

Christian Doppler Laboratory

Software Engineering Integration
For Flexible Automation Systems

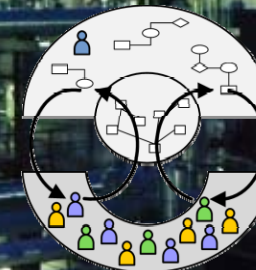
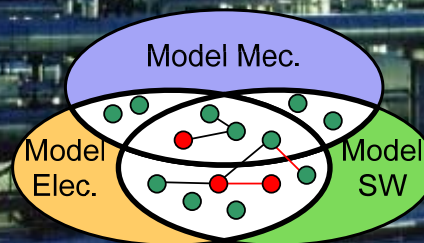
AutomationML Hub Round-trip Engineering in AutomationML Projects

Richard Mordinyi, Stefan Scheiber

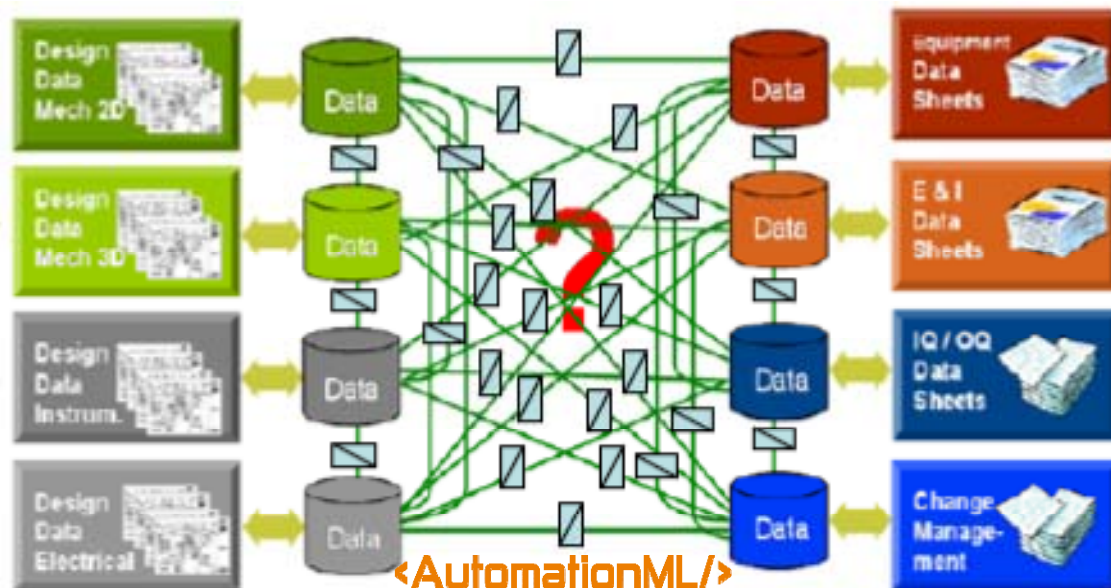
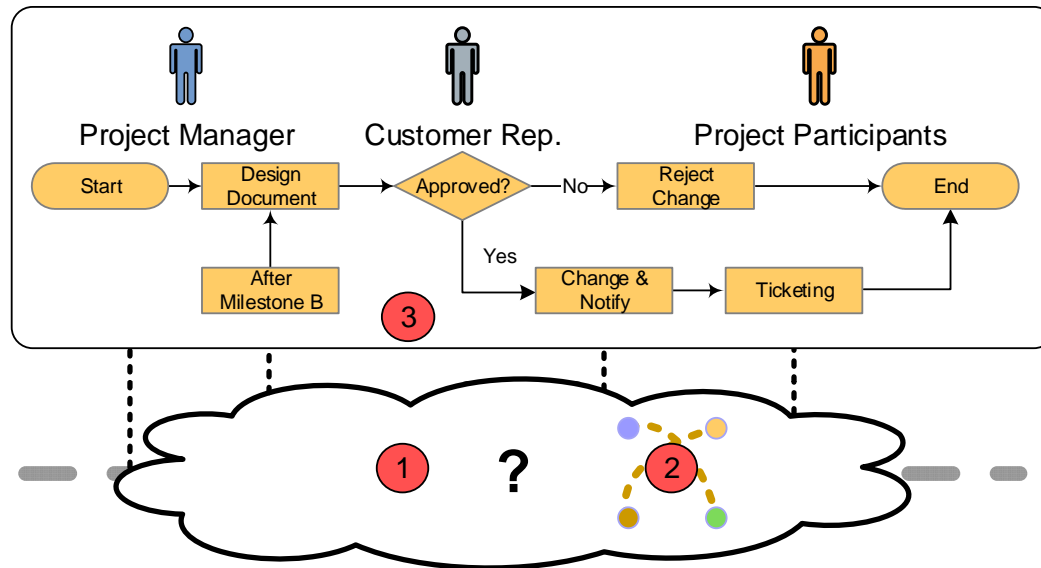
Institute of Software Technology
and Interactive Systems

Vienna University of Technology

<http://cdl.ifs.tuwien.ac.at>



Challenges in Engineering Tool Networks

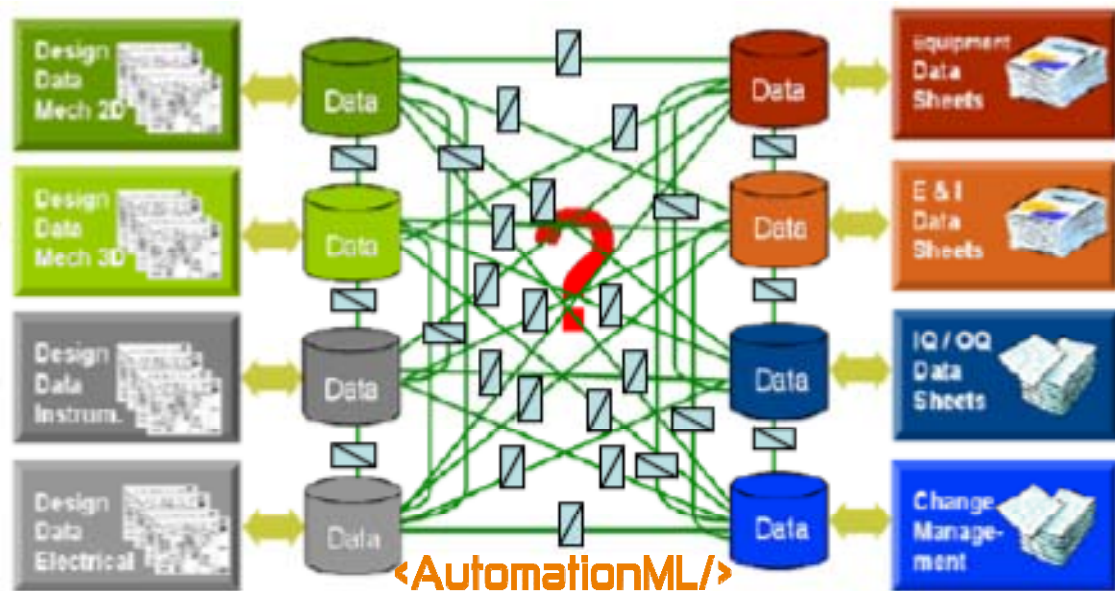
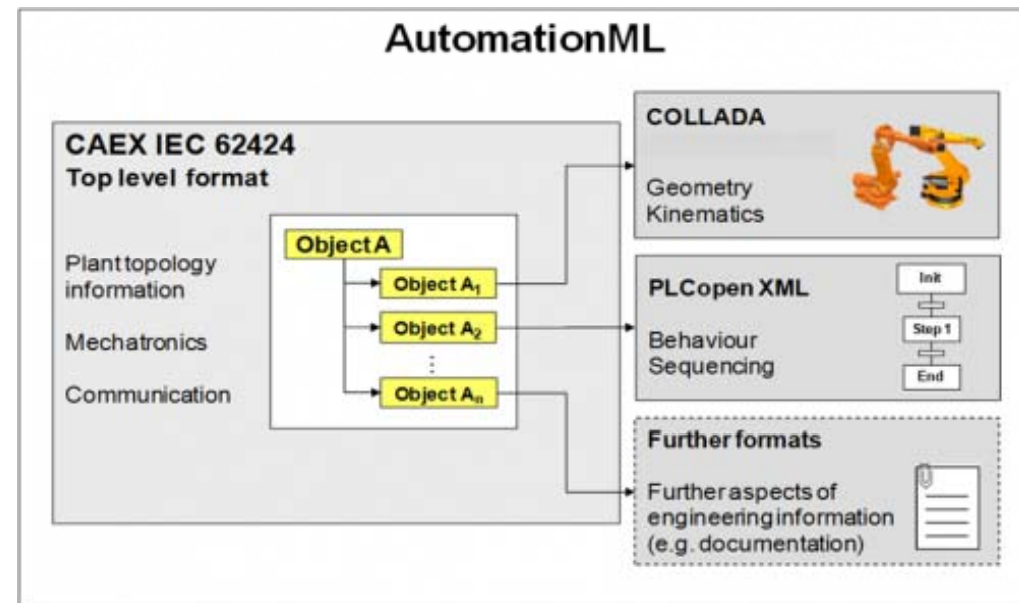


1. Tool networks do not ensure sufficient **data consistency**.
2. Data cannot be queried using an **uniform interface**.
3. Tool networks do not make data easily and efficiently **accessible on project level**, e.g., for risk management, test automation or observation of project progress.

Opportunities from AutomationML vs. Limitations Situation (as we see it)

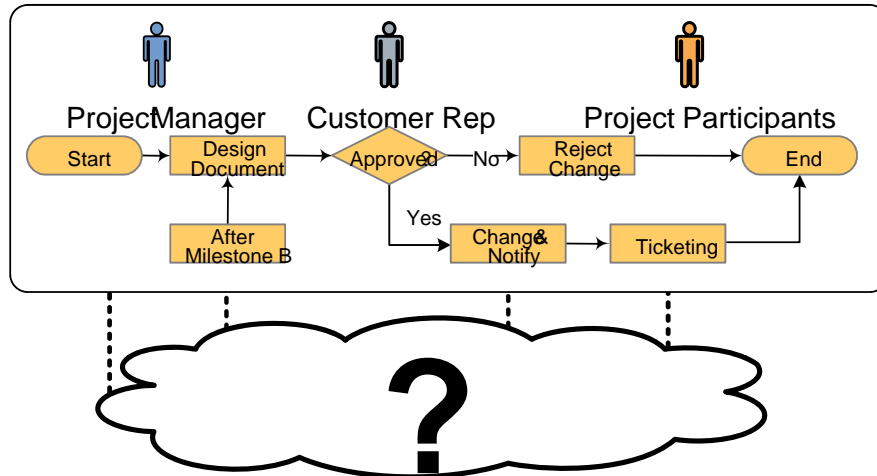


- AutomationML allows **engineering data exchange, modelling and AutomationML-based engineering**
- So engineering could be made more efficient ...



- **Handling, integration and management of AutomationML fragments and incorporated files is often unclear and not supported by SW-tools**

Model-related Challenges in AutomationML-Tool Networks



■ Traditional use of AutomationML for tool data exchange

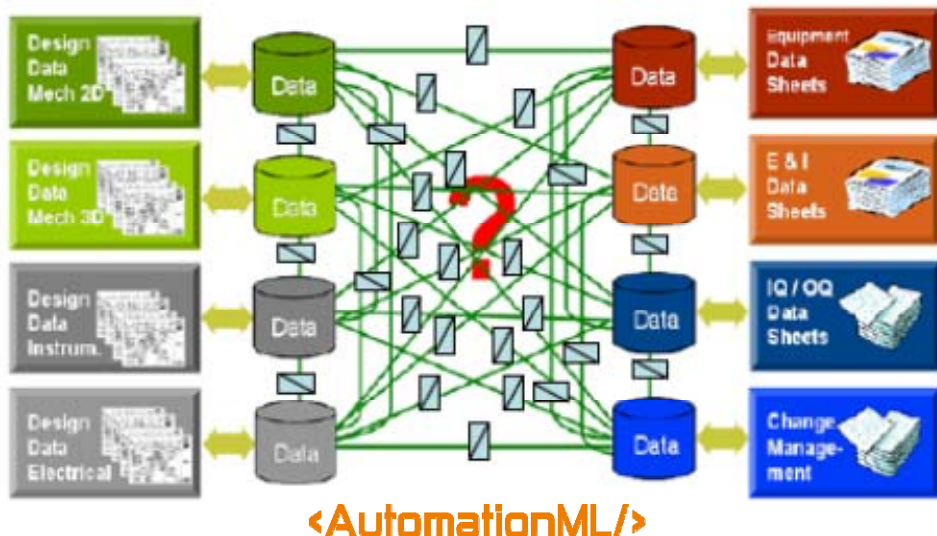
- An AutomationML file may represent a **state**, a **change**, the plant **topology**, details on a plant part.
- AutomationML files represent **fragments**, the project reference is – in general – not present in AutomationML.

■ Mapping is challenging ...

- Between AutomationML **fragments**.
- Between AutomationML fragments and **domain-specific views**.
- Between **plant topology** and domain-specific views.

■ Overview is difficult for

- **Domain experts**
- **Managers**

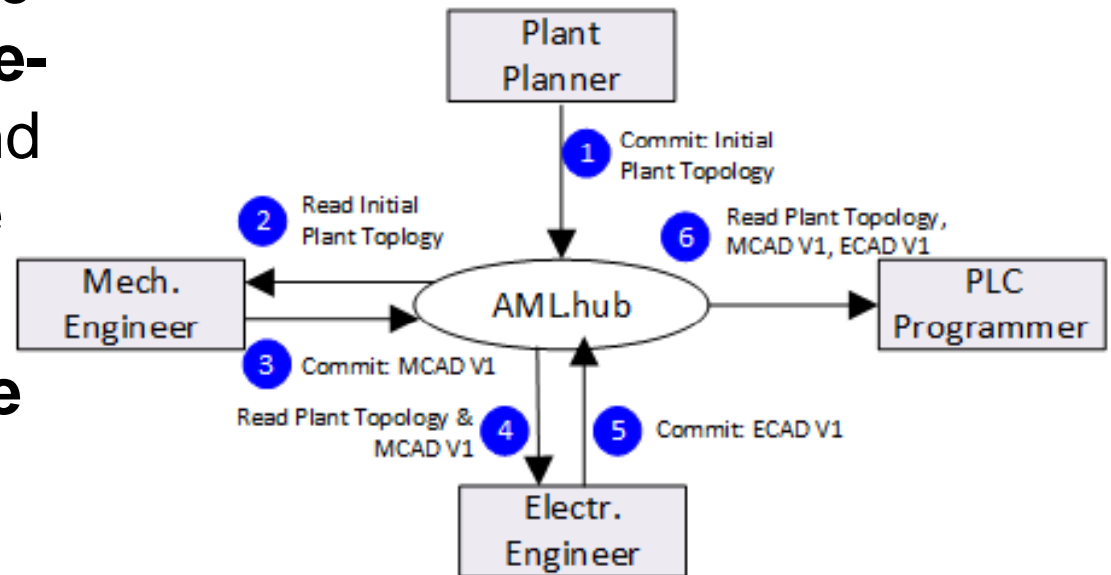


Needs to Benefit from the AutomationML Standard

Aspects of Round-Trip Engineering

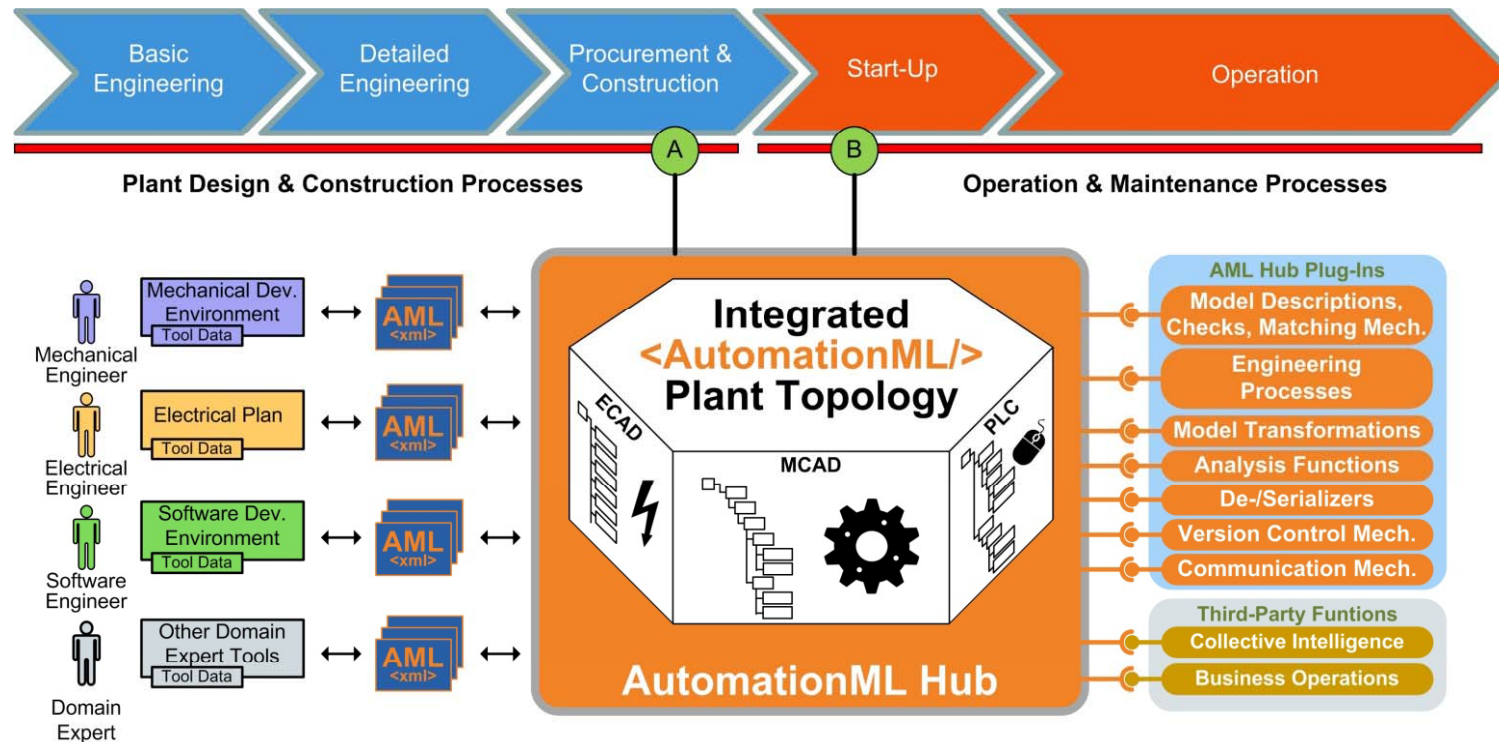


- Engineers would like to define and **maintain** their **discipline-specific topology tree** of and their tool-specific view on the automated system.
- **Practitioners** want to **assure quality** consistently, easily, and efficiently
- **Domain experts** want to **store** AutomationML data from tool chains and connected files of incorporated standards in **traceable versions**.



AutomationML Data Management based on AML Hub

Structure and Elements



- Open **integration and communication** platform
- **Versioned** AutomationML model repository for **plant topology** (AutomationML instance hierarchy) and class libraries
- **File-based version management** system.
- Allows **managing and reporting changes and consistency** of AutomationML-models and -libraries and connections to incorporated models.

AML Hub Demo Context

Production System at OvG University Magdeburg



■ Production System elements

- Conveyors
- Turn tables
- Machines



v0_00

Plant Planner



v0_01

•Base Topology



v0_02



v0_10

Mechanical Engineer



v0_11

•VDMA66415InterfaceClassLib



v0_12

•VDMA66415RoleClassLib



v0_13

•VDMA66415SUCLib



v0_14

•2D Layout



v0_20

Electrical Engineer



v0_21

•CommunicationCLib



v0_22

•CommunicationRCLib



v0_23

•added LogicInterface,
VariableInterface



v0_24

•Buskoppler



v0_25

•IO wiring



v0_26

•InternalLinks



v0_27



v0_30

Software Engineer



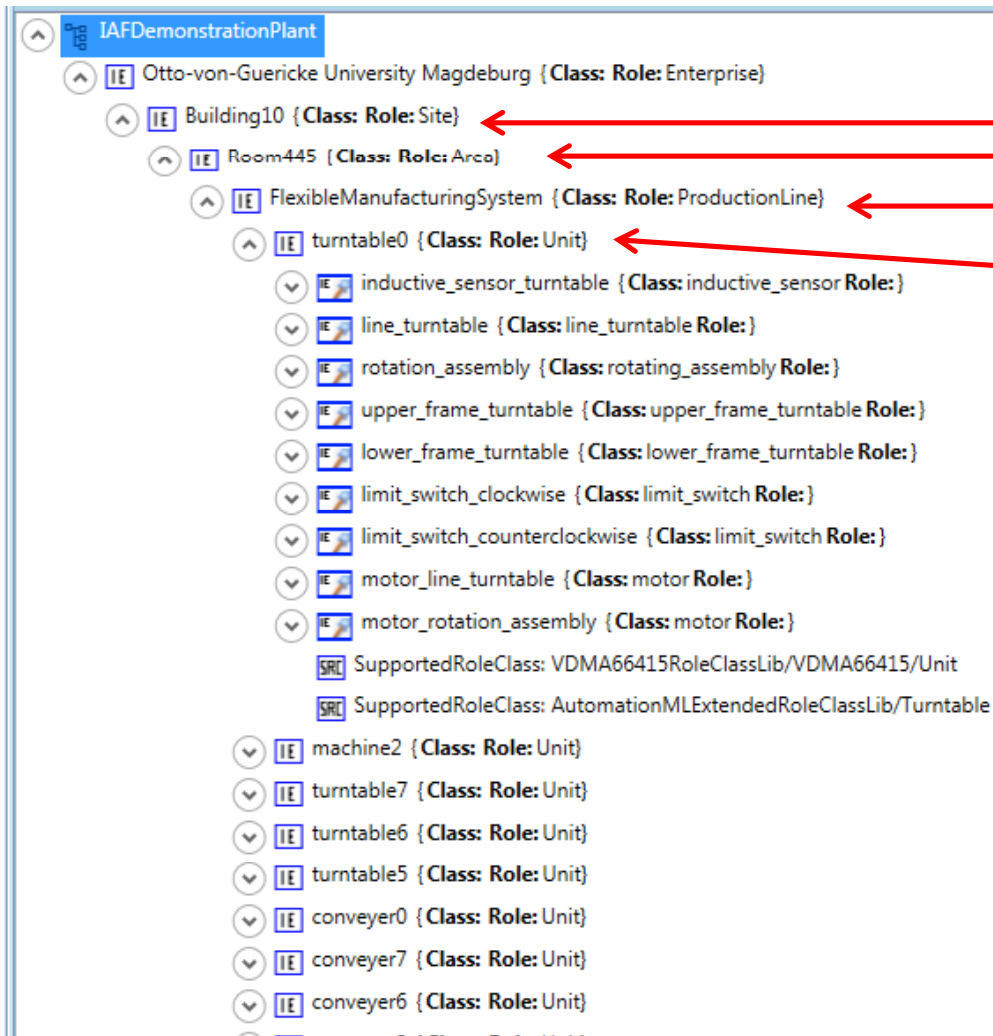
v0_31

* Variables

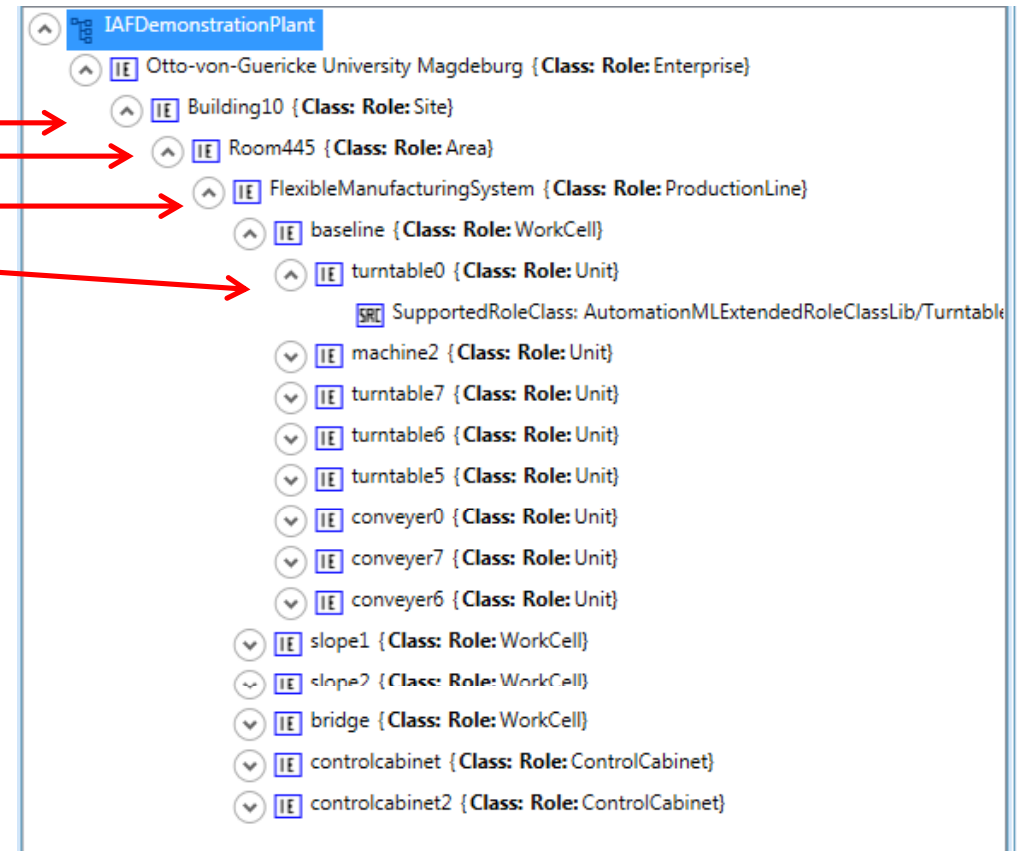
Mapping of Tree Topologies



MCAD



Integrated Plant Topology

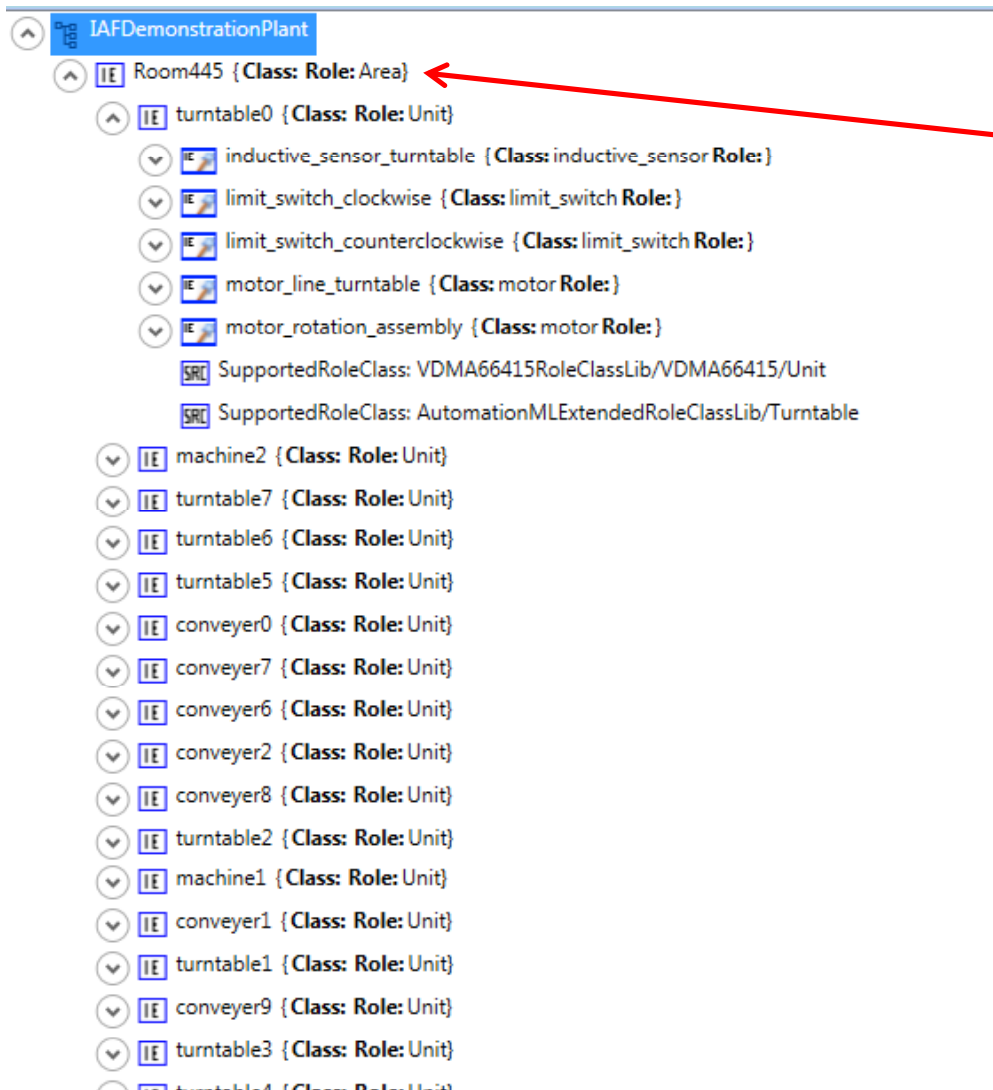


- Focus on mechanical hierarchy
- Neglecting organizational structures

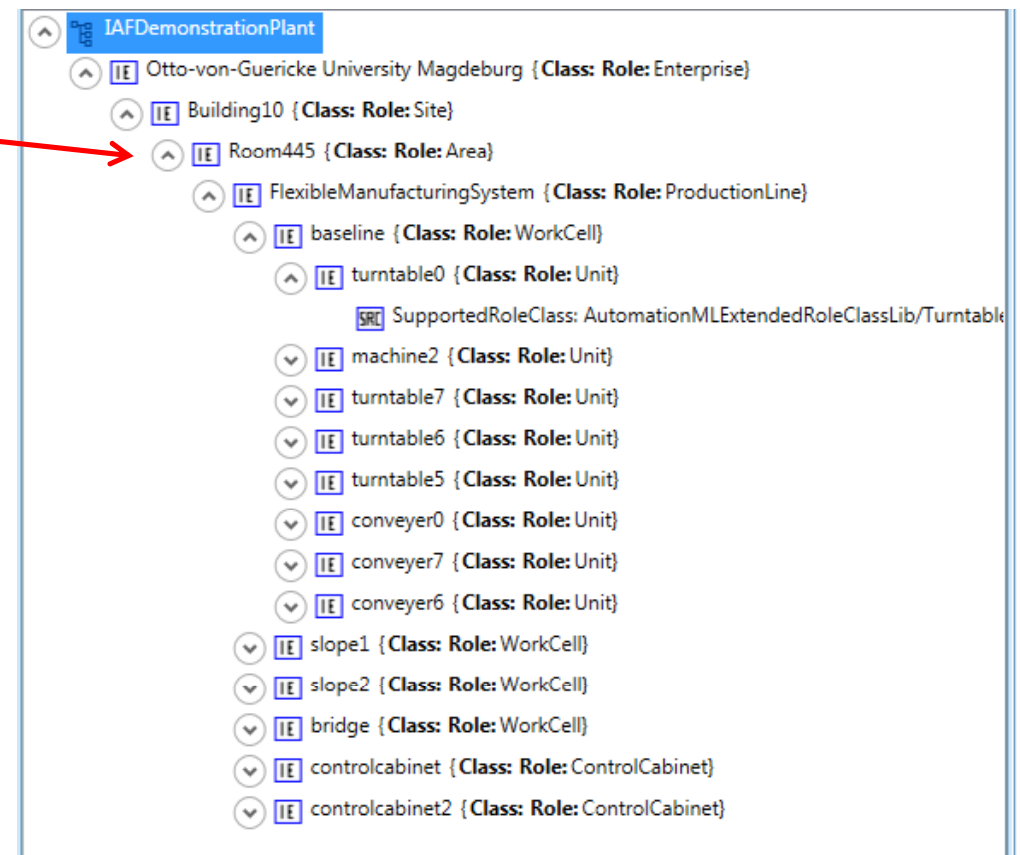
Mapping of Tree Topologies



ECAD



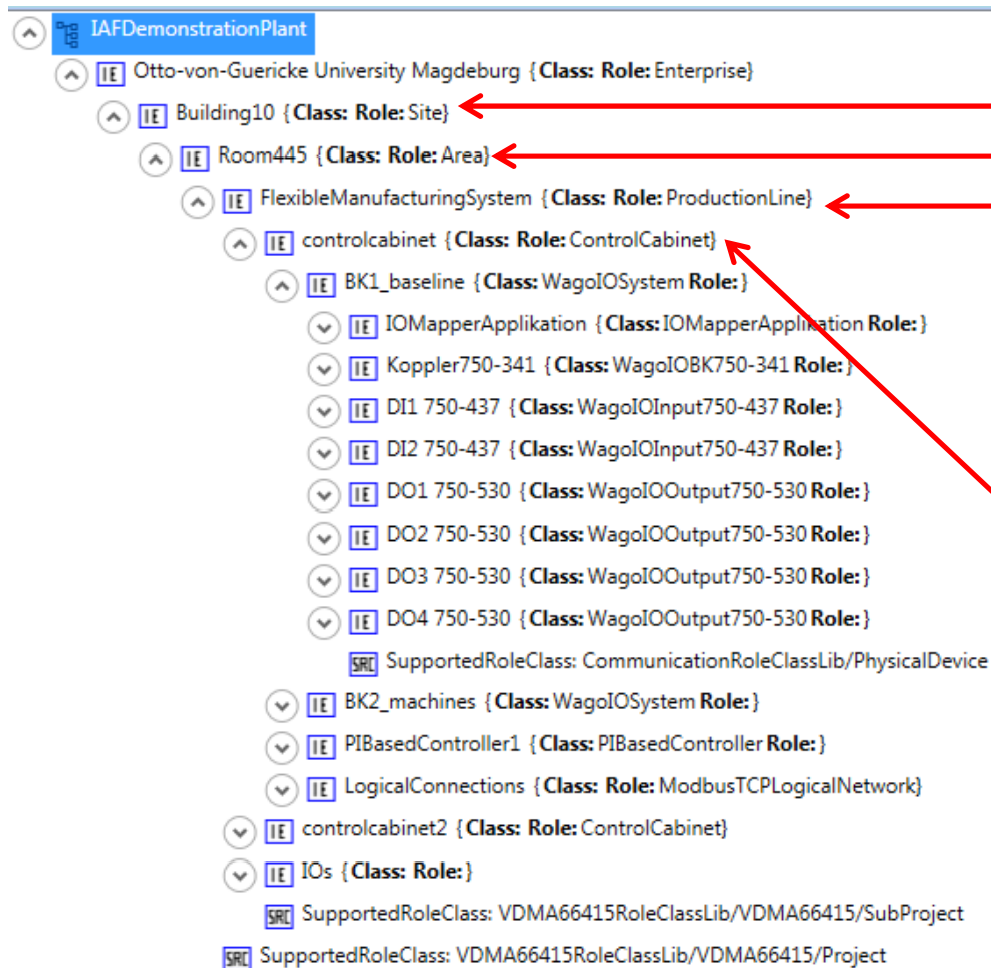
Integrated Plant Topology



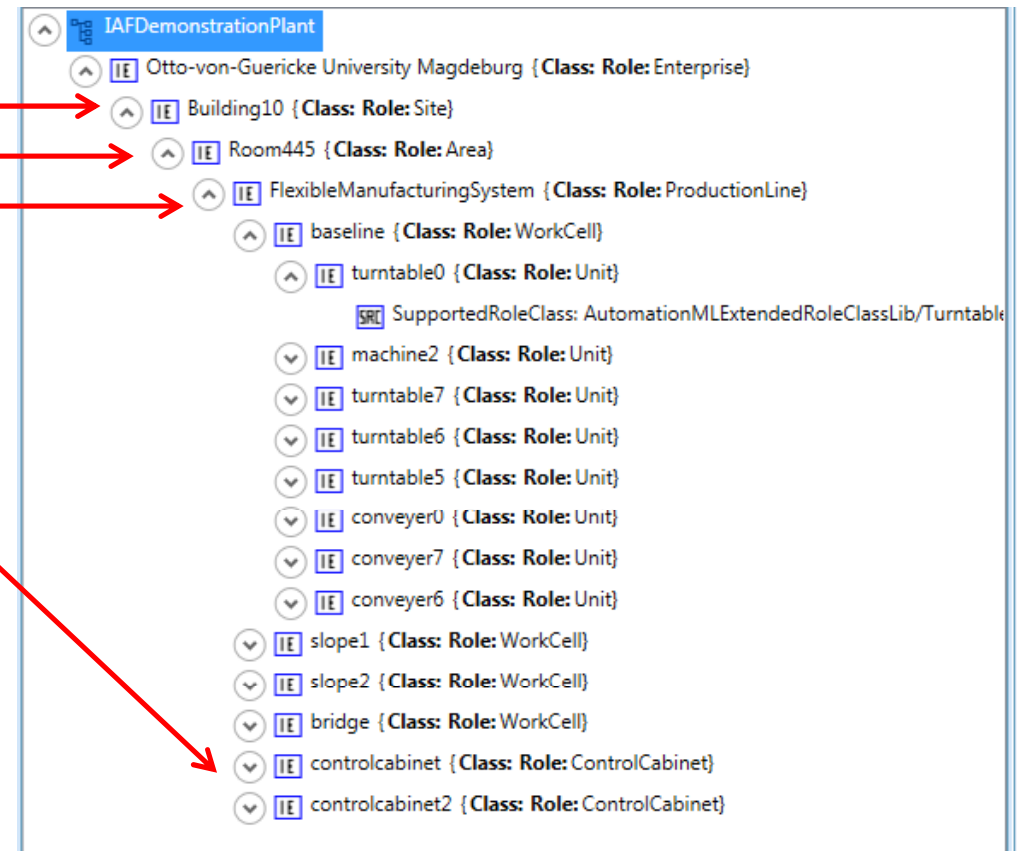
- Focus on device list for wiring

Mapping of Tree Topologies

PLC



Integrated Plant Topology



- Need for device clusters (control cabinet) and device lists



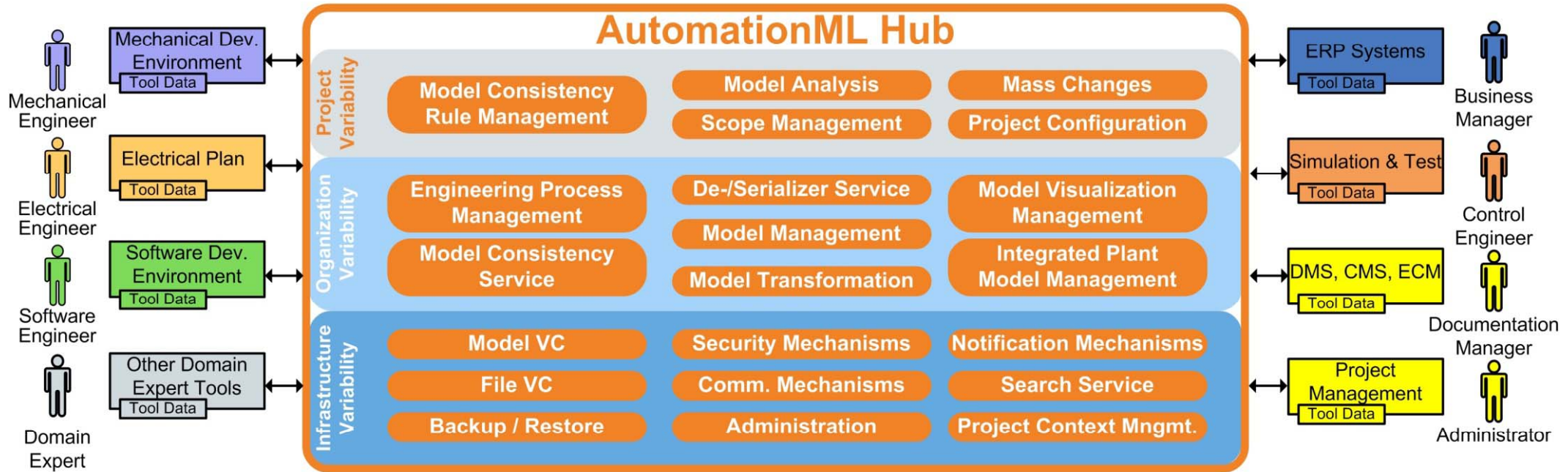
Simple Snippets and Configurations

API



AutomationML Hub

Variability of Components and Services



- Infrastructure Variability
- Organization Variability
 - General systems engineering processes and phases
 - Definition of responsibilities
- Project Variability
 - Domain Knowledge
 - Definition of Views on Plant Topology



Elements in various stages of development

Integrated Plant Model Management

Model Matching – Identifies differences between model versions

- EMF Compare API
- Implement a custom class to compare instances
 - Includes a `Function<EObject, String>` that selects the attributes for comparison
- Build the configuration for EMFCompare
 - Define fallback strategy and matching engines
- Write a method that
 - creates a scope
 - compares with the customized comparator

```
class MyCompare {
    private EMFCompare comparator;

    public MyCompare() {
        // Comparison function for AML models
        Function<EObject, String> idFunction = new Function<EObject, String>() {
            public String apply(EObject input) {
                // Compare by ID of CAEXObject
                return ((CAEXObject) input).getID();
            }
        };

        IEObjectMatcher myMatcher = new IdentifierEObjectMatcher(fallBackMatcher, idFunction);
        IMatchEngine myMatchEngine = new DefaultMatchEngine(myMatcher, comparisonFactory);
        IMatchEngine.Factory engineFactory = new MatchEngineFactoryImpl() {
            public IMatchEngine getMatchEngine() {
                return myMatchEngine;
            }
        };
        matchEngineFactoryRegistry.add(engineFactory);
        builder.setMatchEngineFactoryRegistry(matchEngineFactoryRegistry);
        comparator = builder.build();
    }

    public Comparison compare(Notifier left, Notifier right) {
        // Creating the comparison scope
        IComparisonScope scope = new DefaultComparisonScope(left, right, right);
        // Configuring the comparison
        return comparator.compare(scope);
    }
}

public void compareModels(EObject leftModel, EObject rightModel, Parameters... parameters) {
    DefaultAmlMerger merger = new DefaultAmlMerger();
    // Custom class for comparison of models
    MyCompare compare = new MyCompare();

    Comparison comparison = compare.compare(leftModel, rightModel);
    merger.mergeLeftToRight(comparison);
}
```



De-/Serialization

Serializer – Exports AML model into AML file

```
public interface Serializer<T> {
    /**
     * Returns the data format this Serializer writes into.
     *
     * @return a SerializationFormat
     */
    SerializationFormat getOutputFormat();

    /**
     * Serializes the given Java object graph into the given OutputStream.
     *
     * @param model the object graph to serialize
     * @param out the OutputStream to write to
     * @throws Se the exception if serialization fails
     */
    void serialize(T model, OutputStream out) throws SerializationException;
}

public class ResourceXmlSerializer implements Serializer<Resource> {
    @Override
    public SerializationFormat getOutputFormat() {
        return SerializationFormat.XML;
    }

    @Override
    public void serialize(Resource model, OutputStream outputStream) throws SerializationException {
        XMLResource xml = asXmlResource(model);

        Map<Object, Object> options = new HashMap<>();
        options.put(XMLResource.OPTION_EXTENDED_META_DATA, Boolean.TRUE);

        try {
            xml.save(outputStream, options);
        } catch (IOException e) {
            throw new SerializationException("Error saving Resource", e);
        }
    }

    private XMLResource asXmlResource(Resource resource) {
        return new ExportResource(resource);
    }
}
```



De-/Serialization

Deserializer – Imports AML file as AML model

```
public interface Deserializer<T> {
    /**
     * Returns the data format this Deserializer reads from.
     *
     * @return a SerializationFormat
     */
    SerializationFormat getInputFormat();

    /**
     * Deserializes the given InputStream into a Java object graph.
     *
     * @param stream the data stream to deserialize
     * @return
     * @throws
     */
    T deserialize();
}

public class ResourceXmlDeserializer implements Deserializer<Resource> {
    @Override
    public SerializationFormat getInputFormat() {
        return SerializationFormat.XML;
    }

    @Override
    public Resource deserialize(InputStream in) throws SerializationException {
        XMLProcessor processor = getXmlProcessor();

        Map<Object, Object> options = new HashMap<>();

        try {
            return processor.load(in, options);
        } catch (IOException e) {
            throw new SerializationException("Error deserializing resource", e);
        }
    }
}
```



Model Management

Beyond AML - Loading of Additional Model Descriptions

```
public class InformedResourceXmlDeserializer extends ResourceXmlDeserializer {

    private final EPackage ePackage;

    private final EPackage.Registry ePackageRegistry;

    public InformedResourceXmlDeserializer(EPackage ePackage) {
        this.ePackage = ePackage;
        this.ePackageRegistry = new EPackageRegistryImpl(EPackage.Registry.INSTANCE);
        this.ePackageRegistry.put(null, ePackage);
    }

    @Override
    protected XMLProcessor getXmlProcessor() {
        return new LocalXmlProcessor();
    }

    private class LocalXmlProcessor extends XMLProcessor {
        public LocalXmlProcessor() {
            super(ePackageRegistry);
            extendedMetaData.putPackage(null, ePackage);
        }

        @Override
        protected Map<String, Resource.Factory> getRegistrations() {
            if (registrations == null) {
                super.getRegistrations();
                registrations.put(XML_EXTENSION, new LocalResourceFactory(ePackageRegistry, ePackage));
            }
            return registrations;
        }
    }
}
```




Scope Management

Definition of Scopes

- Scope
 - Defines view on the Integrated Plant Model
 - Set of RoleClasses
- Scope Management
 - Scopes are seen as part of a Role with certain permissions

```
karaf@openengsb(> user-create mengineer password
karaf@openengsb(> project-create IAFDemonstrationPlant
karaf@openengsb(> permission-create "scope:AutomationMLBaseRoleClassLib/AutomationMLBaseRole"
karaf@openengsb(> role-create mengineer-role "scope:AutomationMLBaseRoleClassLib/AutomationMLBaseRole"
karaf@openengsb(> █
```



Early Preview Version for the SPS/IPC/Drives 2015

Halle 6 Stand 331 (logi.cals)

Halle 4 Stand 548 (TU Wien)

DEMO

Screenshots

Project-Context Selection



AML.hub Home Check-in Object Browser Compare commits Checkout My profile Log out

Profile

Summary

Details
Username: plantplaner
Default Project: IAFDemonstrationPlant

Assignments
IAFDemonstrationPlant plantplaner-role

- Selection of current project context
- Selection of default-role of user in that project
 - Implicitly selects Scope

Screenshots

Check-in Process

AML.hub Home Check-in Object Browser Compare commits Checkout

Check-in

Checkin

File input

Browse... v0_11.zip

Please select a file to check in.

Project

IAFDemonstrationPlant

Branch

master

Commit message

v0_11

Submit

- Selection of
 - AutomationML file only
 - Zip file containing both AML file and related files
- Provide commit message to keep others informed

Checkin

Succesfully checked in 1 models and 27 files.

Diff view

Diff Kind	Diff Location	Element Type / Attribute Name	Element / Diff Descriptor
ADD	CAEXFile / IAF_ModelSystemUnitClassLib / parts / line_turntable / line_Turntable_COLLADAInterface	Attribute	refURI = File:///geometry/belt_turntable.dae#ID1
ADD	CAEXFile / IAF_ModelSystemUnitClassLib / parts / line_turntable / line_PLCopOpen_LogicInterface	Attribute	refURI = File:///sequence /line_conveyer.xml#ISID_20131218-600
ADD	CAEXFile / IAF_ModelSystemUnitClassLib / parts / line_turntable / geometry	Description	(LxWxH)
ADD	CAEXFile / IAF_ModelSystemUnitClassLib / parts / line_turntable / geometry	Attribute	length = 160
ADD	CAEXFile / IAF_ModelSystemUnitClassLib / parts / line_turntable / geometry	Attribute	width = 20
ADD	CAEXFile / IAF_ModelSystemUnitClassLib / parts /	Attribute	height = 20

- Difference Report: changes on Integrated Plant Model

Screenshots

AML Browser



You are viewing revision 1444635843232 by mengineer as plant-documenter

Instance hierarchy

- Otto-von-Guericke University Magdeburg
 - Building10
 - Room445
 - FlexibleManufacturingSystem
 - baseline
 - turntable0
 - inductive_sensor_turntable**
 - line_turntable
 - rotation_assembly
 - upper_frame_turntable
 - lower_frame_turntable
 - limit_switch_clockwise
 - limit_switch_counterclockwise
 - motor_line_turntable
 - motor_rotation_assembly
 - machine2
 - turntable7
 - turntable6
 - turntable5
 - conveyer0
 - conveyer7
 - conveyer6
 - slope1
 - slope2
 - bridge
 - controlcabinet

Role Classes

- IAF_ModelRoleClassLib/sensor
- VDMA66415RoleClassLib/VDMA66415/Device

Attributes

Name	Unit	Type	Value
port		xs:string	
input- / output voltage	V	xs:integer	30
circuit time	ms	xs:integer	1
sensitivity	mT	xs:decimal	0.7
weight	g	xs:integer	54
material		xs:string	
geometry		xs:string	
length	mm	xs:integer	15
width	mm	xs:integer	10
height	mm	xs:integer	10
range	mm	xs:integer	70

- Browsing Plant Topology
- Visualization of Scope
- Inspecting details of selected AML element

AutomationML Data Management based on AML.hub

Benefits for User Organization



✓ Support for efficient quality management

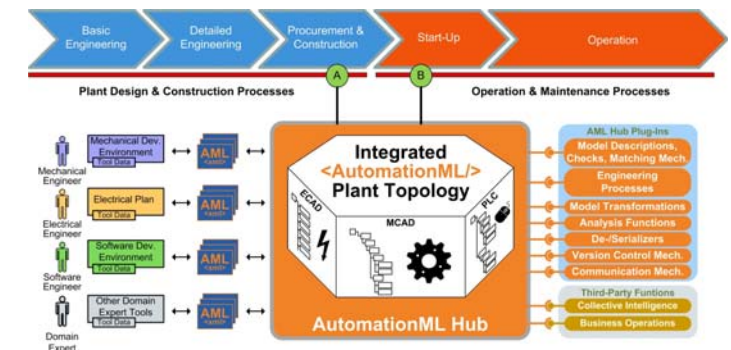
- Individual tool-specific topology view
- Automated view-consistency via Integrated Plant Topology
- Changes in **cross-discipline context** are made visible

✓ Overview on versioned data

- Files and AutomationML data elements
- Instance hierarchy
- Optional: COLLADA data elements and PLCopen data elements

✓ Better coordination in heterogeneous engineering projects

- Traceable engineering process



We are looking forward to ...



- ... work with you on AML-related engineering project data in the AML Hub context.
- ... discuss with you AutomationML-based engineering processes to identify gaps and improvement opportunities with the AML Hub.

Richard Mordinyi, Stefan Scheiber,
(richard.mordinyi|stefan.biffl)@tuwien.ac.at

Heinrich Steininger,
CEO logi.cals Austria
heinrich.steininger@logicals.com
<http://www.logicals.com>

Prof. Dr. Stefan Biffl,
Head of Ch. Doppler Laboratory
stefan.biffl@tuwien.ac.at
<http://cdl.ifs.tuwien.ac.at>