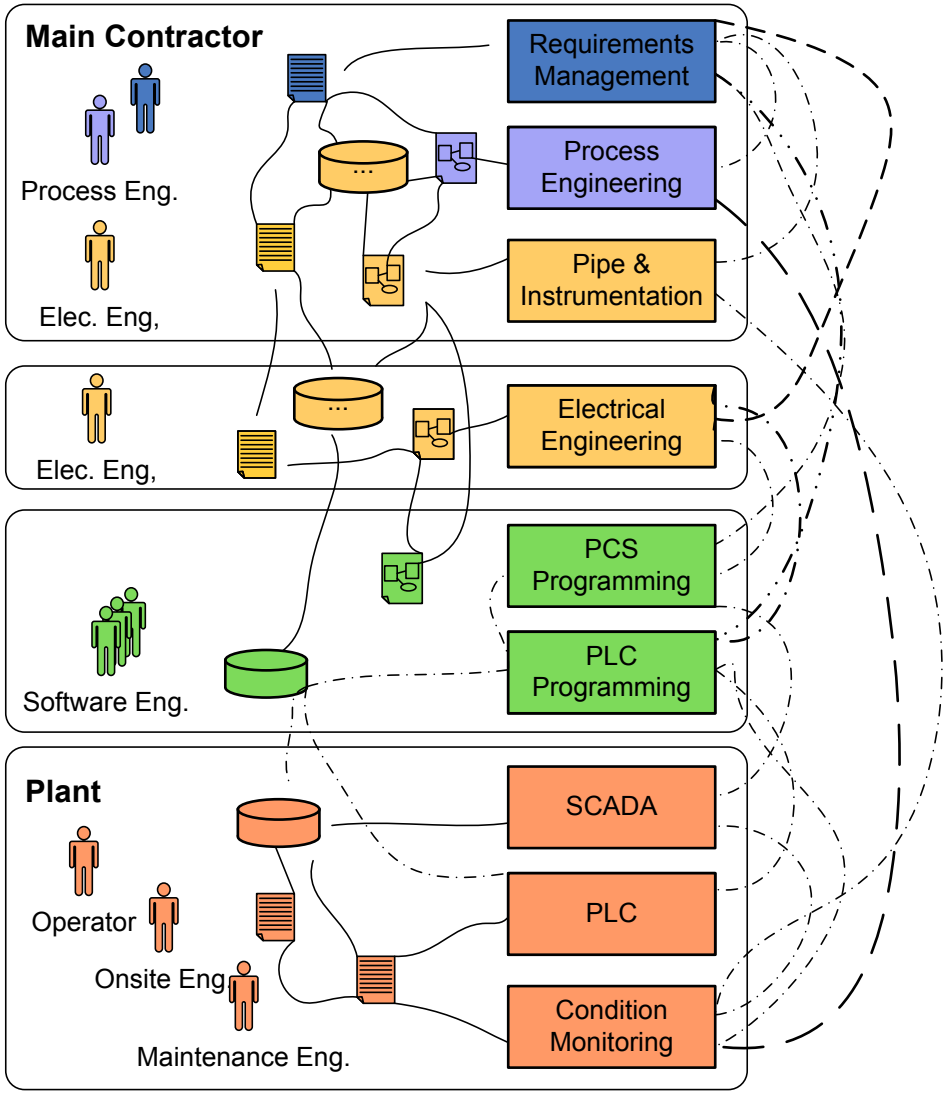
Semantic Mapping Support for Mechatronic Objects in AutomationML

Stefan Biffl Olga Kovalenko Arndt Lüder Nicole Schmidt Ronald Rosendahl
Christian Doppler Laboratory
"Software Engineering Integration for Flexible Automation Systems"
Vienna University of Technology

Faculty Mechanical Engineering, Otto-v.-Guericke University, Magdeburg, Germany

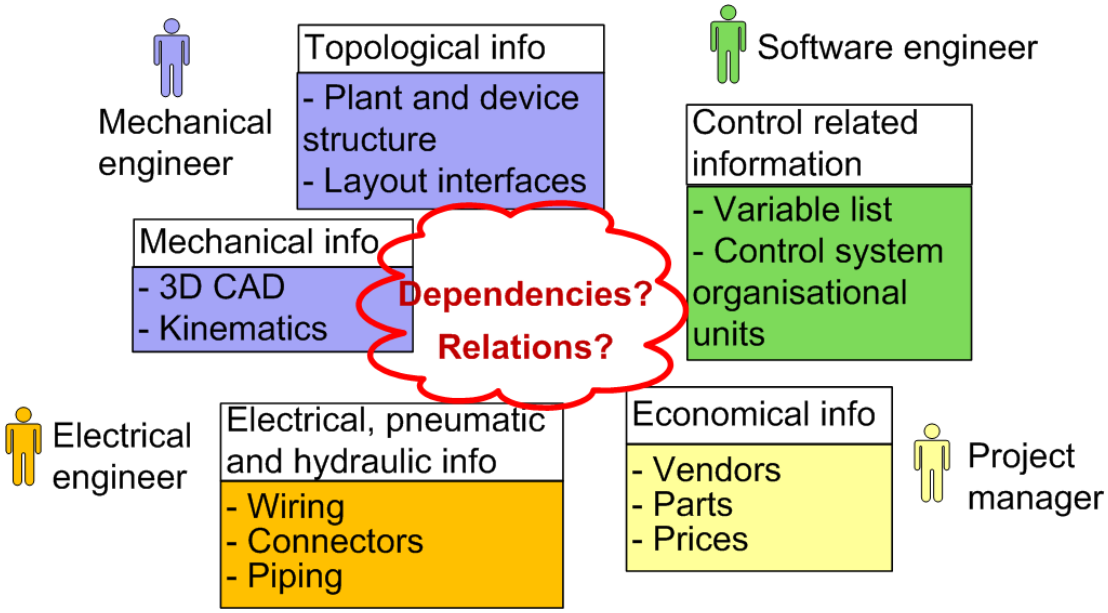
Production Systems Engineering Environments



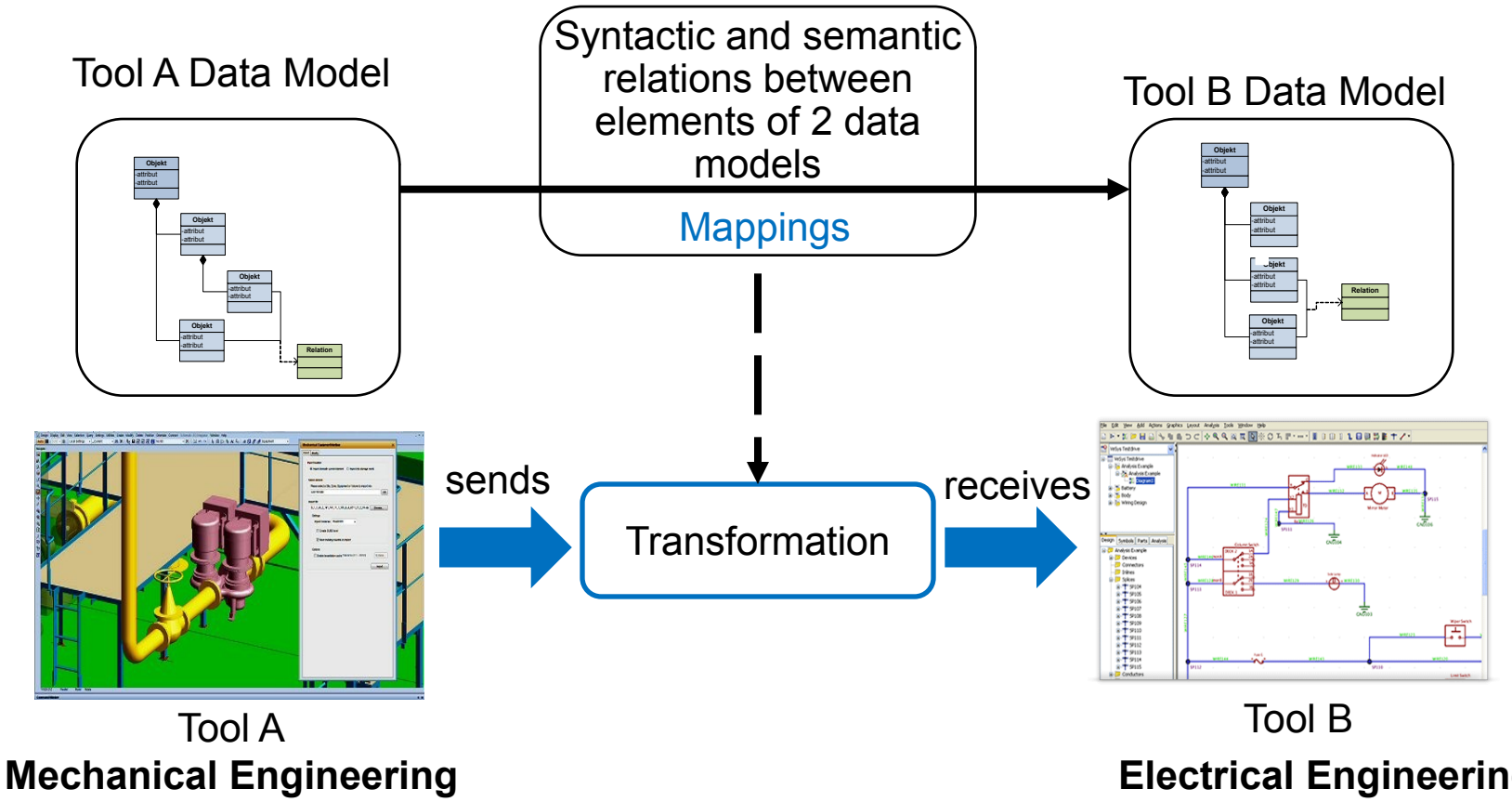
- **Multi-disciplinary engineering**
  - Cooperation of different engineering disciplines
- **Various Heterogeneities**
  - Technical: different eng. tools
  - Semantic: dissimilar data models and data formats
  - Process: tailored engineering processes
- **Needs**
  - Data exchange and data analysis across discipline and tool boundaries
  - Explicit specification of dependencies and relations between heterogeneous data models

Information Exchange in Production Systems Engineering

- Machine-understandable definition of relations between engineering views
  - E.g., to automate consistency checking, change management
- No standardized means to represent relations and dependencies
- **Goal**
  - Investigate capabilities of AutomationML to represent semantic mappings, i.e. Relations and dependencies between eng. models



Interaction Scenario across Discipline and Tool Boundaries in Mechatronic Environments



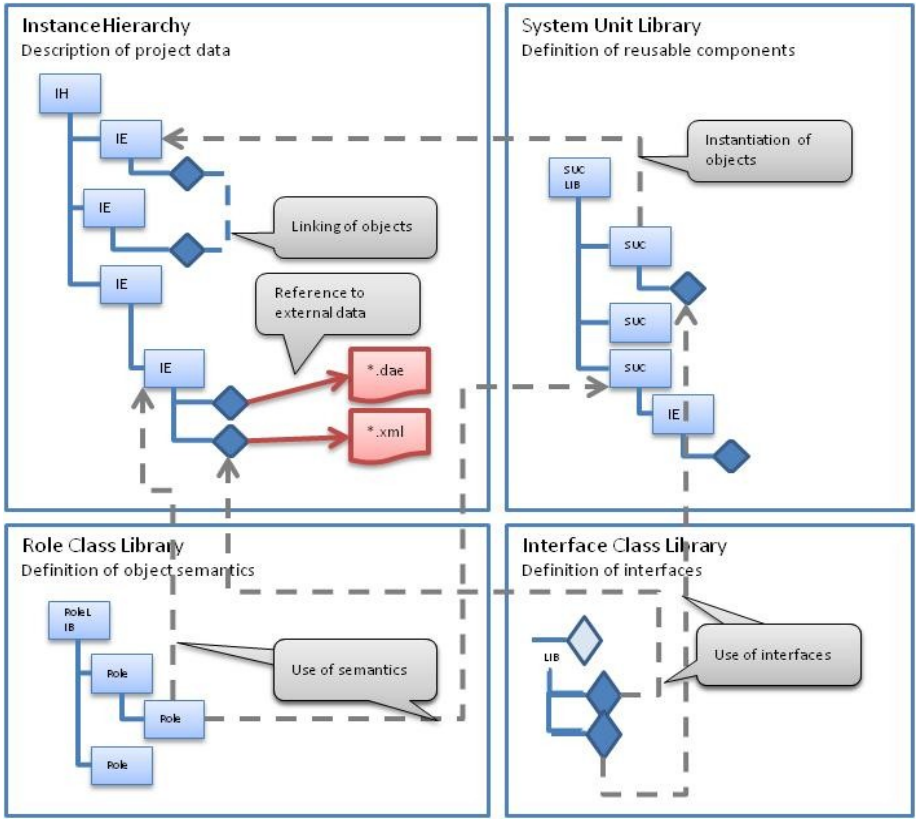
Semantic Mapping Types

- ***Mapping*** – captures the semantic and structural relationships between the entities of two data models, i.e.
  - Formally and explicitly defined
  - In enough detail to allow data interpretation

- **Identified mapping types**:
  - M1: Value processing
    - M1.1: String processing
    - M1.2: Data type transformation
    - M1.3: Math functions
    - M1.4: External functions call
  - M2: Granularity
  - M3: Schematic differences
  - M4: Conditional mappings
  - M5: Bidirectional mappings
  - M6: Grouping and aggregation
  - M7: Restrictions on values

Modeling Mappings in AutomationML

- **Major modeling means**
  - *Role Class Libraries* modeling object semantics
  - *Interface Class Libraries* modeling object relations and references to external information
  - *System Unit Class Libraries* modeling production system component libraries
  - *Instance Hierarchies* with internal elements modeling the hierarchy of production system components in a recent project or setting



Modeling Mappings in AutomationML (cnd)

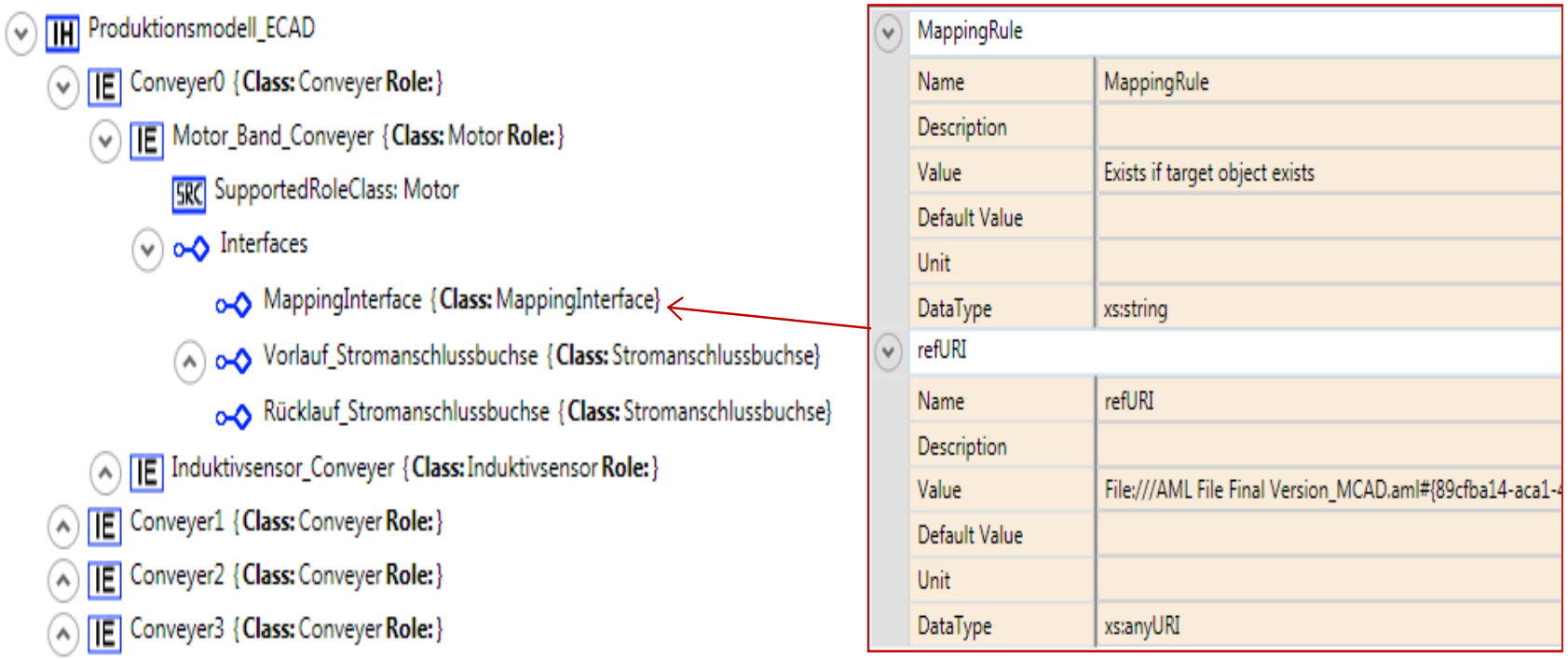
- **Modeling capabilities**
  - A1: Part – Subpart relation in InstanceHierarchy
  - A2: Mirror objects
  - A3: Interfaces
  - A4: Interface – InternalLink combination
  - A5: Interface - InternalLink – InternalElement combination
  - A6: Semantic reference attribute
  - A7: RefBaseSystemUnitClassPath attribute
  - A8: RoleRequirement and SupportedRoleClass Attribute

		Mappings									
		M1.1	M1.2	M1.3	M1.4	M2	M3	M4	M5	M6	M7
AutomationML	A1	-	-	-	-	+	-	-	-	+	-
	A2	-	-	-	-	+	+	-	-	-	-
	A3	+	+	+	+	+	+	+	+	-	+
	A4	-	-	-	-	-	-	-	+	-	-
	A5	+	+	+	-	-	-	+	+	-	+
	A6	+	+	-	-	+	+	-	-	-	-
	A7	-	-	-	-	+	-	-	-	-	-
	A8	-	-	-	-	+	+	-	-	-	-

Different modeling possibilities have different capabilities for mapping representation

AML Example 1: Mappings crossing File Borders

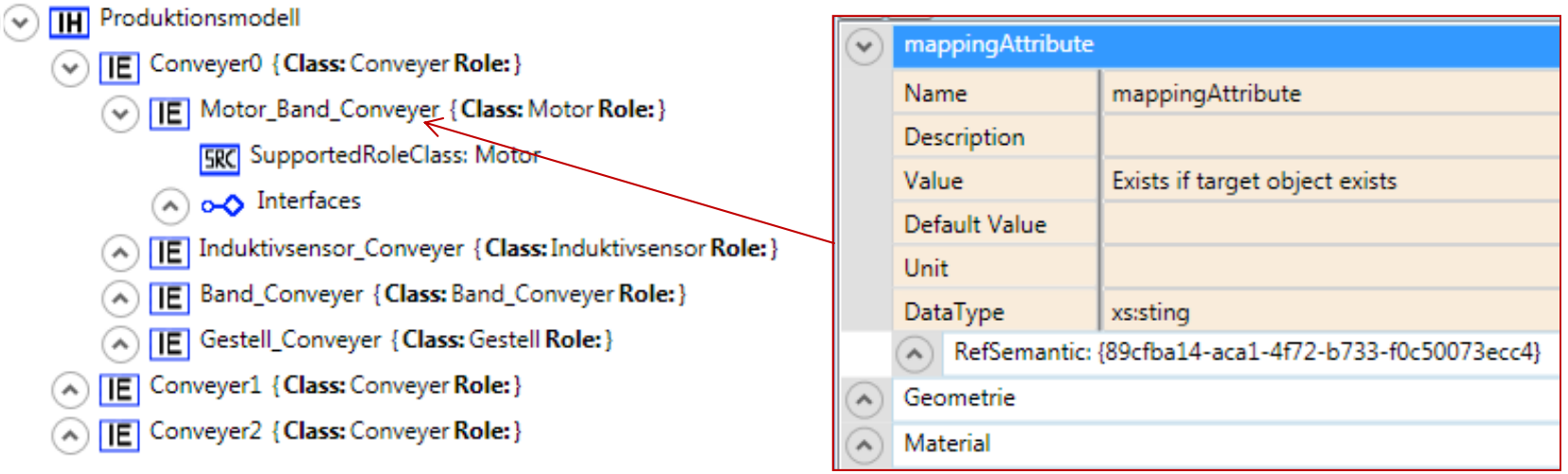
- Different AutomationML files for the mechanical, the electrical and the control engineering parts
  - Only the external interfaces (Modelling type A3) is applicable



AML Example 2: Mappings crossing Domain-Specific Hierarchy Borders

- One CAEX file
- Two or more instance hierarchies for different engineering disciplines

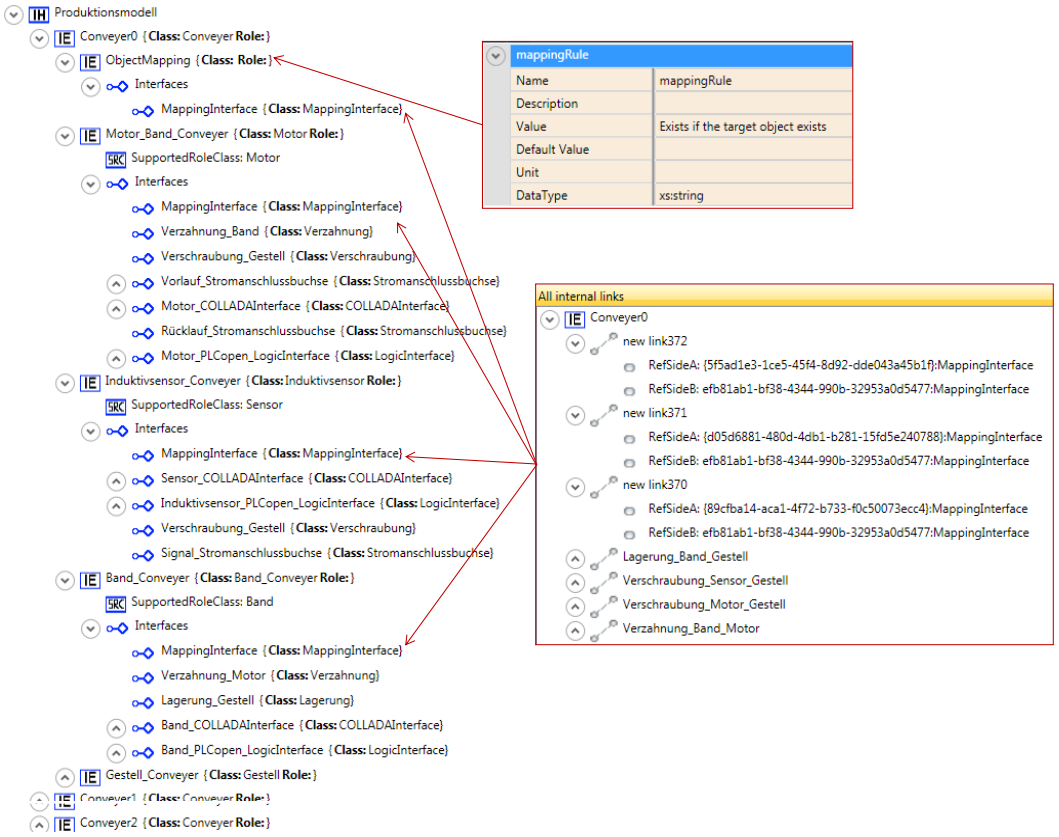
- RefSemantic within attributes (Modelling type A6) is best applicable in this case
  - Can carry an identifier enabling the identification of other objects
  - Attribute hosting the reference can carry the description of the mapping



AML Example 3: Mappings within one Instance Hierarchy

- One CEAX file with one instance hierarchy to model all relevant engineering data

- All modeling types can be applied, except for
  - External interfaces (modelling type A3)
  - Common semantic representations (modelling types A7 and A8)



Summary

- We investigated how semantic mappings between engineering models can be represented in the data exchange standard AutomationML.
- All required mapping types can be represented in AutomationML.
- But there is no clear guideline in the AutomationML context on how to represent relations and dependencies between model views.
- Such a guideline depends on the application case, i.e. whether
  - the mapping connecting objects of different files or in the same file, and
  - different instance hierarchies or
  - the same hierarchy.
- Future work will investigate guidelines for engineers and tools on best practices for representing links between engineering models extending current practice.