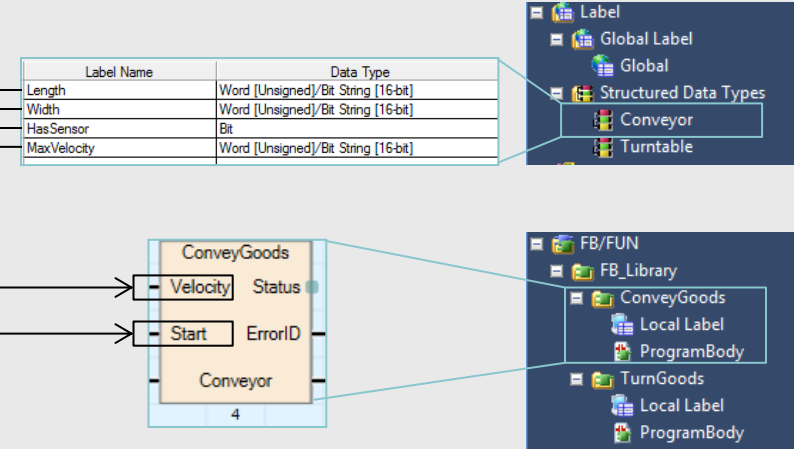


# Details of the Mapping: Types

## AutomationML Role Class Library



## PLC program type definitions

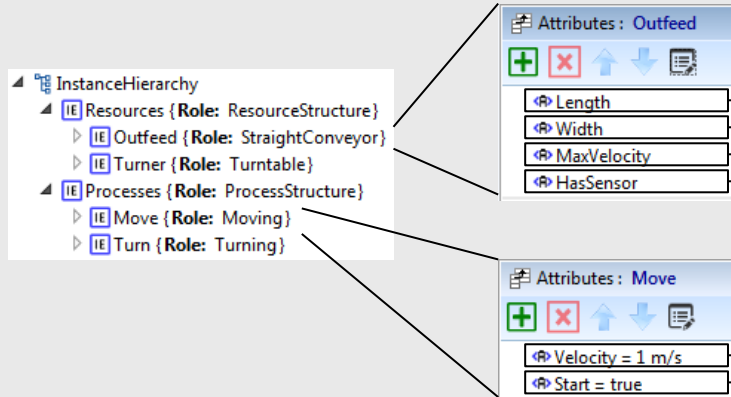


### ① Mapping of Role Classes to Definitions

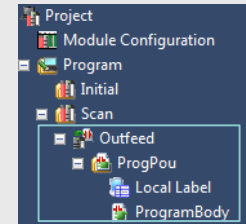
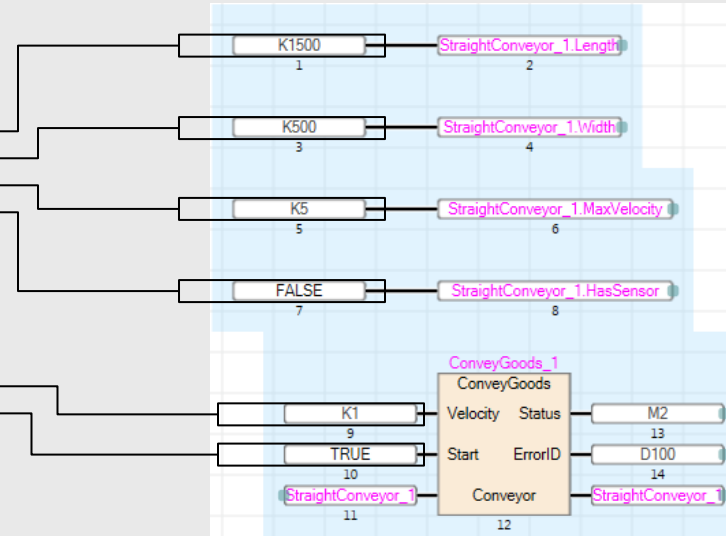
- Based on the Role Classes a mapping is created
- A Role Class derived from Resource is mapped to a definition of an SDT and an FB library
  - Attributes of the Role Class are mapped to Members of the SDT
- A Role Class derived from Process is mapped to a definition of an FB of the library
  - Attributes of the Role Class are mapped to In- and Outputs of the FB

# Details of the Mapping: Instances

## AutomationML Instance Hierarchy



## PLC Program (POUs)



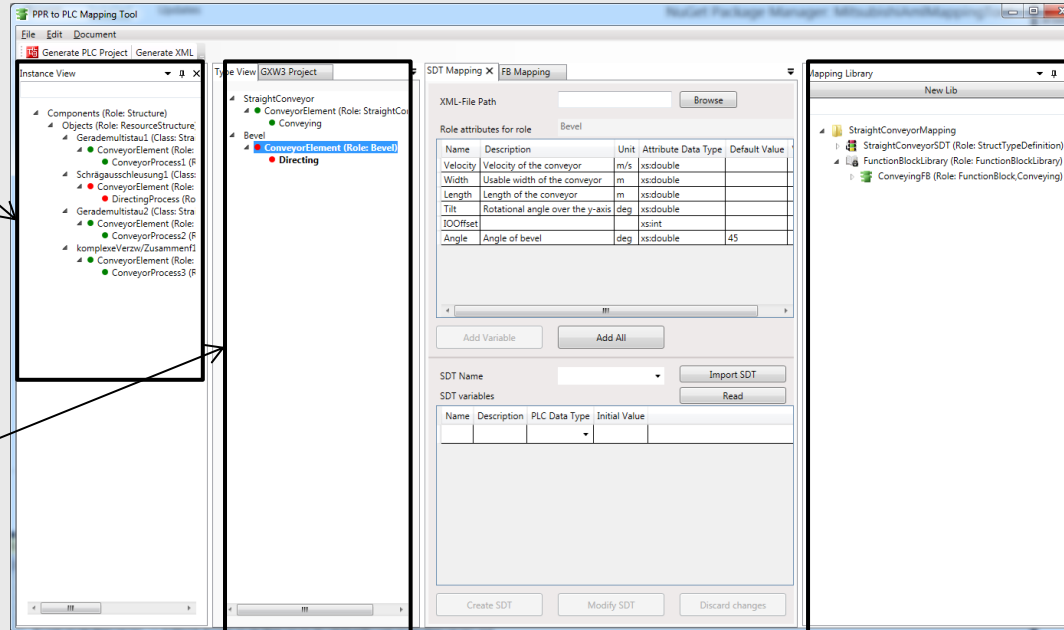
## ② Mapping of Internal Elements to Instances

- Based on the Internal Elements in the Instance Hierarchy, which implements a Resource or a Process role, instances of SDTs and FBs are generated in the PLC program
  - Generate a POU for each resource
  - Create an Instance of the SDT mapped to this resource in the POU
    - Assign the values of the AML Attributes to the members of the instantiated SDT
  - Create an Instance of the FB mapped to the process in the POU
    - Assign the values of the AML Attributes to the members of the instantiated FB



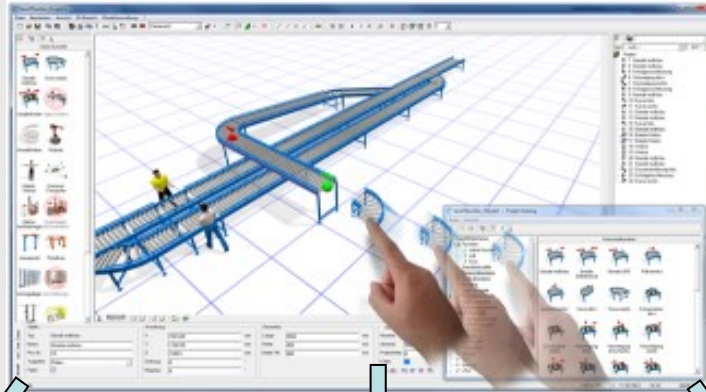
# Defining a reusable mapping and store it in AutomationML

- Development of a prototype tool that can import AutomationML Instance Hierarchy and extract Internal Elements with a PPR Relation
- Create a mapping library based on the Role Classes and System Unit Classes of the Internal Elements which can be reused when other Instance Hierarchies are imported



# Where to get such data from?

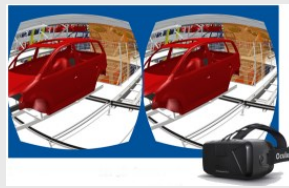
Design 3D Layouts by wide ranged libraries for logistics and production planning



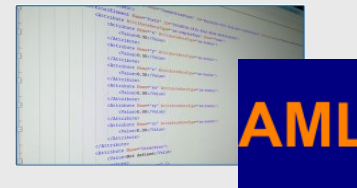
- Import of 2D/3D
- Design 3D Layout
- Create Animated 3D Visualization
- Present Videos/VirtualReality
- **Export of plant configuration**



Animated Videos

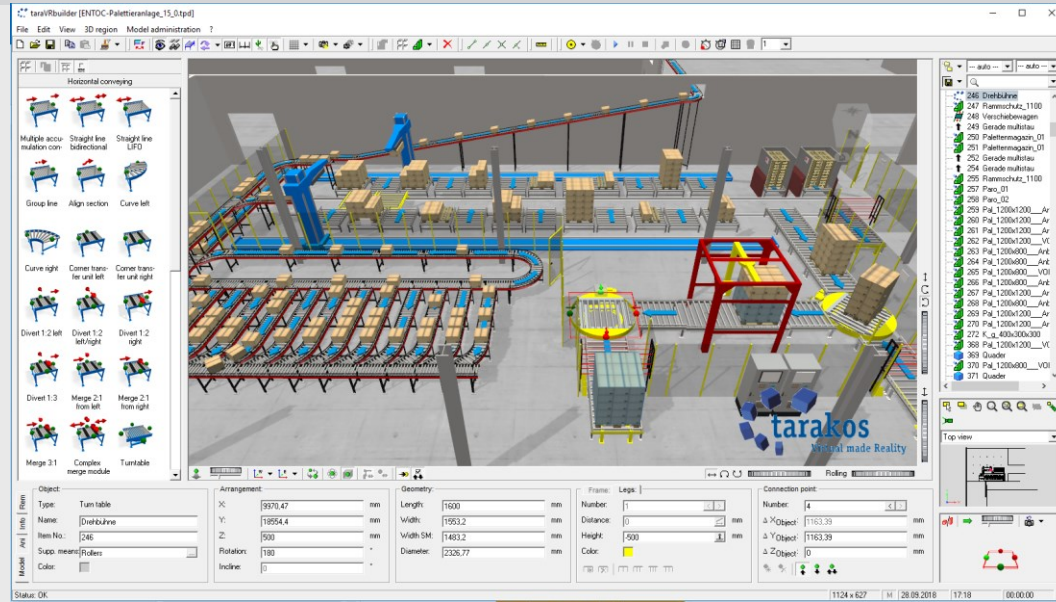


Virtual Reality



Plant Configuration

# Setup the factory layout



- Create 3D factory layout
- taraVRbuilder - Object types
- Position of all objects
- Attributes of taraVRbuilder objects



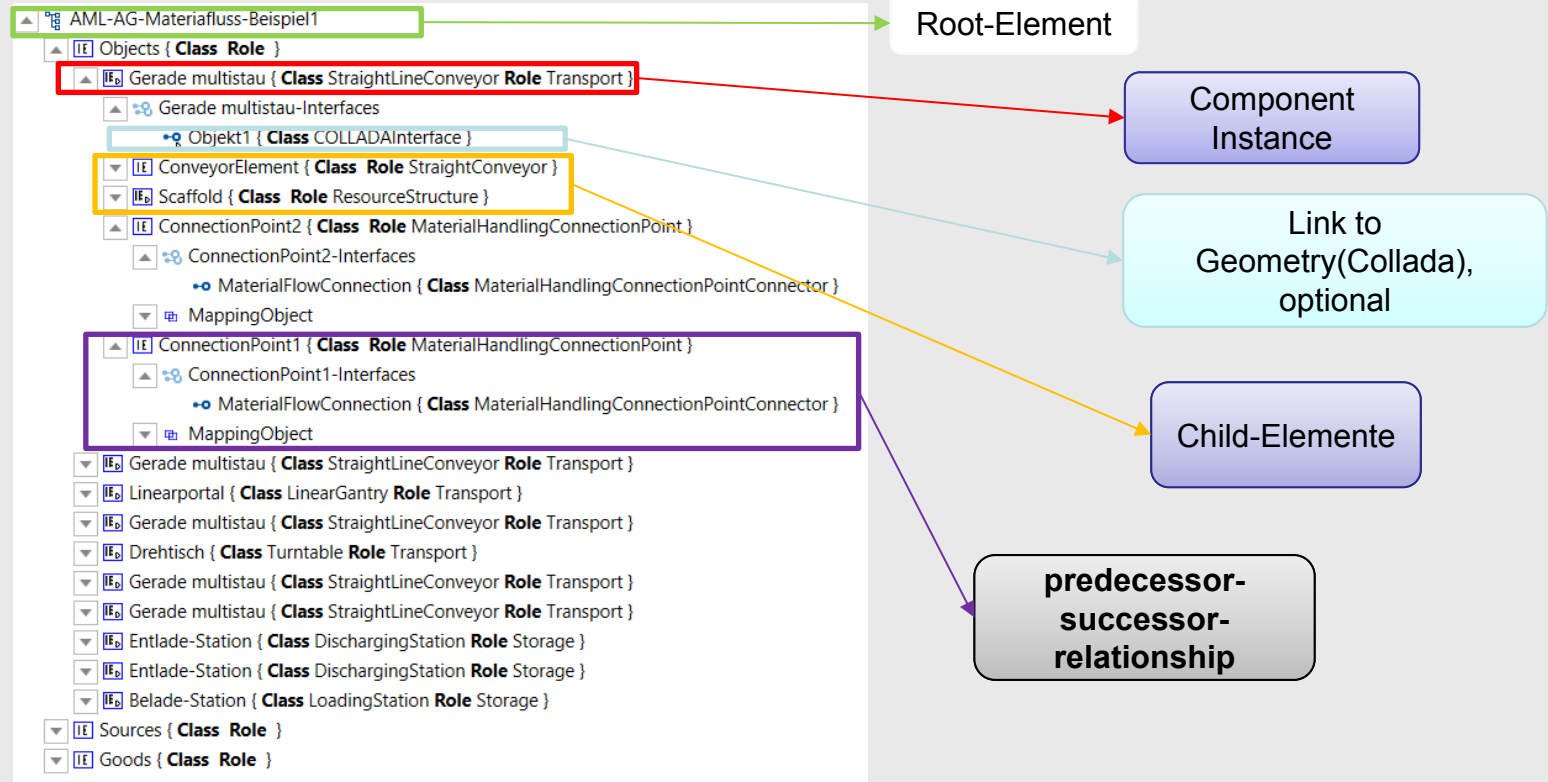
Object:		Arrangement:		Geometry:	
Type:	Turn table	X:	9970.47 mm	Length:	1600 mm
Name:	Drehbühne	Y:	18554.4 mm	Width:	1553.2 mm
Item No.:	246	Z:	500 mm	Width SM:	1483.2 mm
Supp. means:	Rollers	Rotation:	180 °	Diameter:	2326.77 mm
Color:		Incline:	0 °		

Position, Shape, Size (Ressource)

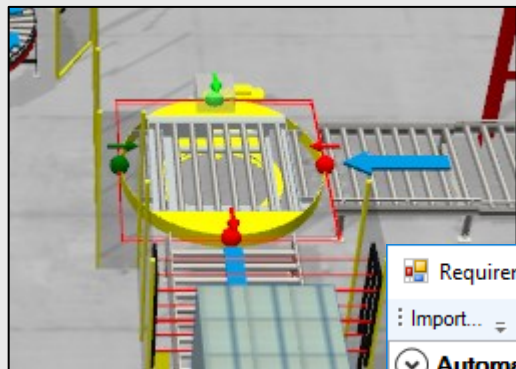
Speed:		Priority:		Distribution:		Mode of operation:	
Convey:	1 m/s	Strategy:	FIFO	Mode of good:	Any type	Capacity:	1
Turn:	10 1/min					Rotate good:	0 °

Behaviour (Process)

# taraVRbuilder-AML-Export - Base Structure



# Add your individual RoleClasses



- Select object in factory layout
- Add requirements from imported RoleClassLibs to objects
- Optional: Set values for RoleClass-Attributes

Imported RoleClassLib

ProcessRole

RessourceRole

RequirementsDialog

Import...

☒ AutomationMLBaseRoleClassLib Delete

☒ SampleRoleClassLib Delete

☒ Name: Moving  
Base: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Process  
Description:

☒ Name: Turning  
Base: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Process  
Description:

☐ Name: StraightConveyor  
Base: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource  
Description:

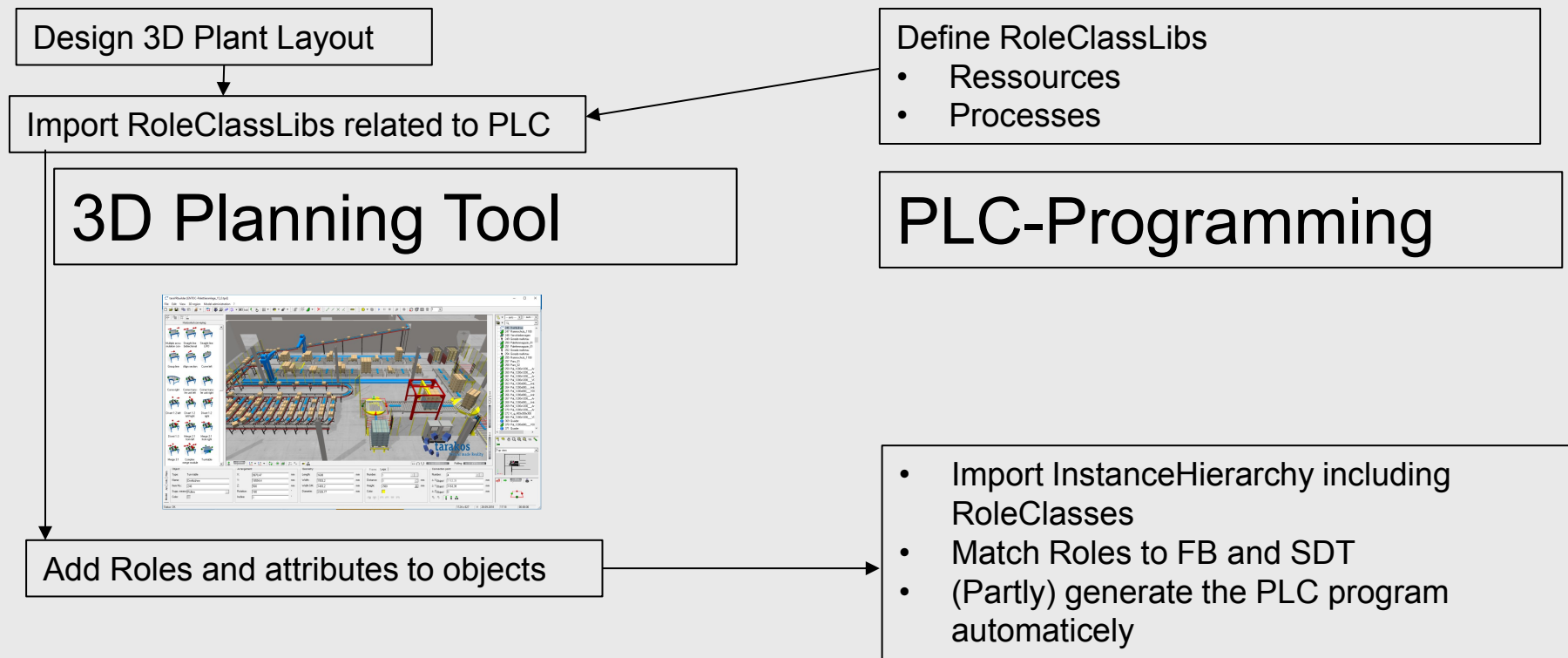
☒ Name: Turntable  
Base: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource  
Description:

☒ Length: 1600

☒ Width: 1500

☒ NumberOfPositions: 2

☒ DistanceBetweenPositions: 90



## ■ Context

- Planning of industrial production systems involves specifications of
  - Functional requirements, e.g. definitions of the materials to be handled, definitions of production and transportation processes, structural definitions of production resources, ...
  - Non-functional requirements, e.g. definition of safety and security constraints, delivery times, budgets, ...
- AutomationML can be used as exchange-format for engineering data

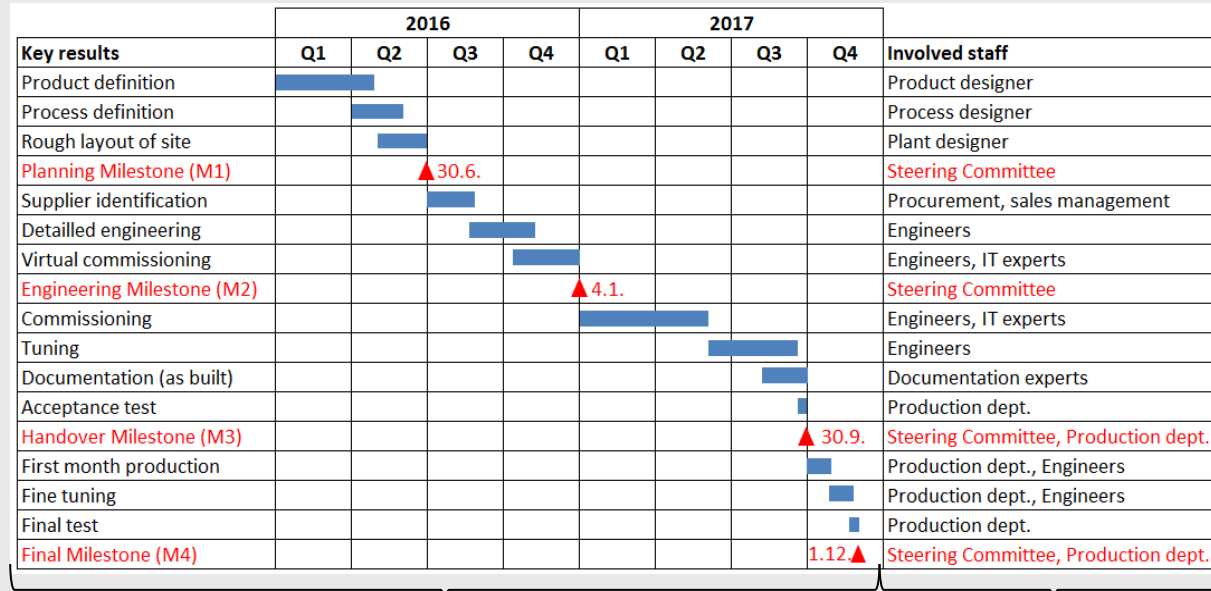
## ■ Problem statement

- How can AutomationML be used to formalize data about the process of creation or modification of industrial production systems?



# Project Planning based on AutomationML

- The goal
  - Formalize milestone plans

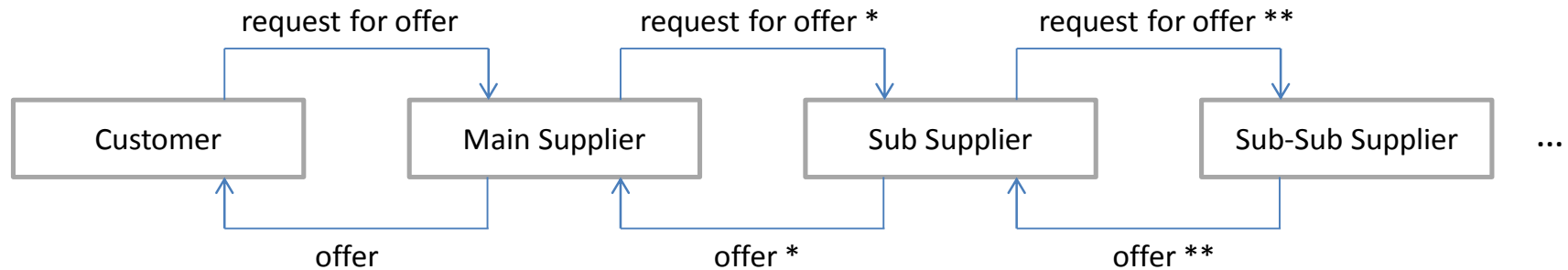


to be exchanged

to be assigned via ERP (not exchanged)

## ■ Motivation

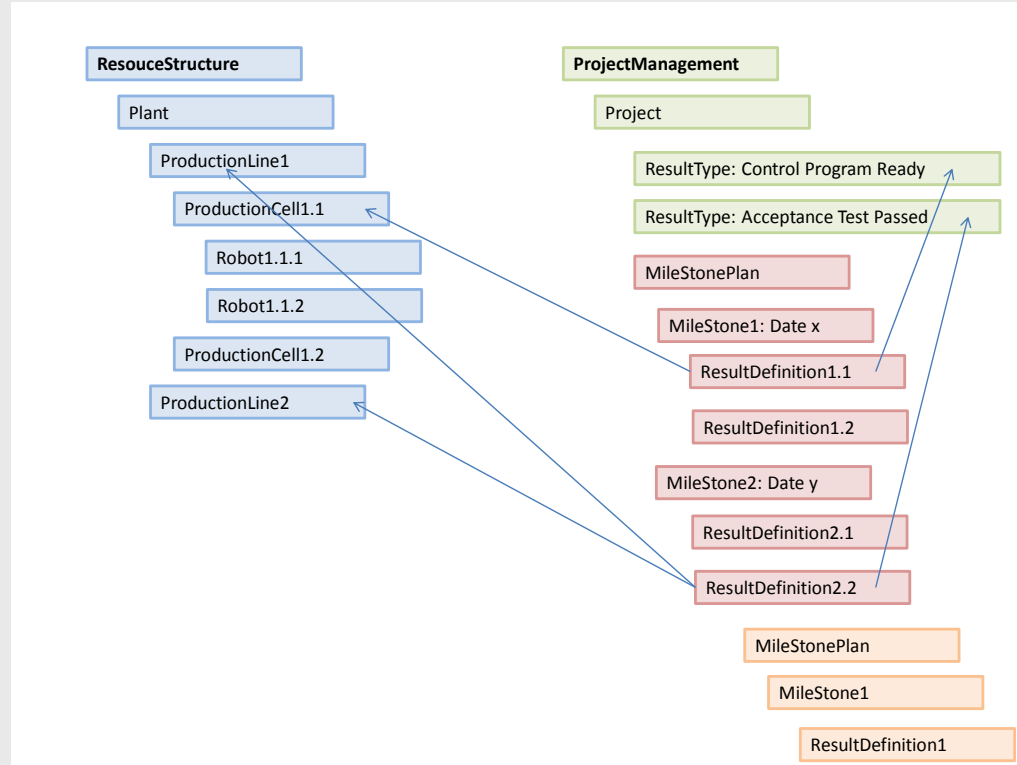
- Planning data is exchanged over large parts of the supplier-chain
- Formalized exchange of time plans improves accuracy of equipment delivery



# Project Planning based on AutomationML

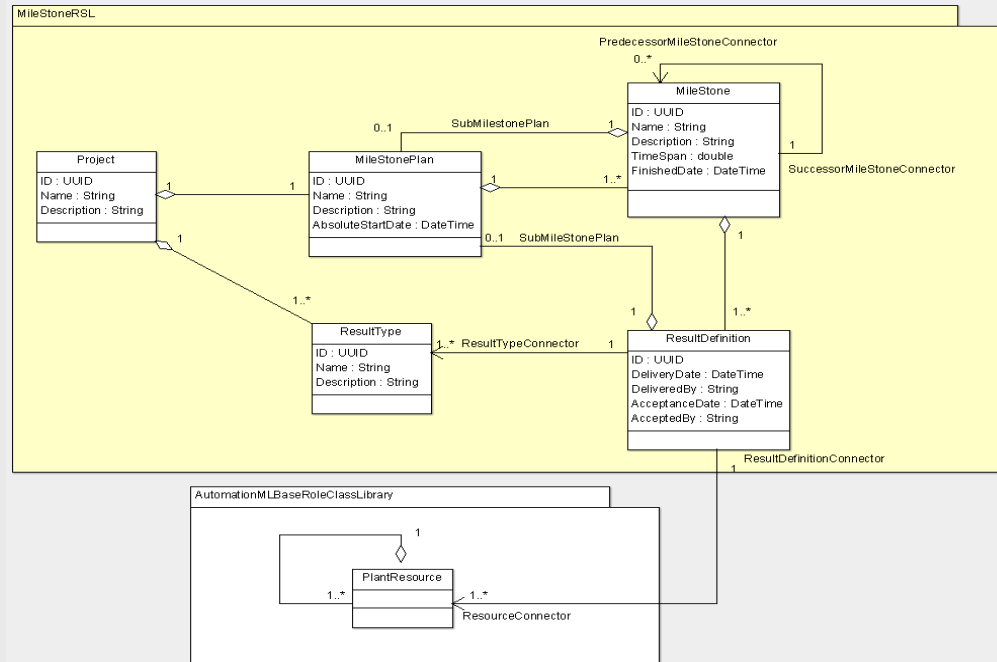
## ■ Solution

- Re-use the plant structure created with other (3D-planning) tools
- Specify a project time plan in terms of milestones
- Add relations between equipment structures and result types to milestones
- Define a formalization of the planning data in AutomationML



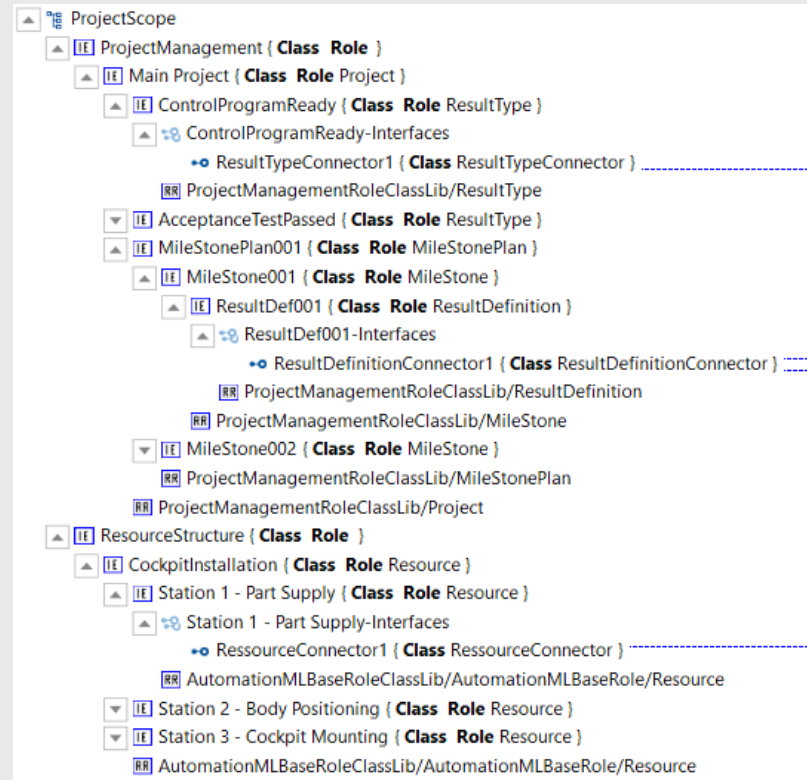
# Project Planning based on AutomationML

- **Detail of Solution**  
Define a formalization of the planning data in AutomationML



# Project Planning based on AutomationML

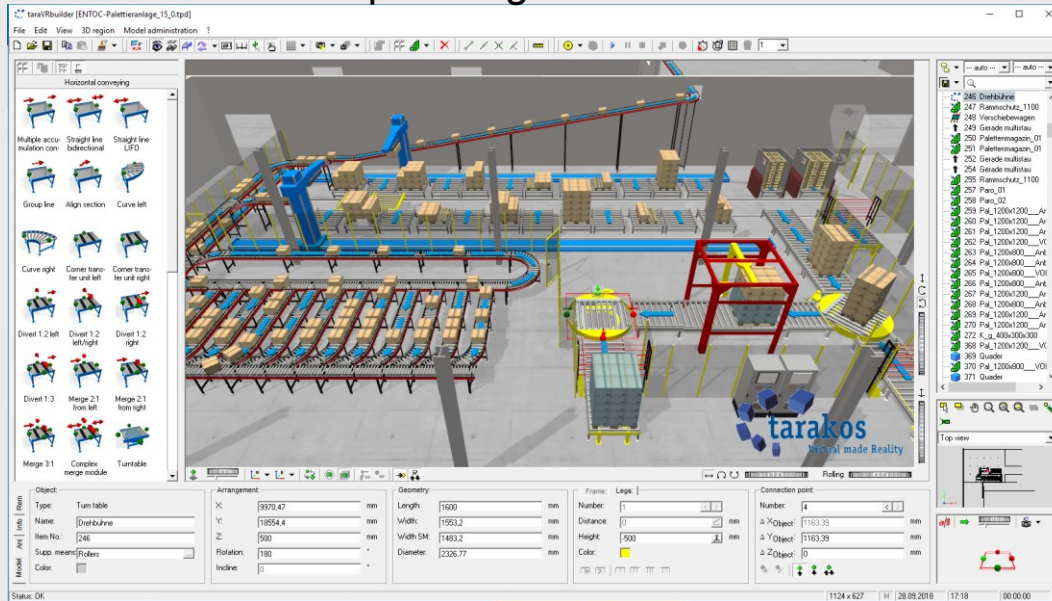
- Hand-writing files is possible, but inconvenient



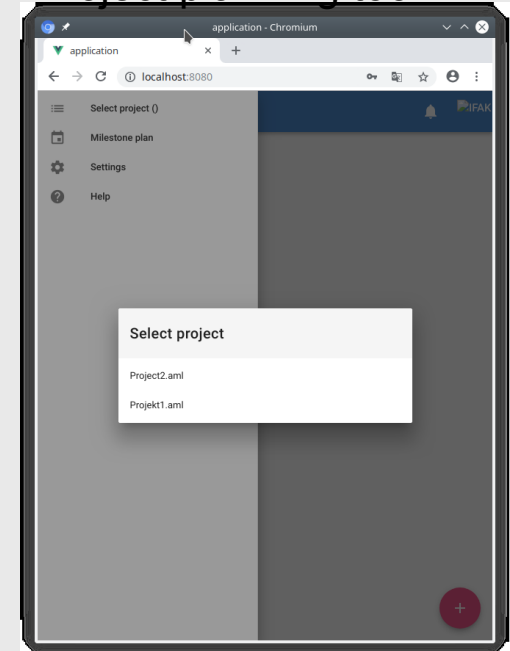
# Project Planning based on AutomationML

- Use the right tools for the planning purposes

## 3D planning tool



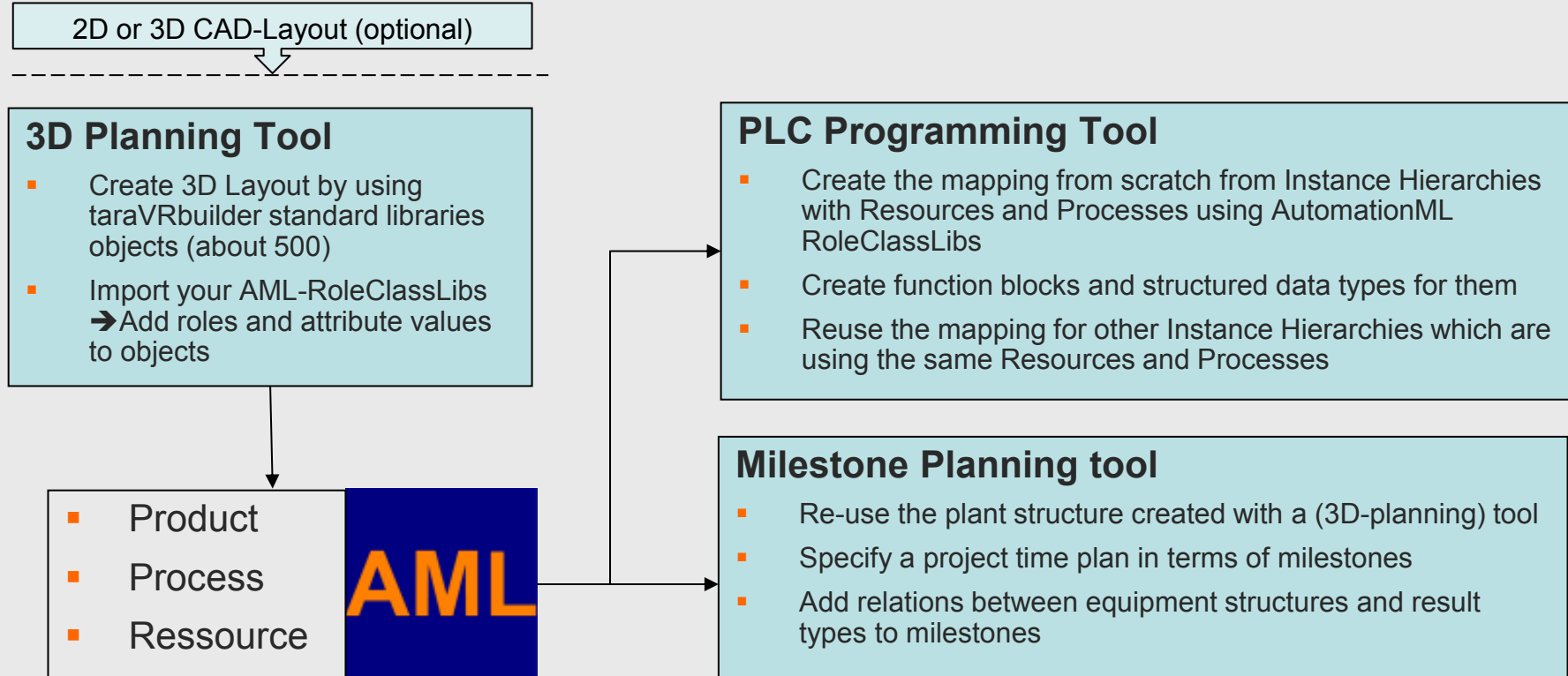
## Project planning tool



## **Demonstration of the data exchange and generation of the PLC program.**

- **Export from Tarakos VRbuilder**
- **Import to Mitsubishi Mapping Tool prototype**
- **Confirm existing mappings and define missing additional mappings**
- **Generate PLC program framework in PLC Engineering Tool GX Works3**





- Based on the PPR Connector and a mapping, the generation of the PLC program framework can be automated and simplified
- Effort for entering the same parameters in several tools can be reduced and errors can be prevented
- After importing and generating the PLC program framework the Instance Hierarchy could be extended the reference into the PLC program
- This could enable a roundtrip engineering where changes in one tool can be reflected back to the others
  - Parameter changes in in the PLC tool could be reflected back to the origin tool
  - Additions to the factory layout can be imported to the PLC tool and the program can be updated
  - Tools at later stages of the engineering chain could use the AutomationML data which includes factory layout and PLC information as well



# Thanks for your attention!

Parts of this conference contribution (plug-in for taraVR builder and project planning tool) have been funded by the German Federal Ministry of Education and Research (BMBF) in scope of the European ITEA3 project 15015 „ENTOC“ (<https://entoc.eu/>).

