



Exchange of engineering data for communication systems based on AutomationML using an EtherNet/IP™ example

Dipl.-Wirtsch.-Ing. Falko Bendik
Otto-von-Guericke-University Magdeburg

October 14, 2015





Overview

- **Motivation**
- **Requirement Analysis**
- **Overview of AutomationML**
- **General communication system modeling methodology in combination with EtherNet/IP™ example**

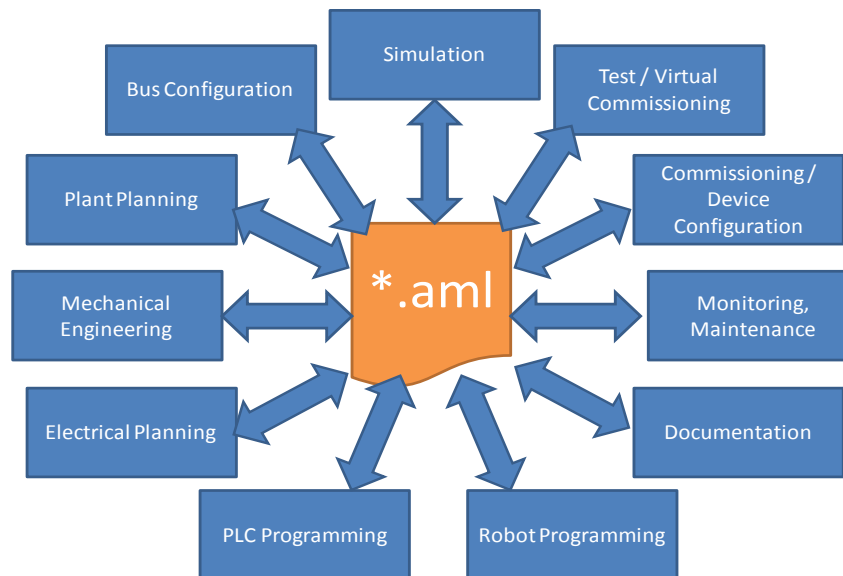


Motivation

Requirements on the engineering of production systems

- *complexity of the systems*
- *amount of data to be handled*
- *high development costs*
- *different applicable efficient engineering processes*
- *different sets of software tools in the development process*
- *exploiting different engineering artifacts*
- *modular control structures / distributed control systems*

**Involved engineering
activities exchanging
engineering artefacts**



General semantic for communication systems with AutomationML

- *A significant cost factor of industrial plants is the engineering process.*
- *To reduce this factor new methods and description tools are needed!*



- Project management (3%)
- Pre-commissioning (2%)
- Assembly (11%)
- Bought-in parts (28%)
- Geometry simulation (1%)
- Robot programming offline/online (5%)
- Engineering [incl. Commissioning] (50%)

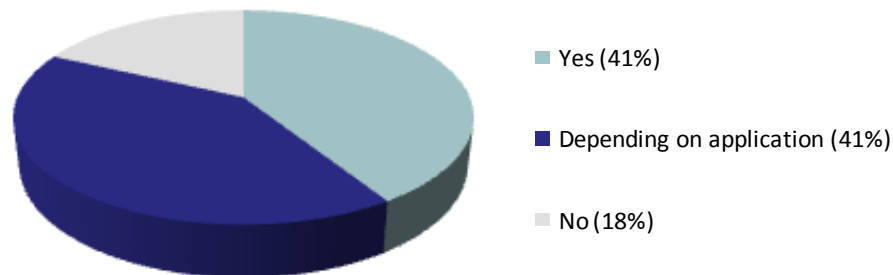
Motivation

Where are the avoidable costs exactly?

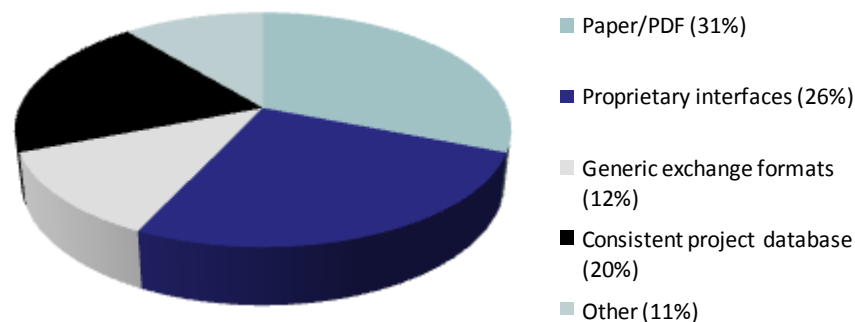
• A survey reveals:

- That 82% of the respondents cannot exclude non-redundancy of planning steps.
- That the pdf/paper interface is the most widespread interface with 31%.
- That only 12% of the respondents use standardized interfaces.

Do redundancies exist in the engineering process?



Which interfaces are currently used?





Motivation

Solutions for an efficient engineering of production systems

- *highly sophisticated engineering tools*
- *fault free creation of engineering artifacts*
- *consistent exchange of engineering data*
 - ***planning of communication systems, addressing and communication package structure***
- *device configuration data and structures describing the communication network*
- *contain topological information*
- *information on communication links and their quality*
- *basic semantic libraries and process models of communication models*

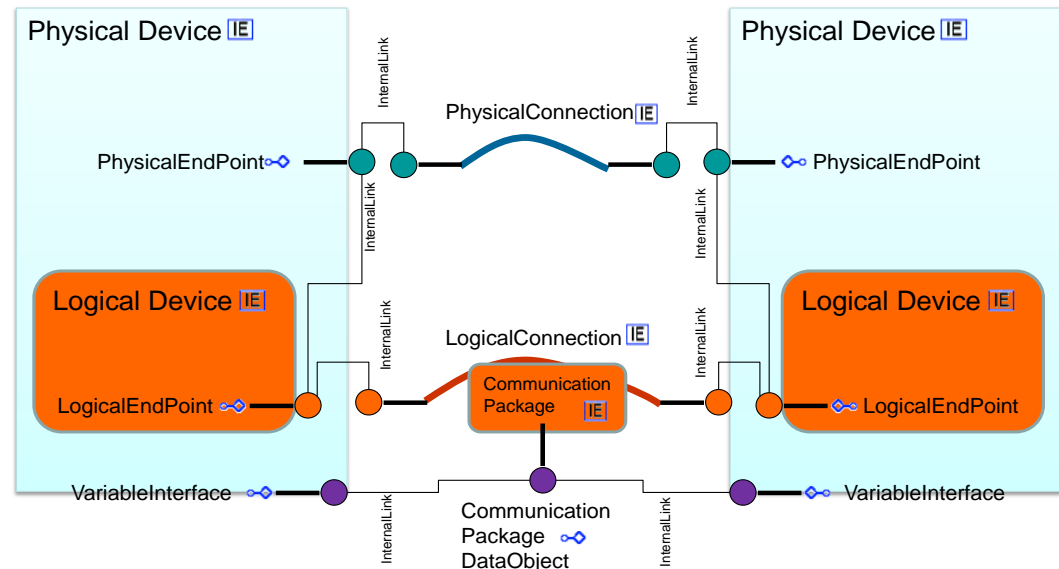
Logical and physical topology of communication systems

Logical modeling

- *addressing of applications and the expected transmission rate of information between application parts*

Physical modeling

- *physical communication links between devices*





Requirement Analysis

What is needed?

- *identify types of information characterizing a communication system*
- *which information should be covered by a method for a neutral exchange of communication system engineering information*
- *data in the engineering process of a production system*
 - *mechanical and electrical engineering data*
 - *control programming “PLC Code”*
 - *virtual commissioning*
 - *commissioning*
- *definition of involved automation devices and their wiring*
- *control programming, runtime information (represented by control variables)*
- *identify additional technical details such as data types, transmission rates etc.*



Requirement Analysis

What is used?

- *electrical engineering: specifying wiring diagrams, wiring*
- *control programming: specify device configurations, variable lists, link descriptions*
- *virtual commissioning: used to test the physical realization of the intended data exchange*
- *commissioning: used variable declaration and linking descriptions for monitoring and diagnosis*





Requirement Analysis

What information is needed to describe the modeling method?

- *control application*
 - *components are linked with each other by exchanging variables*
 - *need to establish logical connections by logical interfaces (namely the corresponding exchanged variables or sets of them)*
 - *identify the requirements in terms of communication properties on the exchange of data within the logical connections*
 - *create properties of the control application components, e.g. processing times, characteristics of logical interfaces, or a port number*



Requirement Analysis

What information is needed to describe the modeling method?

- *control devices*
 - *control application components are running on different physical devices*
 - *the logical connections must be realized by the physical data exchange*
 - *physical device has interfaces for physical connections (necessary cables for communication system realization)*
 - *describe the active and passive infrastructure components such as switches, interface converters, plug, or cable types*
 - *analogy to the control application parts, the logical connections, and the logical interfaces*
 - *represent the properties of the physical devices, the physical interfaces and the physical links*



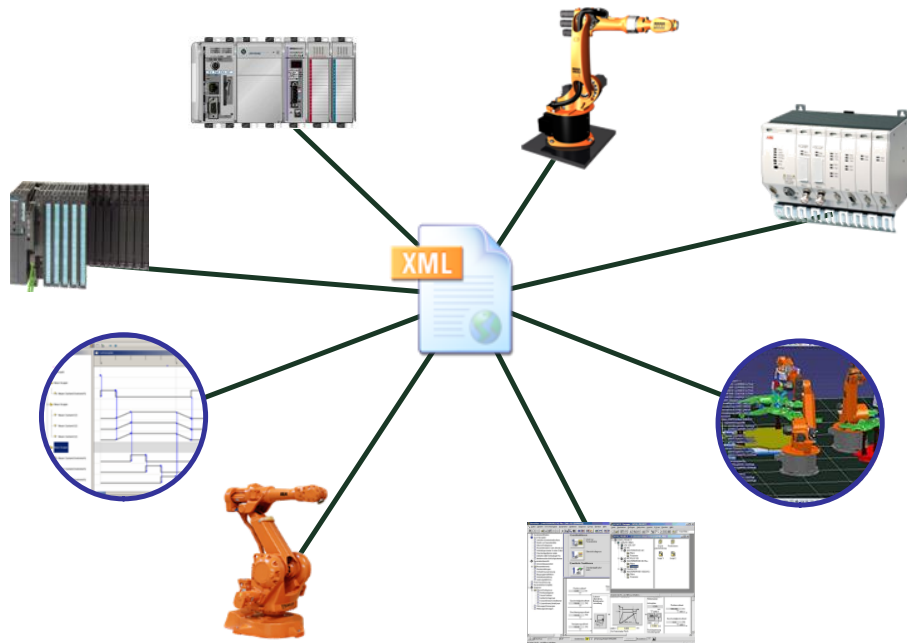
Requirement Analysis

What information is needed to describe the modeling method?

- *general modeling requirements*
 - *represent the relations between logical and physical connections*
 - *map the corresponding structures at least at the used interfaces to each other*
 - *communication system information is transmitted by data packages*
 - *represent the mapping of variables within the data package sender to variables within the data package receiver*

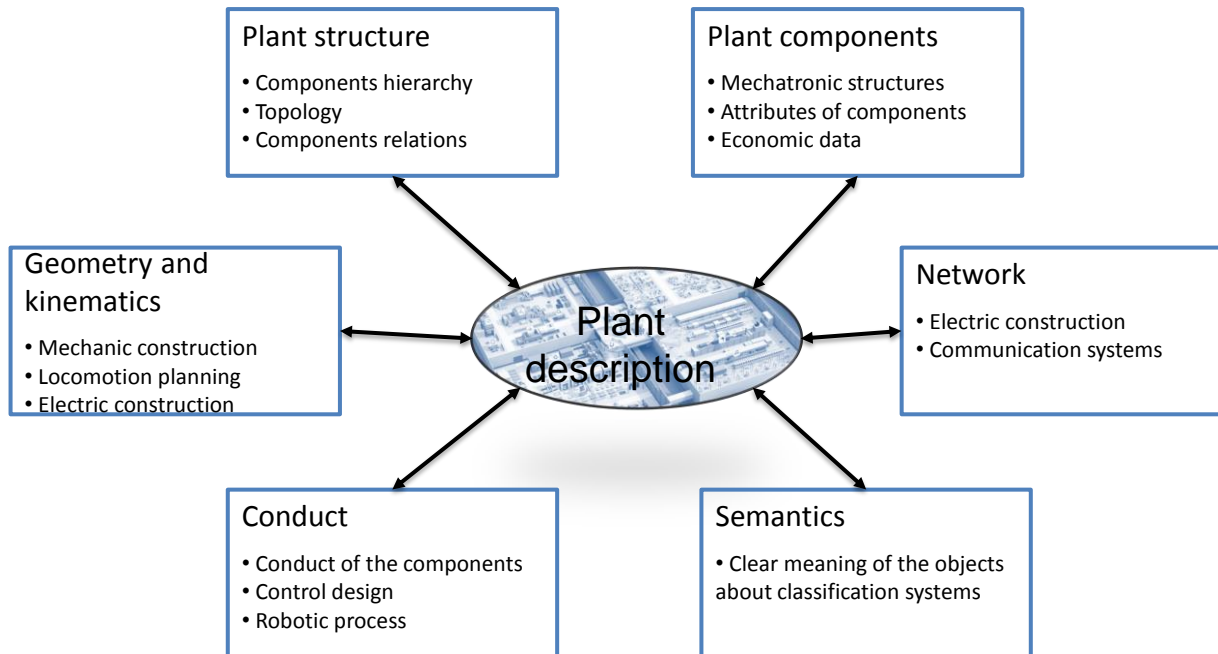
Overview of AutomationML

- *AutomationML is a XML based data format.*
- *It is an international standard (IEC 62714) and free of charge*
- *AutomationML allows a consistent data exchange within different tool chains*
- *It allows the integration of the world of tools into the digital factory of the future*



Overview of AutomationML

Which data contents are covered by AutomationML?





Overview of AutomationML

AutomationML is ...

- *Data format, that allows a data exchange of engineering data of production systems independent of the manufacturer.*
- *Storage format for information.*
- *Connection between different discipline-specific engineering tools with that it is usable in the whole engineering process.*
- *Object orientated and allows the modeling of plant components as data objects summarizing different aspects.*
- *Combination and adaptation of already existing industry formats that were developed for exchange and storage of different engineering aspects.*
- *consistent, distributed document architecture, that enables the handling of large amounts of data and the outsourcing of libraries to external documents.*



Overview of AutomationML

AutomationML is NOT...

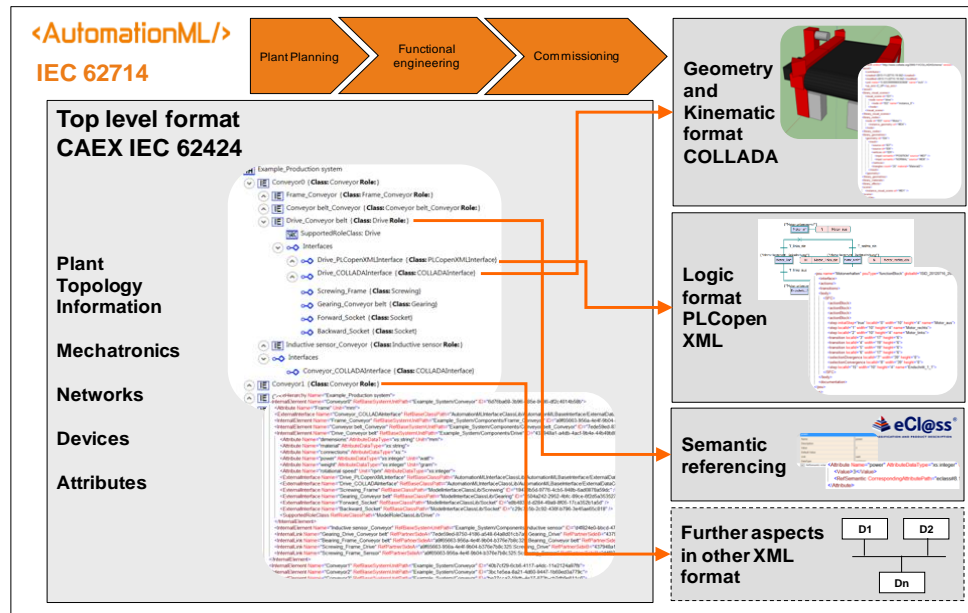
- *Tool functionality*
- *Review of conditions, attribute values, relations, references, or semantic correctness of several data objects*
- *Review and matching of the consistence resp. version of data objects*
- *Automatic standardization of user specific information*
- *Automatic creation of libraries*
- *Automatic management of versions and variants*
- *Project management tool*
- *Project management database*

► *But it allows the storage of all data required for that*

Overview of AutomationML

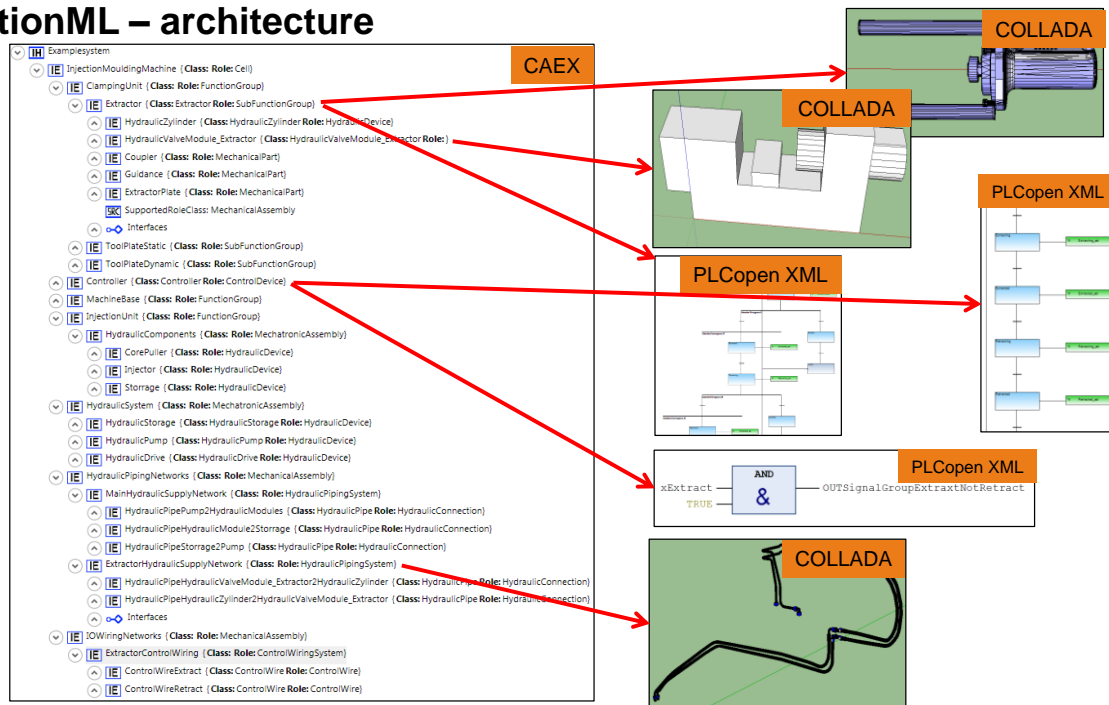
The AutomationML – architecture

- *Based on the use of existing XML data formats*
 - CAEX - structure/relation of the plant objects
 - COLLADA - geometry and kinematics
 - PLCopen XML - behavior



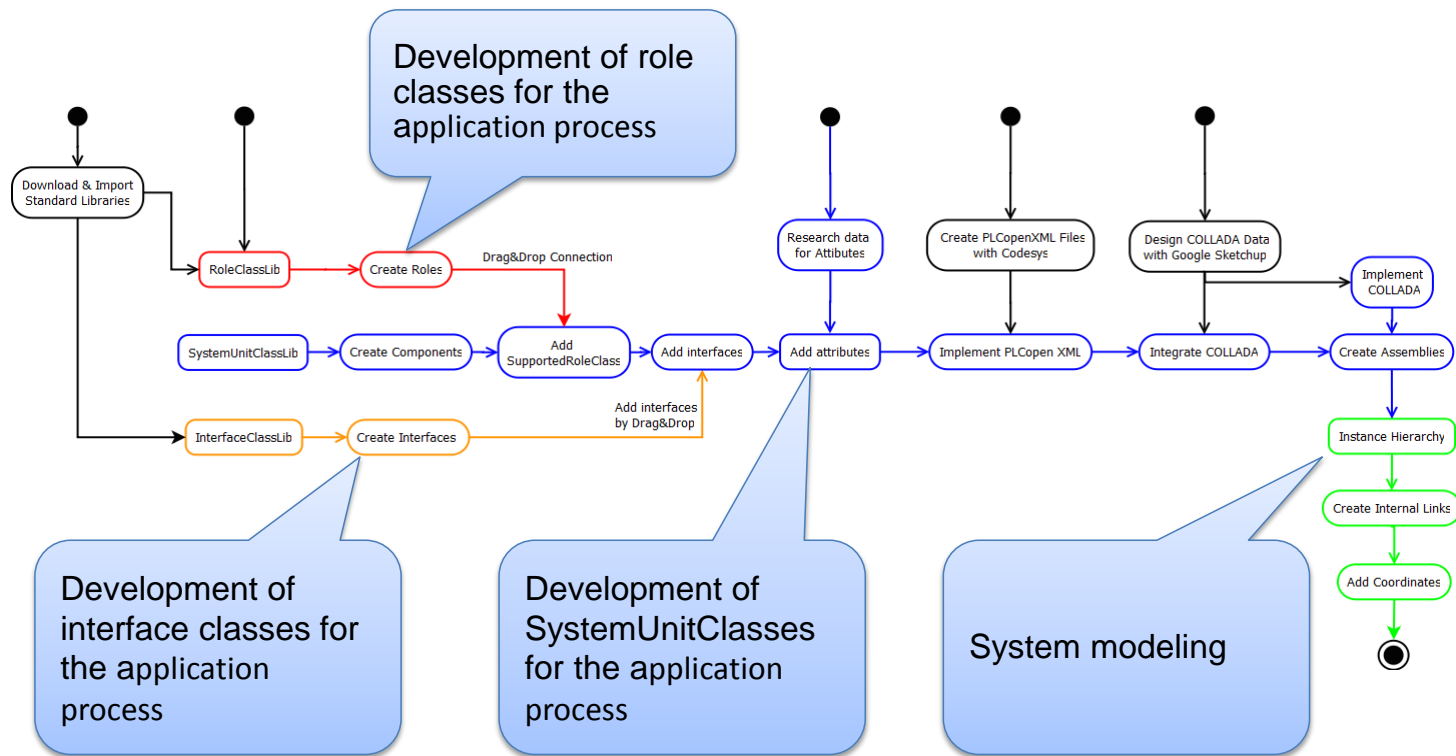
Overview of AutomationML

The AutomationML – architecture



Overview of AutomationML

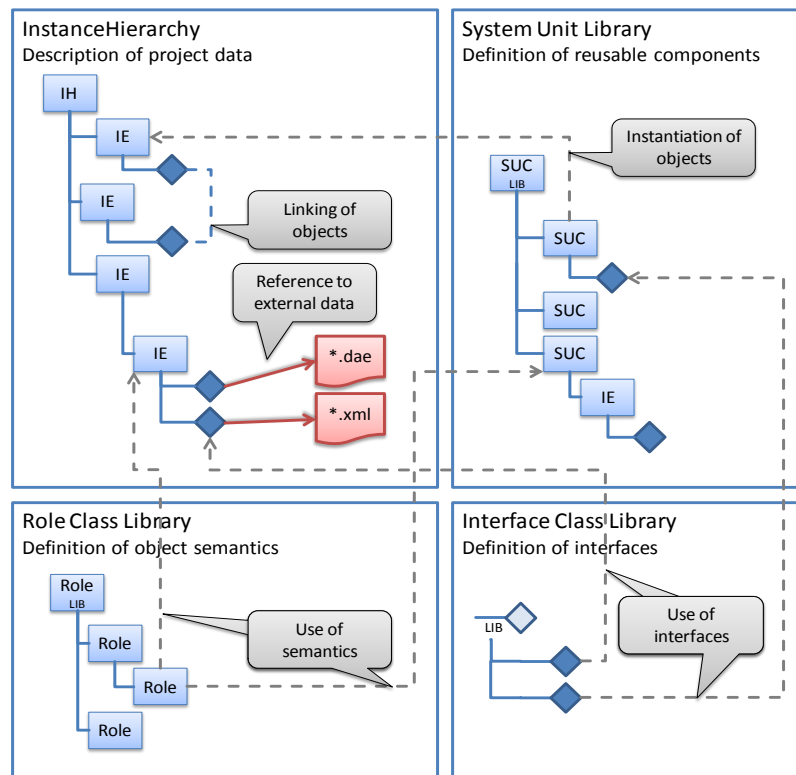
Application process



Overview of AutomationML

Topology description with CAEX

- Definition of the meaning of objects by roles
 - Components
 - Interfaces
 - Roles
- Representation of project data as project tree
- Integration of object descriptions as attributes
- Relations between objects and references to external documents



Overview of AutomationML

Topology description with CAEX

Description of
planning data

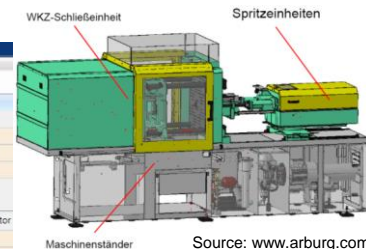
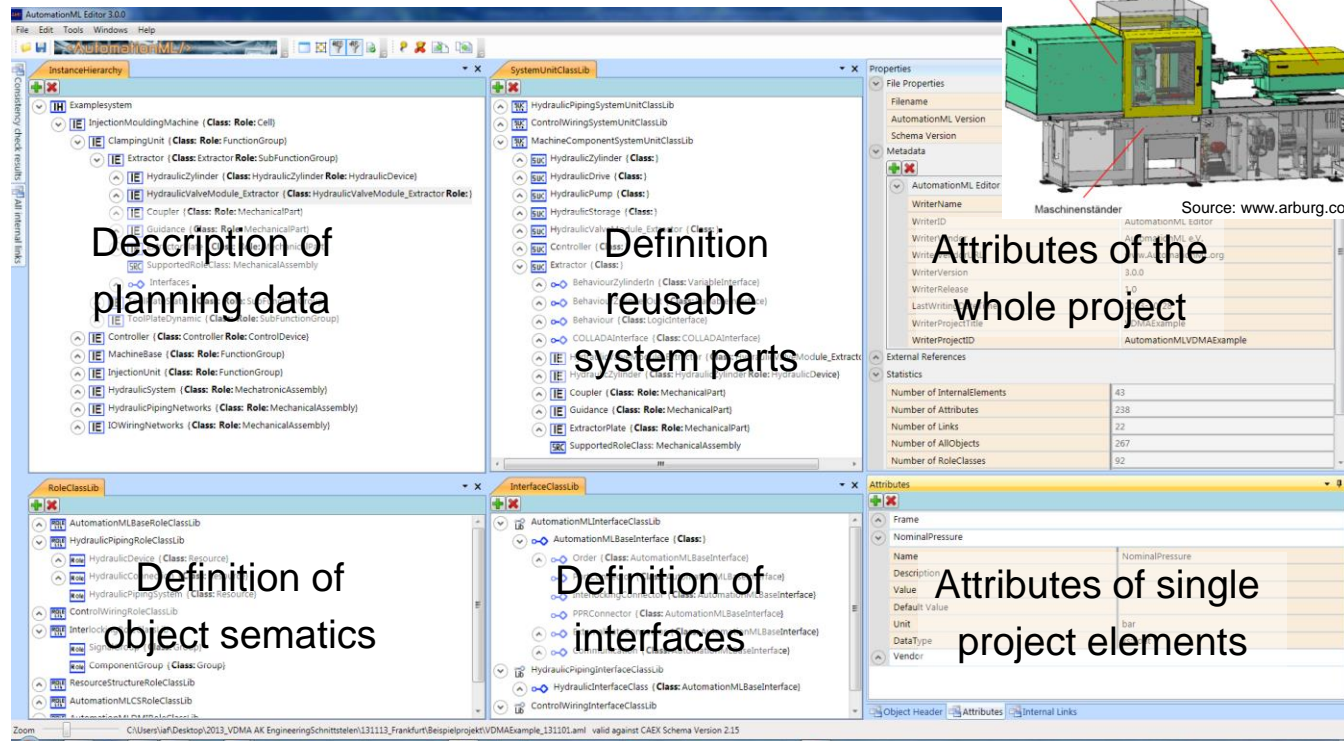
Definition
reusable
system parts

Definition of
object semantics

Definition of
interfaces

Attributes of the
whole project

Attributes of single
project elements

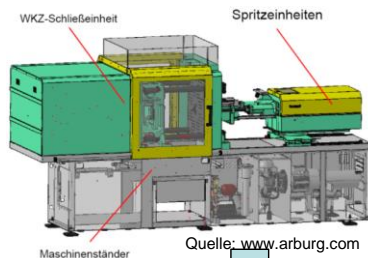



The screenshot displays the AutomationML Editor 3.0.0 interface. The main workspace is divided into several panels:

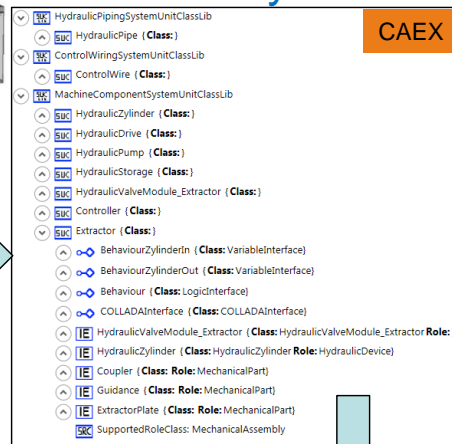
- InstanceHierarchy:** Shows a tree structure of the machine assembly, including components like InjectionMouldingMachine, ClampingUnit, Extractor, HydraulicCylinder, HydraulicDrive, HydraulicPump, HydraulicStorage, HydraulicValveModule_Extractor, Controller, MachineBase, InjectionUnit, HydraulicSystem, HydraulicPipingNetworks, and IOWiringNetworks.
- SystemUnitClassLib:** Lists system unit classes such as HydraulicPipingSystemUnitClassLib, ControlWiringSystemUnitClassLib, MachineComponentSystemUnitClassLib, HydraulicCylinder (Class), HydraulicDrive (Class), HydraulicPump (Class), HydraulicStorage (Class), HydraulicValveModule_Extractor (Class), Controller (Class), BehaviourZylinderIn (Class: VariableInterface), Behaviour (Class: LogicInterface), COLLADAInterface (Class: COLLADAInterface), HydraulicValveModule_Extractor (Class: HydraulicValveModule_Extractor), Extractor (Class), COLLADAInterface (Class: COLLADAInterface), HydraulicCylinder (Class: HydraulicCylinder), Coupler (Class: MechanicalPart), Guidance (Class: MechanicalPart), and SupportedRoleClass: MechanicalAssembly.
- RoleClassLib:** Lists role classes such as AutomationMLBaseRoleClassLib, HydraulicPipingRoleClassLib, HydraulicDevice (Class: FunctionGroup), HydraulicCoupler (Class: MechanicalPart), HydraulicPipingSystem (Class: MechanicalPart), ControlWiringRoleClassLib, Interlock (Class: MechanicalPart), Sign (Class: MechanicalPart), ComponentGroup (Class: Group), ResourceStructureRoleClassLib, and AutomationMLCSRoleClassLib.
- InterfaceClassLib:** Lists interface classes such as AutomationMLInterfaceClassLib, AutomationMLBaseInterface (Class), Order (Class: AutomationMLBaseInterface), PPRConnector (Class: AutomationMLBaseInterface), HydraulicPipingInterfaceClassLib, HydraulicInterfaceClass (Class: AutomationMLBaseInterface), and ControlWiringInterfaceClassLib.
- Properties:** Displays metadata for the current project, including File Properties (Filename, AutomationML Version, Schema Version) and Metadata (AutomationML Editor, WriterName, WriterID, WriterVersion, WriterRelease, LastWritten, WriterProjectID, and WriterProjectID).
- Attributes:** Displays attributes for the current project element, including Frame, NominalPressure, Name, Description, Value, Default Value, Unit, DataType, and Vendor.

Overview of AutomationML

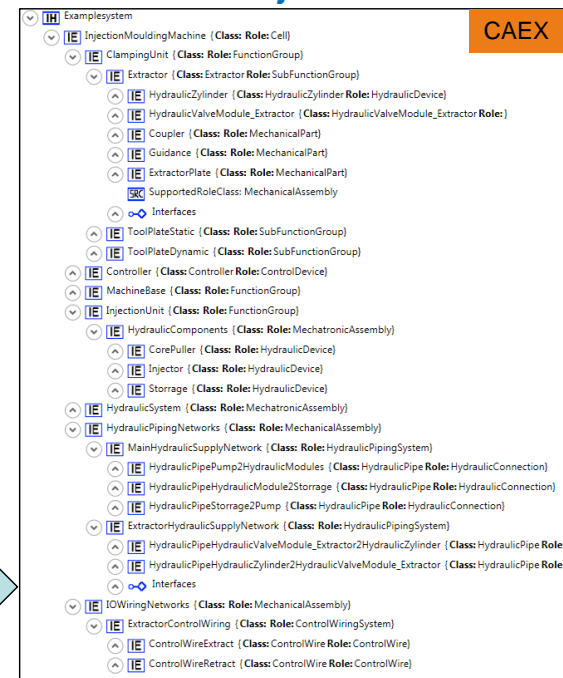
Topology example



Component library



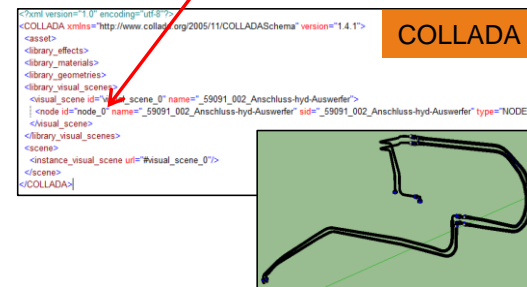
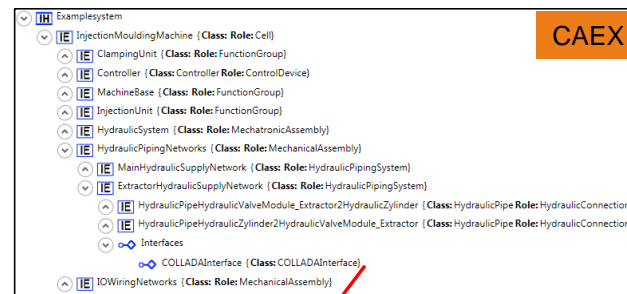
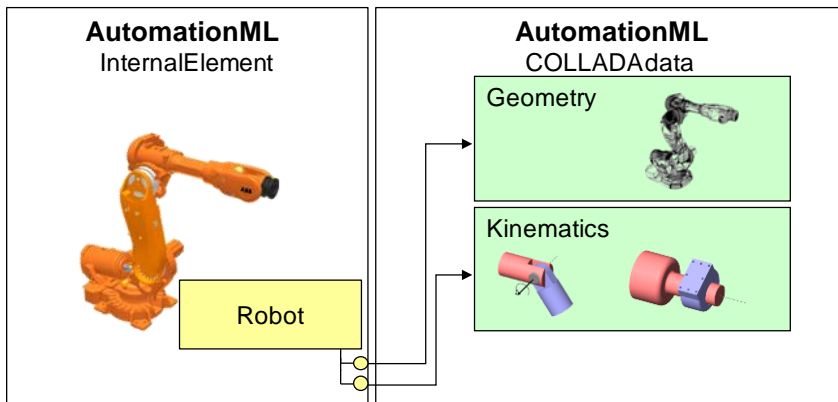
Project



Overview of AutomationML

Geometry and kinematic description with COLLADA

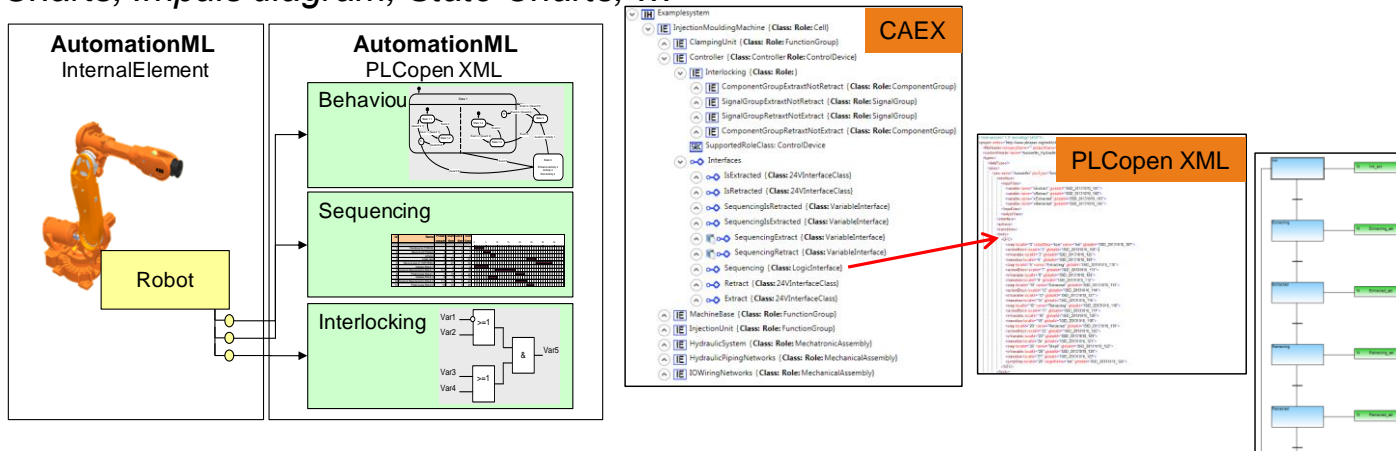
- COLLADA is standardized as ISO/PAS 17506
 - KHRONOS
- Originally developed for the gaming industry
 - Main driver: Sony



Overview of AutomationML

Behavior description with PLCopen XML

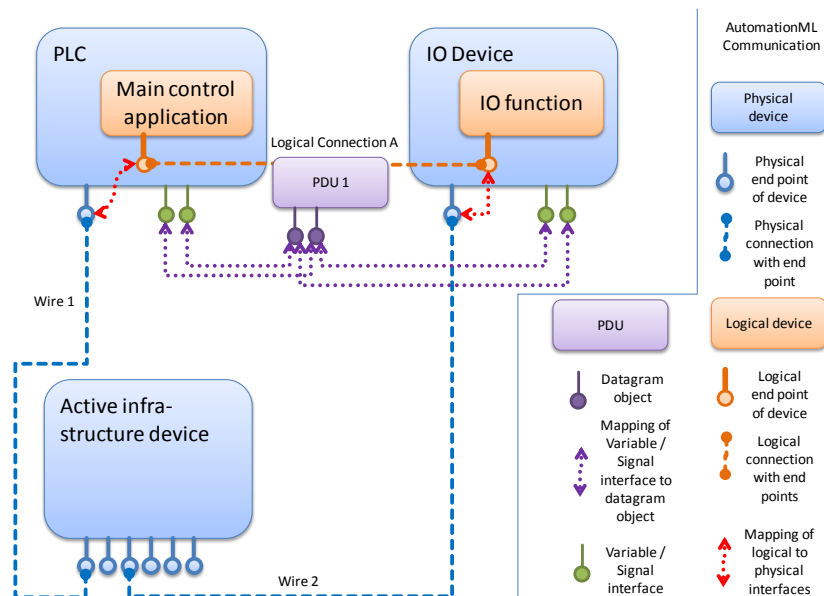
- *PLCopen XML based on IEC 61131-3*
- *Derived from the SPS world*
- *Allows the description of conduct about different types of models like Gantt Charts, PERT Charts, Impuls diagram, State Charts, ...*



General communication system modeling methodology

Networks are the integral part of modern communication systems

- *Combination of used elements*
- *Requirement: Illustration of*
 - *Logical network structures at application level*
 - *Physical network structures for the technical realization of interactions*
 - *Relationship between both views*
 - *Descriptive attributes of both views*



General communication system modeling methodology

Example network with EtherNet/IP™

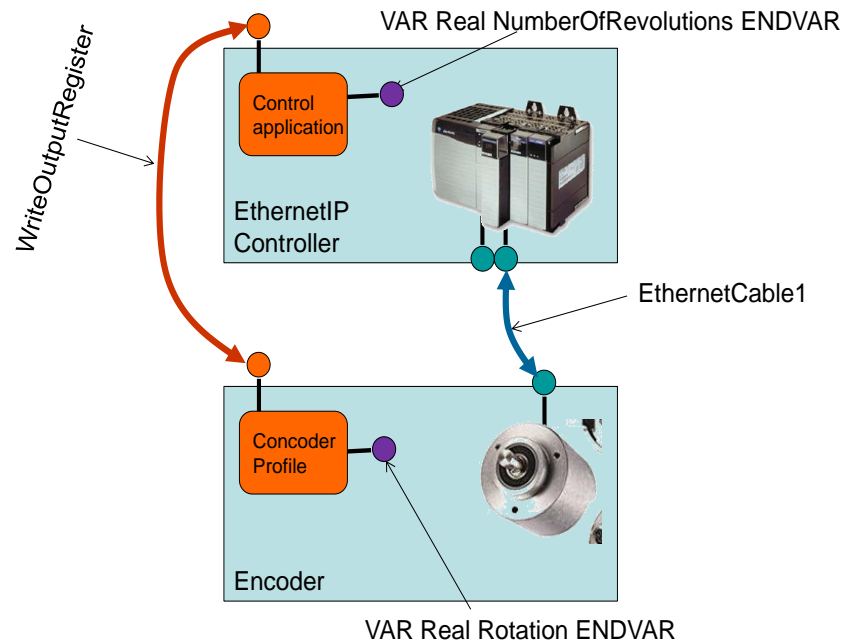
0. Application of predefined basic libraries

roles

- ▼ **ROLE** CommunicationRoleClassLib
 - ▼ **Role** PhysicalDevice { **Class:** AutomationMLBaseRole }
 - Role** PhysicalPortList { **Class:** AutomationMLBaseRole }
 - Role** PhysicalConnection { **Class:** AutomationMLBaseRole }
 - Role** PhysicalNetwork { **Class:** AutomationMLBaseRole }
 - ▼ **Role** LogicalDevice { **Class:** AutomationMLBaseRole }
 - Role** LogicalPortList { **Class:** AutomationMLBaseRole }
 - Role** LogicalConnection { **Class:** AutomationMLBaseRole }
 - Role** LogicalNetwork { **Class:** AutomationMLBaseRole }

interfaces

- ▼ **Lib** CommunicationInterfaceClassLib
 - PhysicalEndPoint** { **Class:** Communication }
 - LogicalEndPoint** { **Class:** Communication }

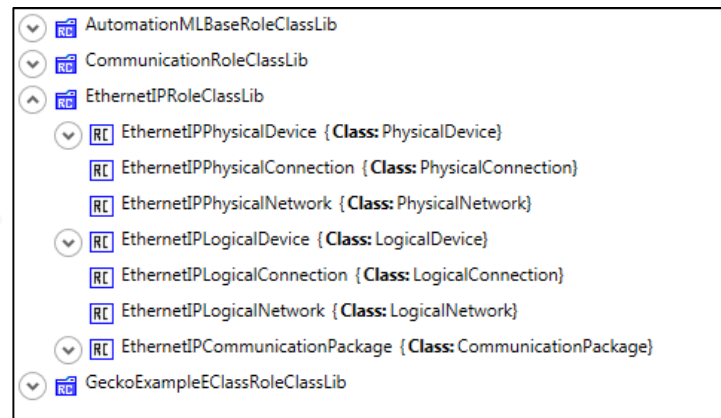
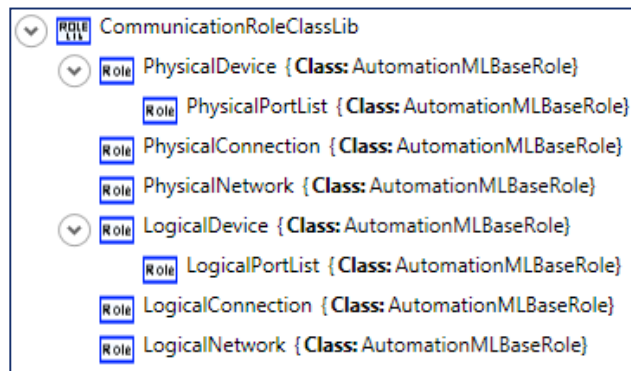


General communication system modeling methodology

Example network with EtherNet/IP™

1. Definition of the role classes for the corresponding use case

► RoleClassLib

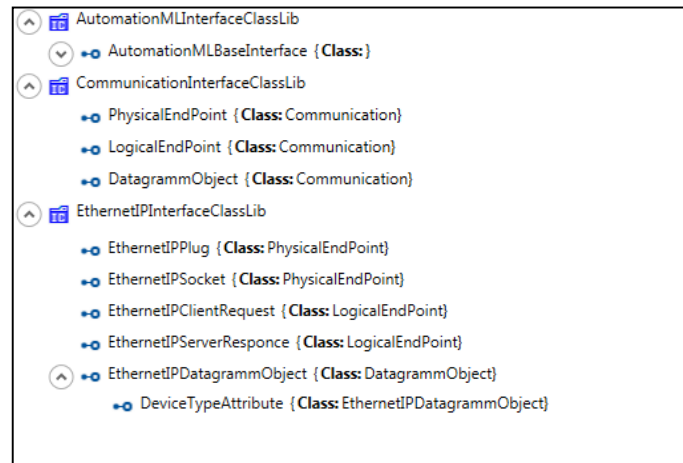
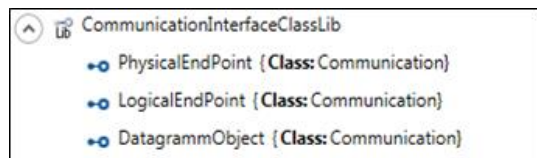


General communication system modeling methodology

Example network with EtherNet/IP™

2. Definition of the interfaces for the corresponding use case

► InterfaceClassLib

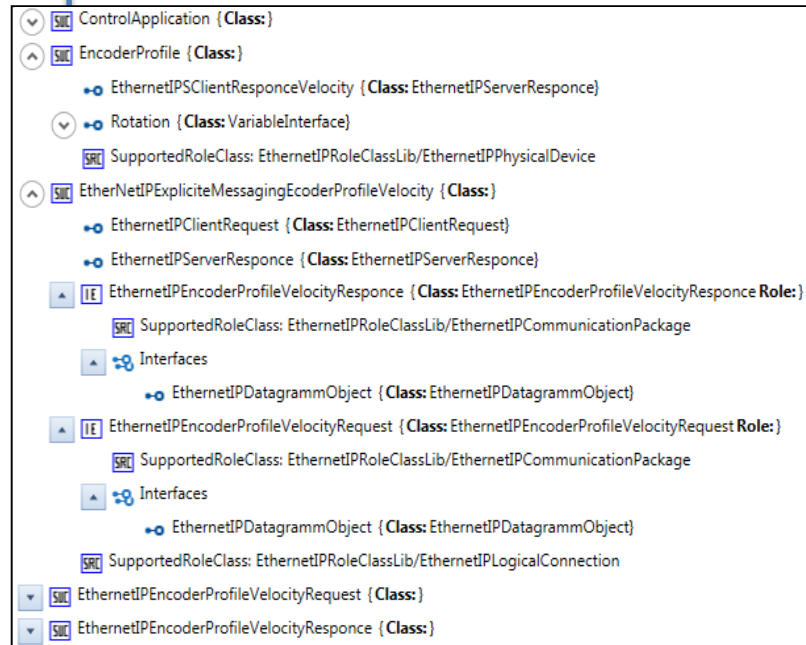
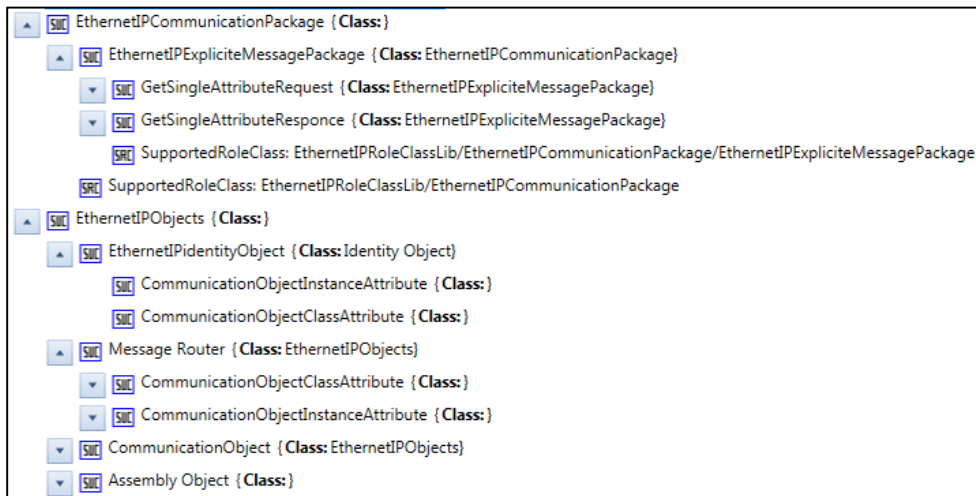


General communication system modeling methodology

Example network with EtherNet/IP™ (Part 1&Part 2)

3. Modeling of the used devices and connections

► SystemUnitClassLib

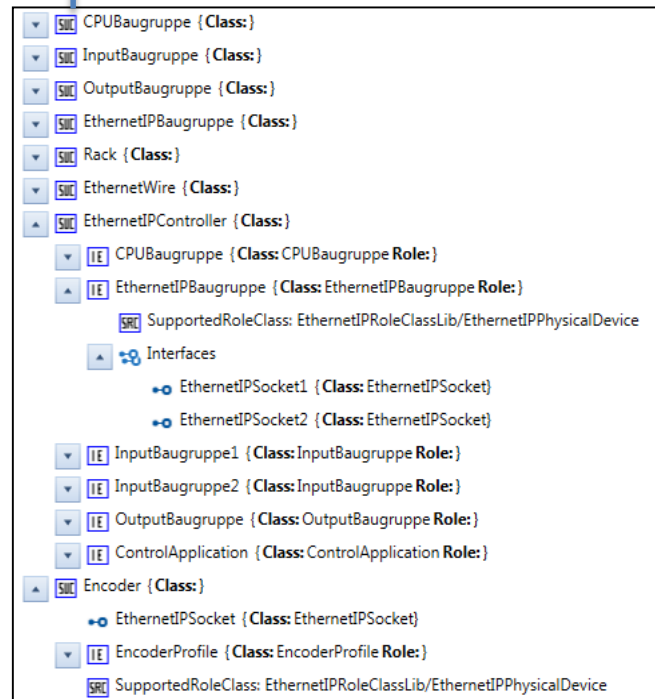
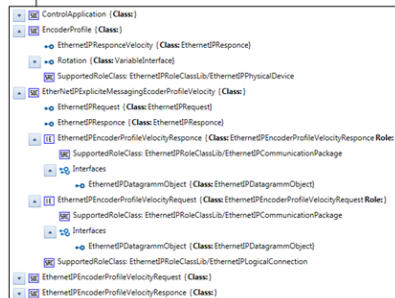


General communication system modeling methodology

Example network with EtherNet/IP™ (Part 3)

3. Modeling of the used devices and connections

► SystemUnitClassLib

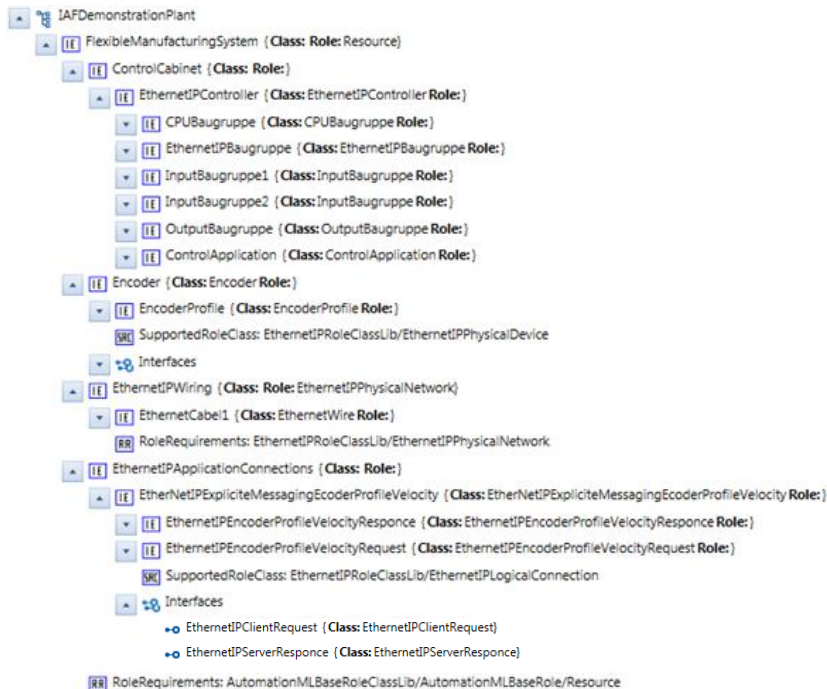


General communication system modeling methodology

Example network with EtherNet/IP™

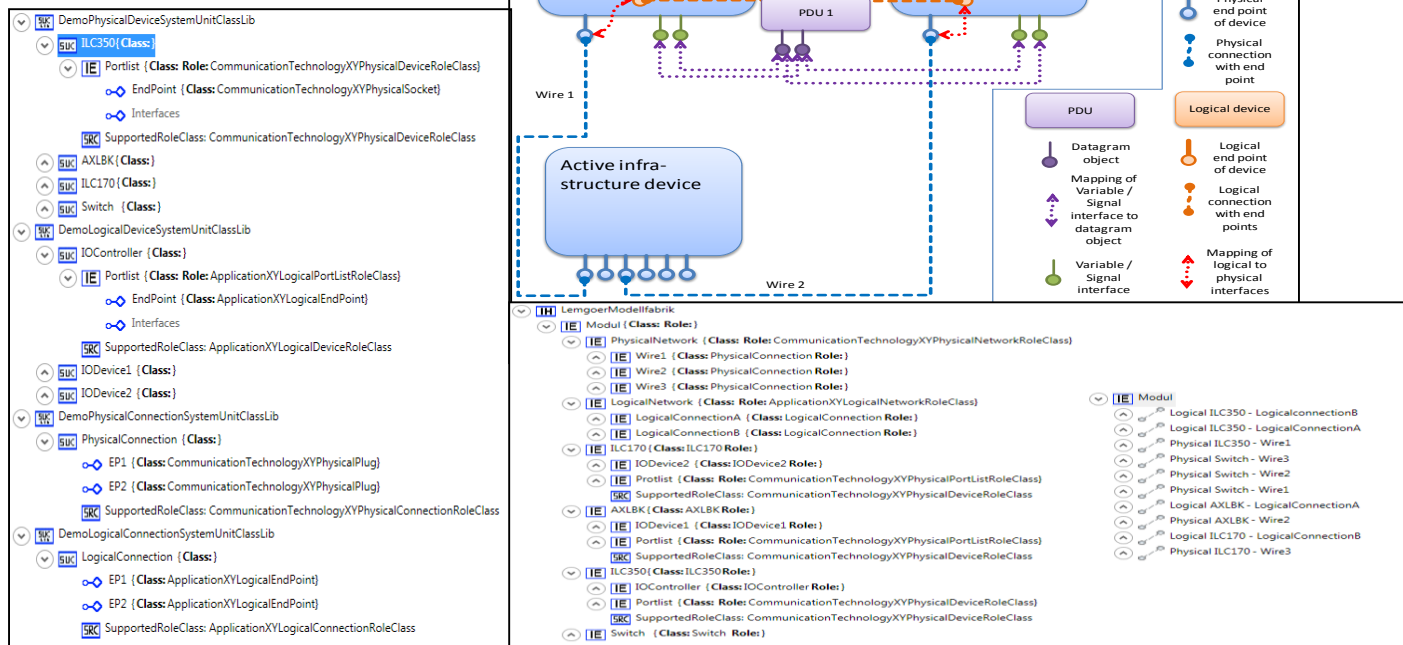
4. Modeling of the system

► InstanceHierarchy



General communication system modeling methodology

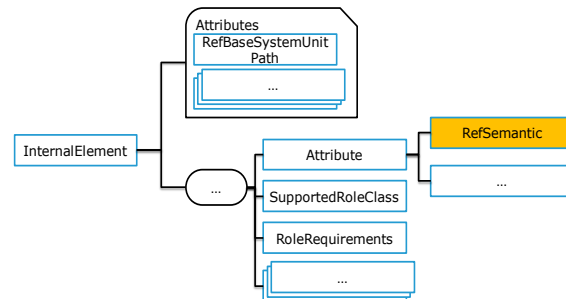
Application to communication network



General communication system modeling methodology

Determination of the semantics for several attributes

| | |
|-------------|--|
| EPath | |
| Name | EPath |
| Description | Address description of requested object in explicite messaging |
| Value | |
| ObjectClass | |
| Name | ObjectClass |
| Description | Object class attribute of EPath |
| Value | 23 |
| Instance | |
| Name | Instance |
| Description | Instance attribute of EPath |
| Value | 01 |
| Attribute | |
| Name | Attribute |
| Description | Attribute attribute of EPath |
| Value | 24 |
| Service | |
| Name | Service |
| Description | Service attribute of EPath |
| Value | |

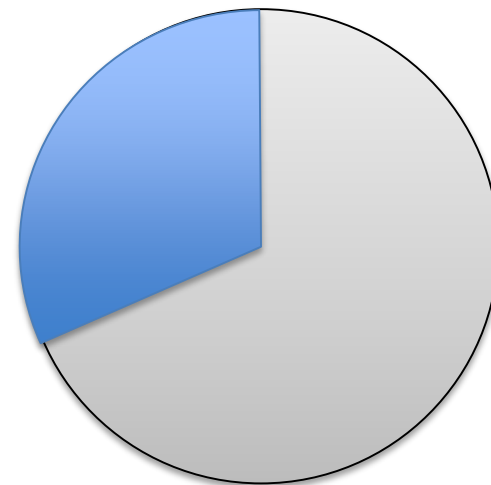
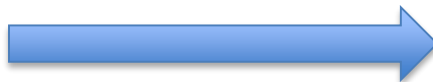


```

<InstanceHierarchy Name="ExampleIH">
  <InternalElement Name="AutomationMLObject" ID="de88f618-6802-40cc-9dd3-2d73546e6280">
    <Attribute Name="length" AttributeDataType="xs:integer" Unit="mm">
      <Description>Attribute with an eCl@ss RefSemantic</Description>
      <DefaultValue>5</DefaultValue>
      <Value>5</Value>
      <RefSemantic CorrespondingAttributePath="ECLASS:0173-1#02-BAA018#004" />
    </Attribute>
    <Attribute Name="minSize" AttributeDataType="xs:string" Unit="mm">
      <Description>Attribute with an eCl@ss RefSemantic</Description>
      <DefaultValue>10</DefaultValue>
      <Value>10</Value>
      <RefSemantic CorrespondingAttributePath="ECLASS:0173-1#02-BAE496#005" />
    </Attribute>
  </InternalElement>
</InstanceHierarchy>
  
```

Summary

- *neutral data format*
- *system models / engineering of automation technology*
- *fault free creation of engineering artifacts*
- *modelling industrial communications*
- *basic semantic libraries*
- *decrease of planning costs*





OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

MB

FACULTY OF
MECHANICAL ENGINEERING

THANK YOU