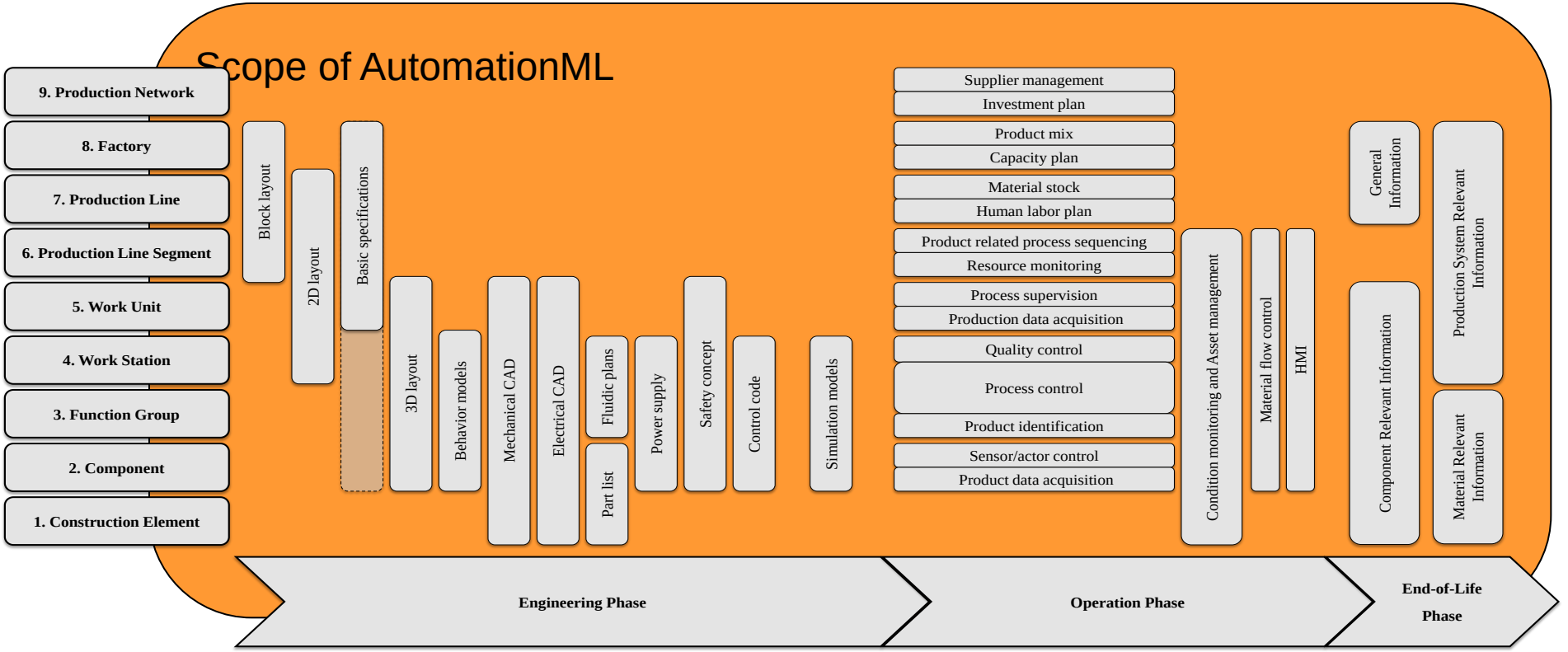
Designing tool chains based on common concepts - a data integration approach – an advanced workshop

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Production systems are complex

- During the life cycle phases and the different layers of the system hierarchy different information are relevant

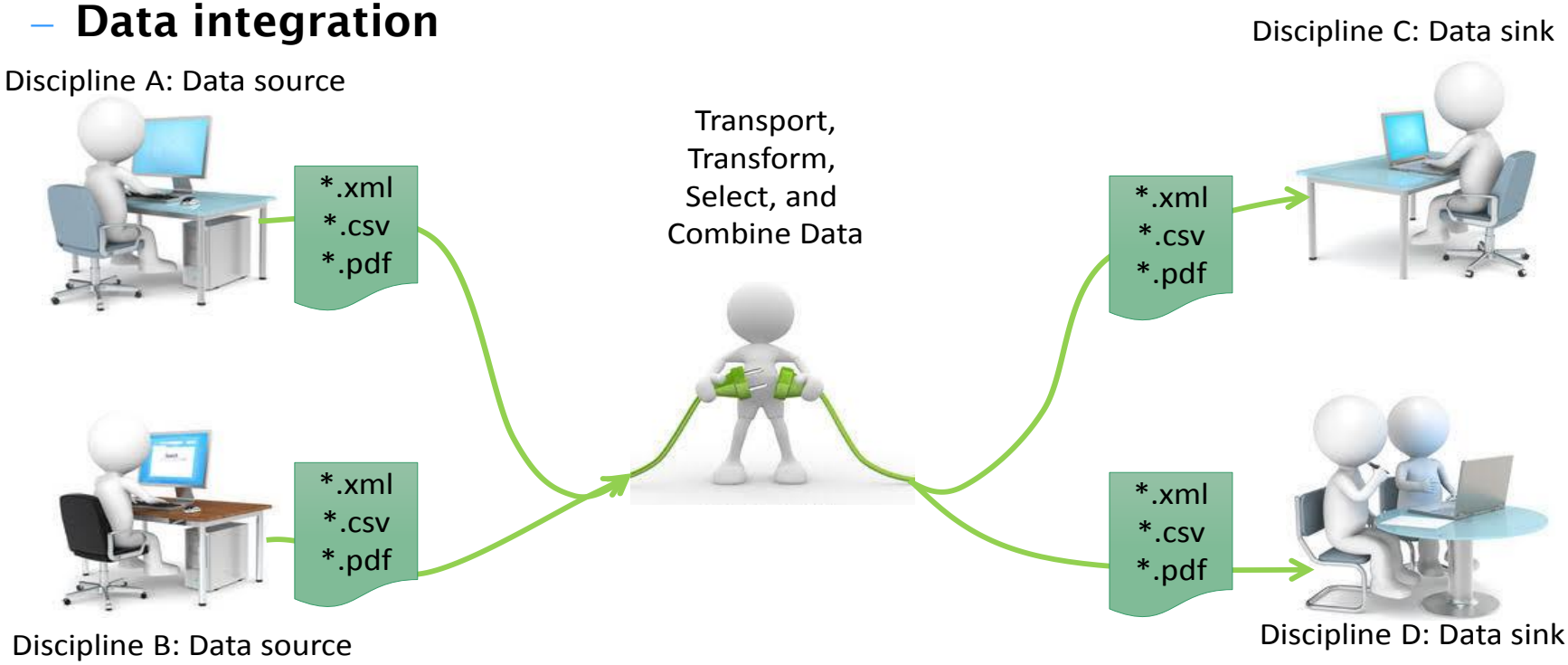


OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Starting points

- **Data exchange process between more than two different tools requires beyond syntax and semantics:**
  - **Identification of identical objects**
  - **Data integration**

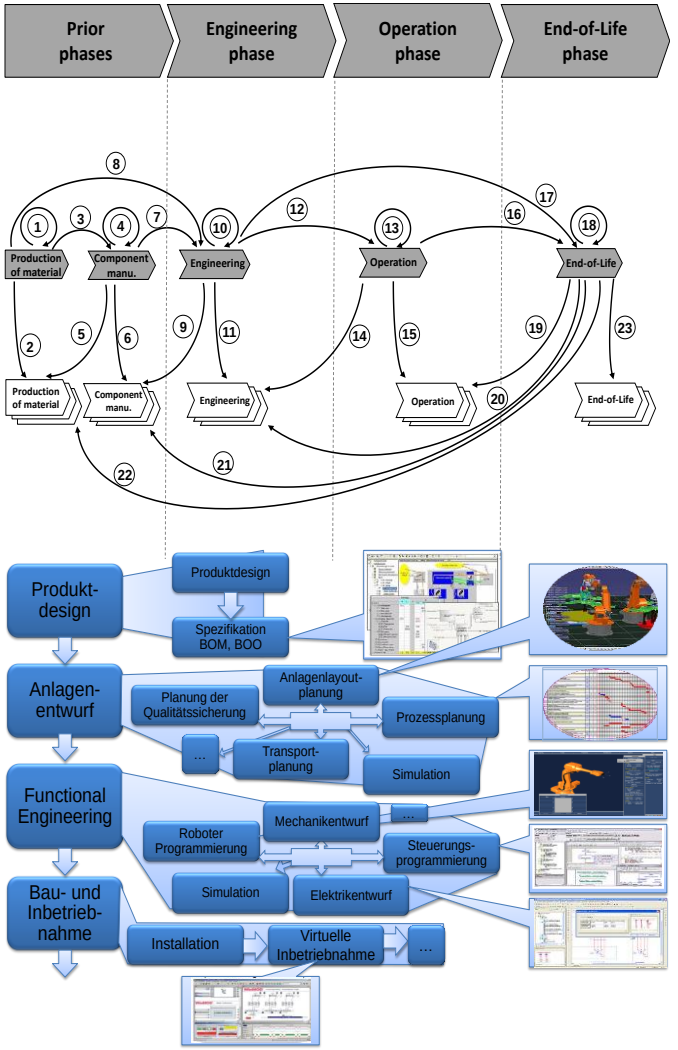


  - **Both must be supported by the data logistic system**

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

- **Complex Structures of data generation and use along the production system life cycle**
  - **multidisciplinary approach**
  - **several engineers of several disciplines with several tools**
  - **complex data exchange networks including data reuse of other life cycles**



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Problem

Within each discipline:

- **Engineering habit**

➔ **Usually used tools, information models, and engineering methods following best practice of the discipline**

➔ **Best tailored to discipline related needs**

➔ **Resulting in *discipline-specific data models with discipline related semantics***

Between disciplines:

- **Engineering networks**

➔Usually used data exchange formats following best practice of discipline connection

➔Best tailored to data exchange needs

➔Resulting in *data models applied for data exchange between engineering disciplines with discipline integrating semantics*

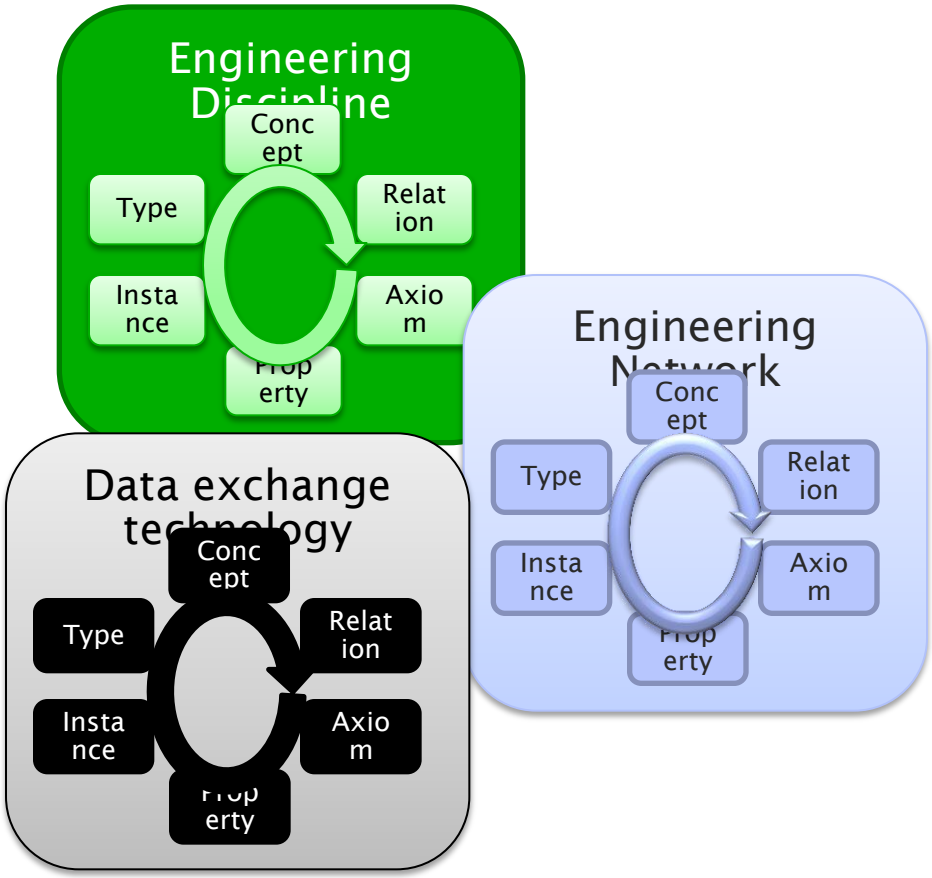
NOT HARMONIZED

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

Challenge – two streams of data models

apl. Prof. Dr.- Ing. habil. A. Lüder

- **There are various views on semantics within engineering networks**
  - Semantics of engineering disciplines
  - Semantics of engineering network
  - Semantics of data exchange technologies
  - …



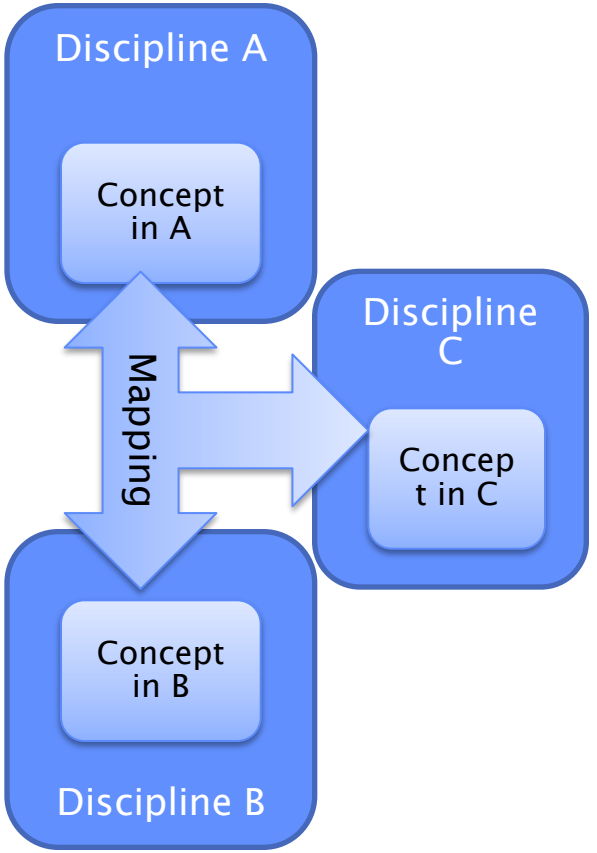
Challenge – different views on semantics

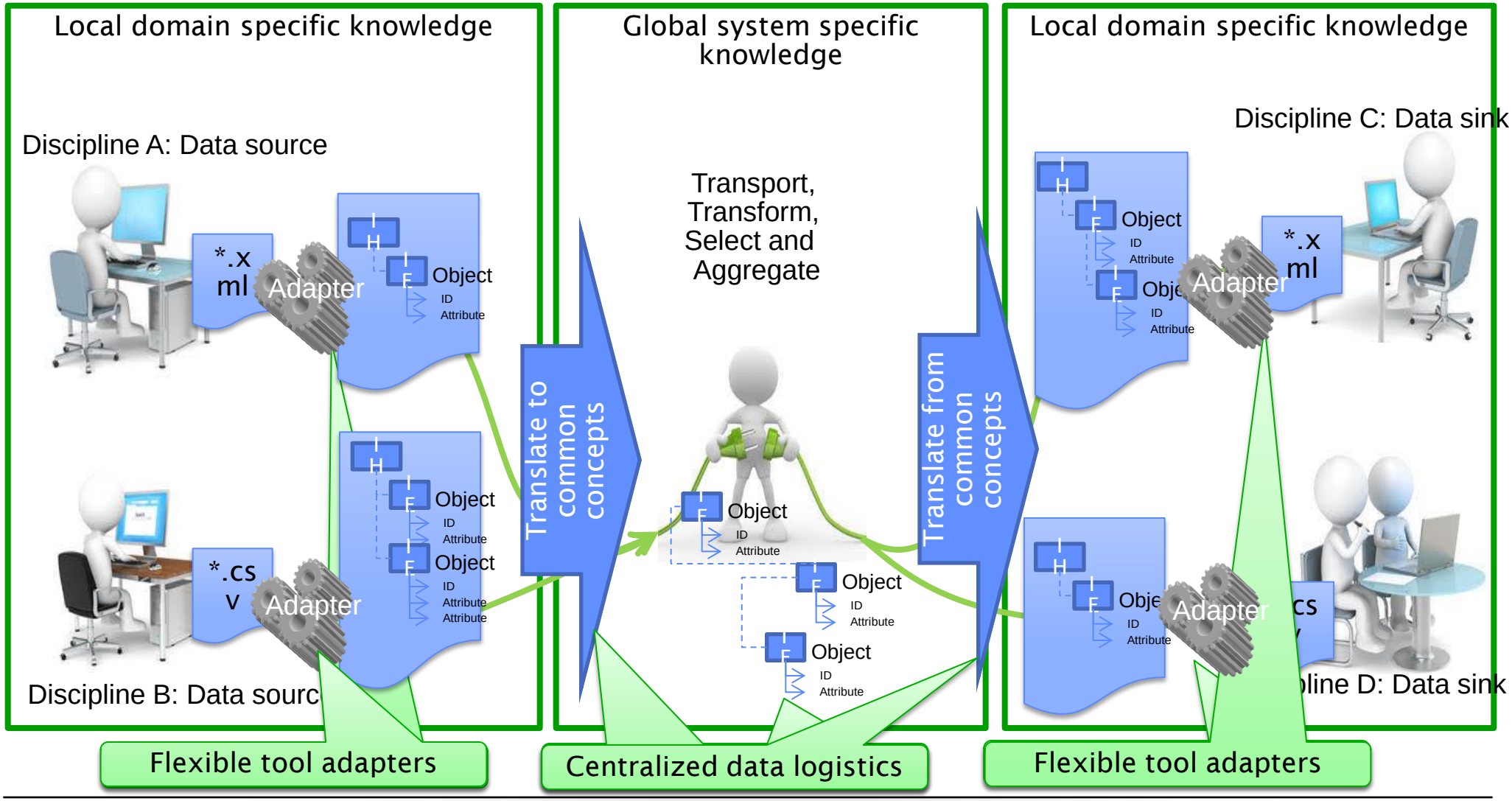
apl. Prof. Dr.- Ing. habil. A. Lüder

- Development of the two sets of data models:
  - discipline-specific data models
  - discipline-crossing data models

  Reflecting the necessary coupling of local discipline-specific data models by
  - mapping the model elements and
  - converting data content between the data models

➔ Data models shall describe identical facts always in a uniform way and this description semantics has to be possible to map to each other (VDI Guideline 3685)





Engineering data logistics

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

- Each engineer thinks in terms of conceptualized objects that can be related to each other

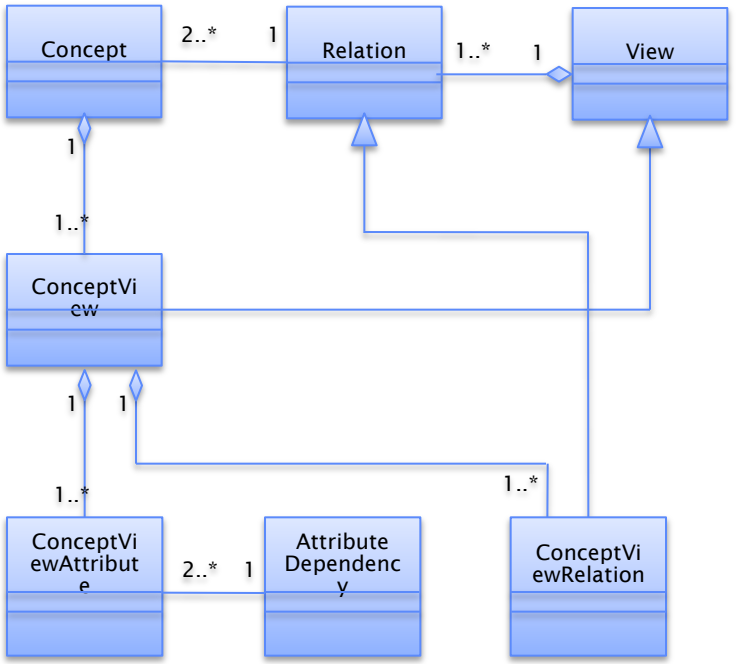
➔ Identification of ***common object concepts*** by discussing the *specific view* on these concepts with engineers in a discipline

➔ Discipline-specific views contain

- ➔ **Attributes**
- ➔ **Relations between concepts**

➔ Attributes within views may depend on each other

- ➔ **Dependencies can be based on natural laws, best practices, regulations, and other sources**
- ➔ **Often cross discipline/view boundaries**

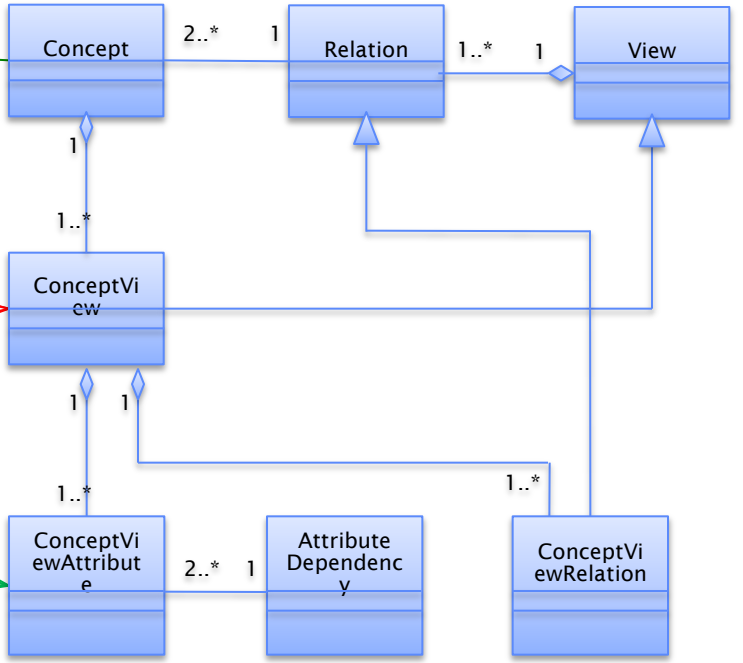


Data integration meta model

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

```
- concept: Project
  views:
    - view: PPR
    - view: Function
      derivedFrom: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure
      attributes:
        - attribute: FunctionViewID
          dataType: xs:string
          isIdentifier: true
          relations:
            - relation: source
              mimeType: smsxml
              fileSemantic: PCCExport?name;id
              reference: PARA_PXNAME
        - attribute: Description
        - attribute: Manufacturer name
        - attribute: Manufacturer part number
        - attribute: Functional description
      interfaces:
        - viewInterface: PngRef
          derivedFrom: FunctionViewInterfaceClassLib/PngRef
    - view: Mechanic
      derivedFrom: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure
      attributes:
      interfaces:
    - view: Electric
      derivedFrom: AutomationProjectConfigurationRoleClassLib/AutomationProject
      attributes:
        - attribute: ElectricViewID
          dataType: xs:string
          isIdentifier: true
      interfaces:
        - viewInterface: PngRef
          derivedFrom: ElectricViewInterfaceClassLib/PngRef
```



Representation by jaml

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

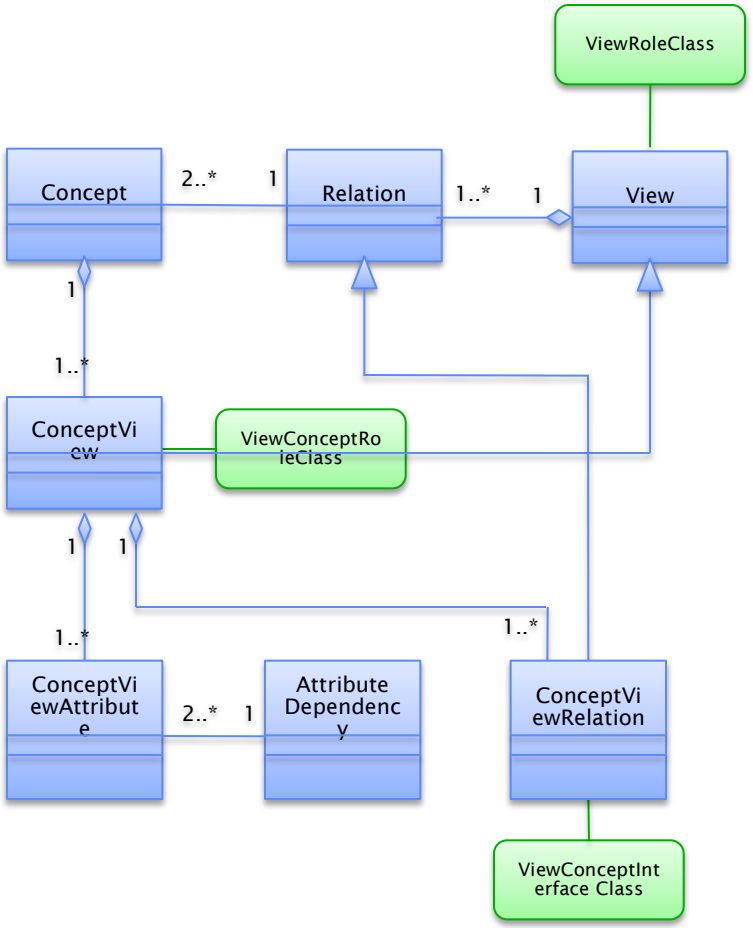
apl. Prof. Dr.- Ing. habil. A. Lüder

Step 1: Identification of views

- Identification of engineering disciplines with
  - data models applied within the engineering discipline and
  - data models applied for data exchange between engineering disciplines
- Definition of data-model-indicating *role classes for each view*, i.e., *view role classes*
  - Accompanied by *interface classes* required to model the hierarchy relations within the view

Step 2: Identification of view-related concepts

- Identification of view-related *concepts*
- Definition of concept related *view role classes*
  - Including *attributes* modelling the properties relevant for the *concept* in the considered *view*
  - Including *interfaces* used to represent the *relations* in the *view*



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

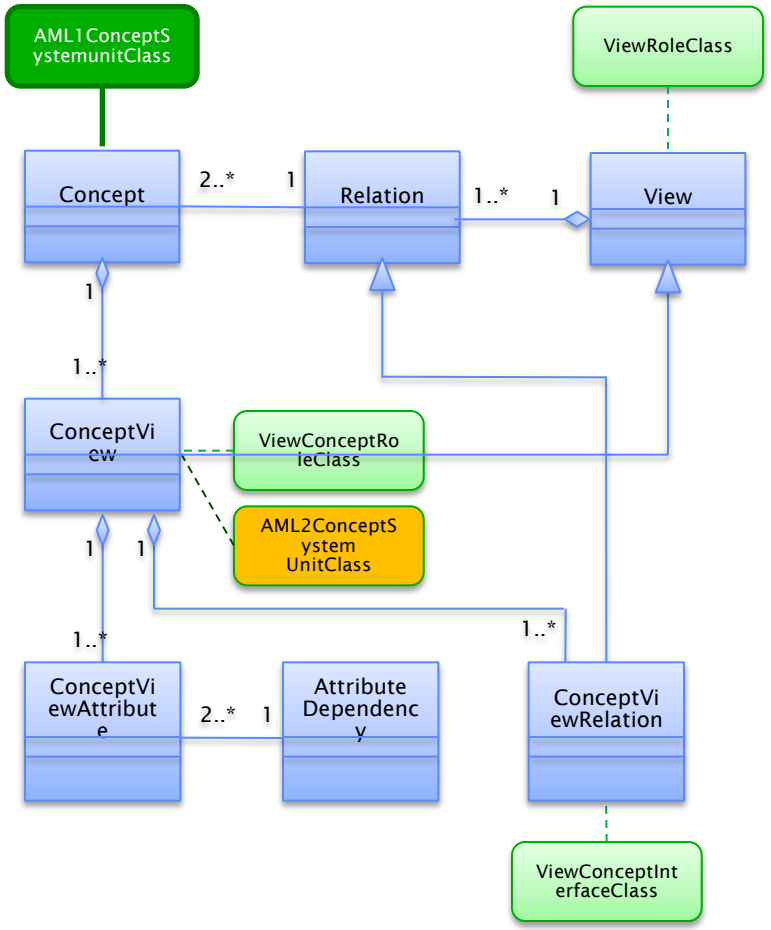
apl. Prof. Dr.- Ing. habil. A. Lüder

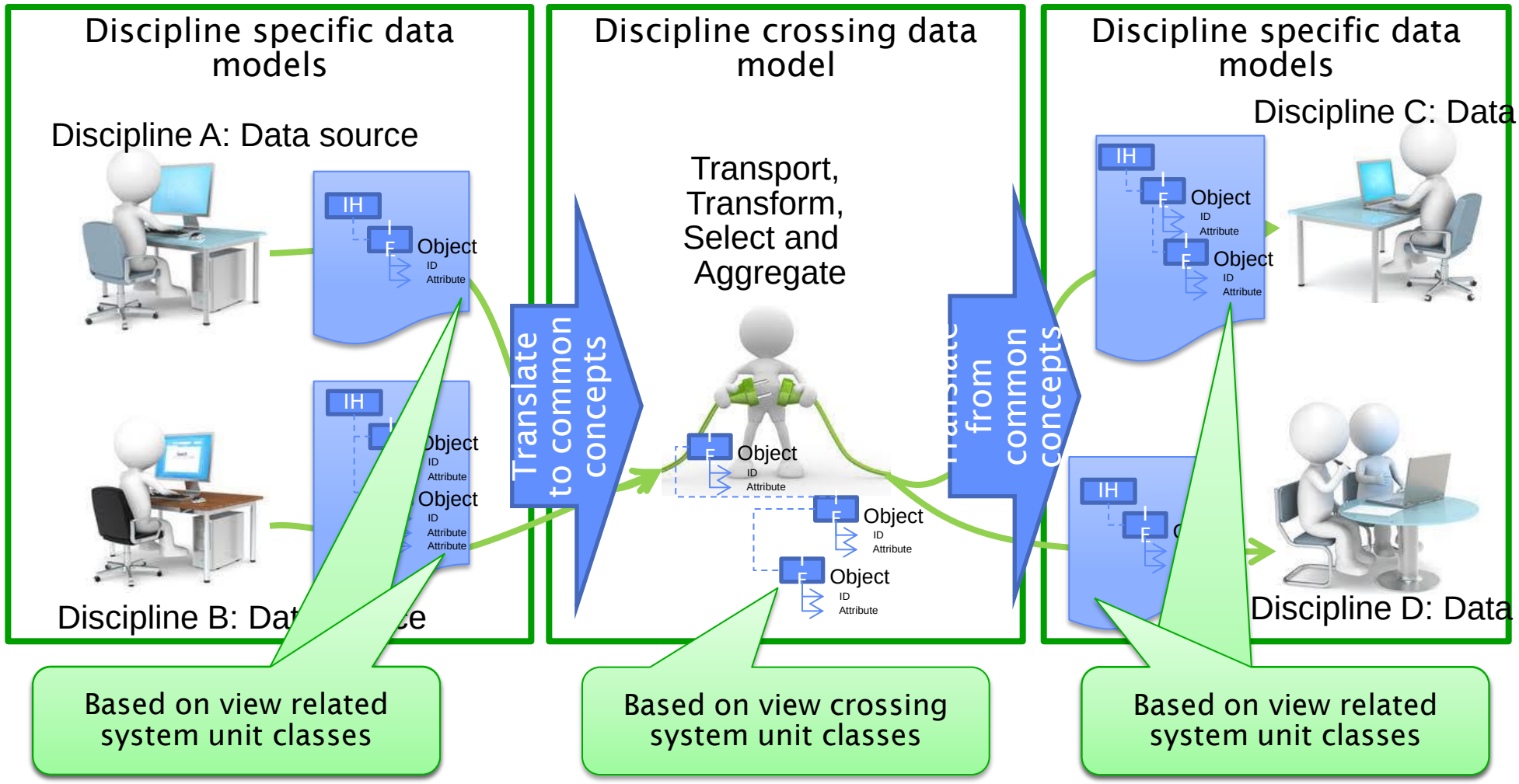
Step 3: Design of single-discipline data models

- Define *single-discipline data model* for each *view by* appropriate *system unit classes* using the *view-related roles*
- For each relevant *concept*, a *system unit class* is developed
  - Referencing related role as *role class*
  - Containing *attributes* of this role
  - Containing *interfaces* of this *view*

Step 4: Design of multi-discipline data model

- Define *multi-discipline data model* by combining *single-discipline data models*
- For each concept, an *system unit class* is developed
  - Containing *InternalElements* created by instantiating the related single-discipline data model *system unit classes* of the *views* relevant for the *concept*
  - Containing dependencies between *attributes* of the different *InternalElements* of the *concept*-related *system unit class* modelled as *attribute* dependencies

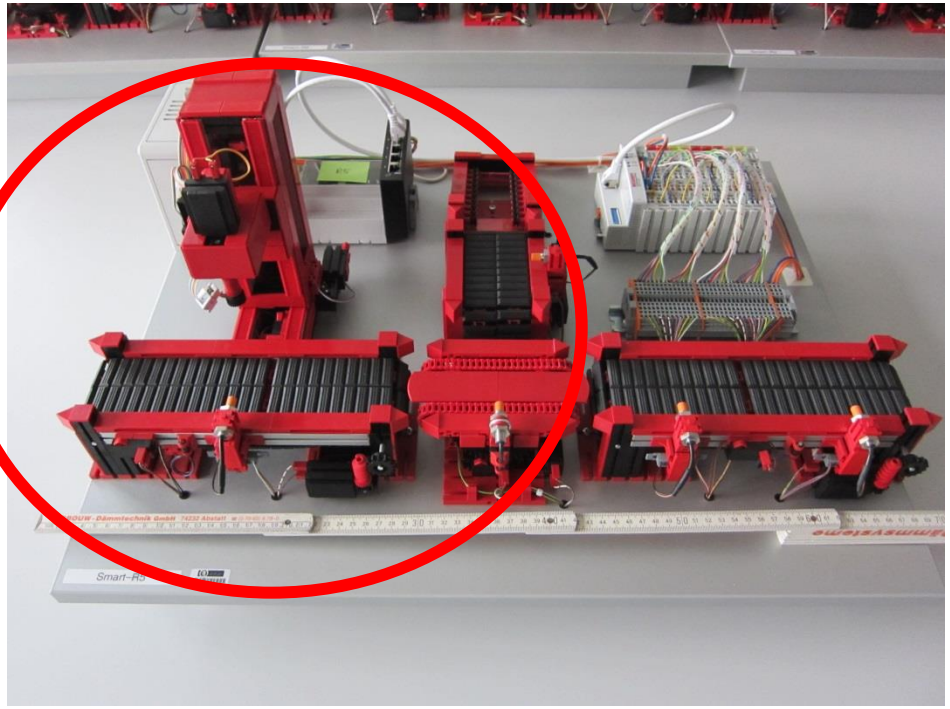


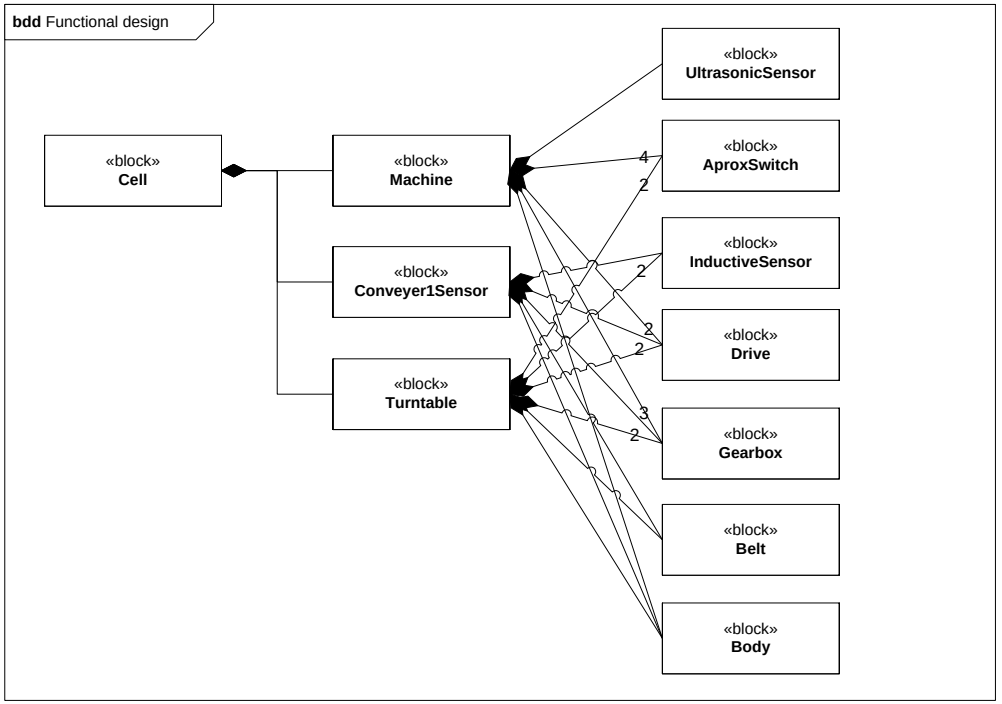


OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

Use process exploiting AutomationML

apl. Prof. Dr.- Ing. habil. A. Lüder

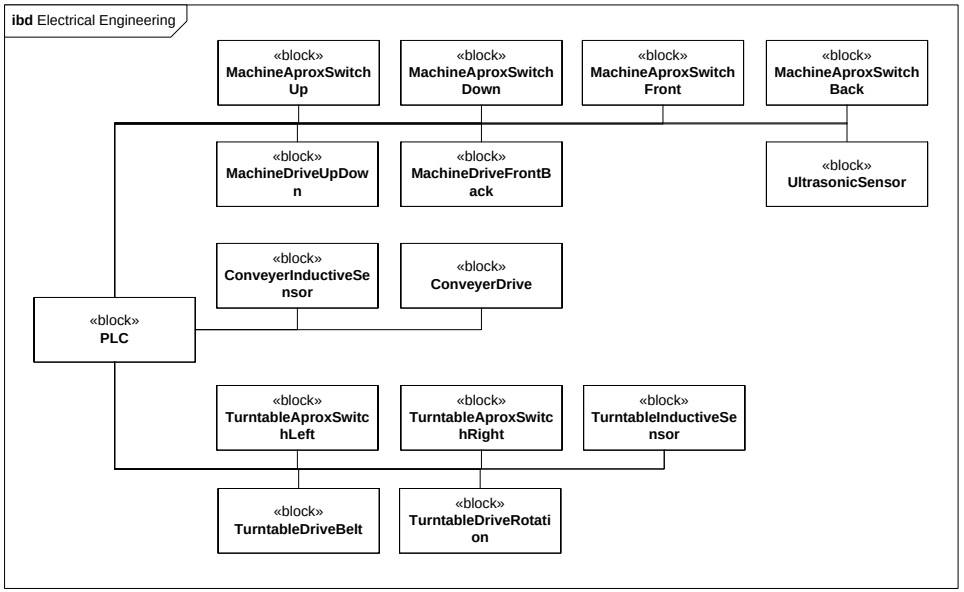


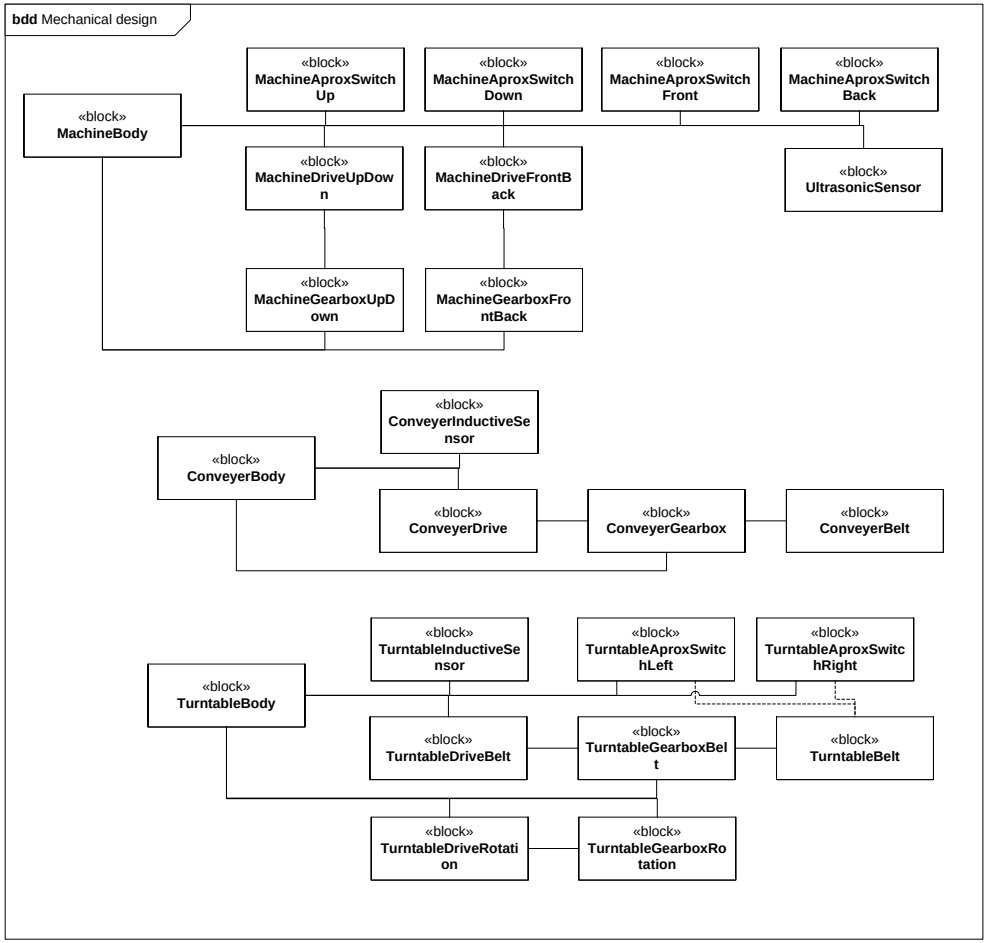


Example – Ultrasonic measurement cell

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

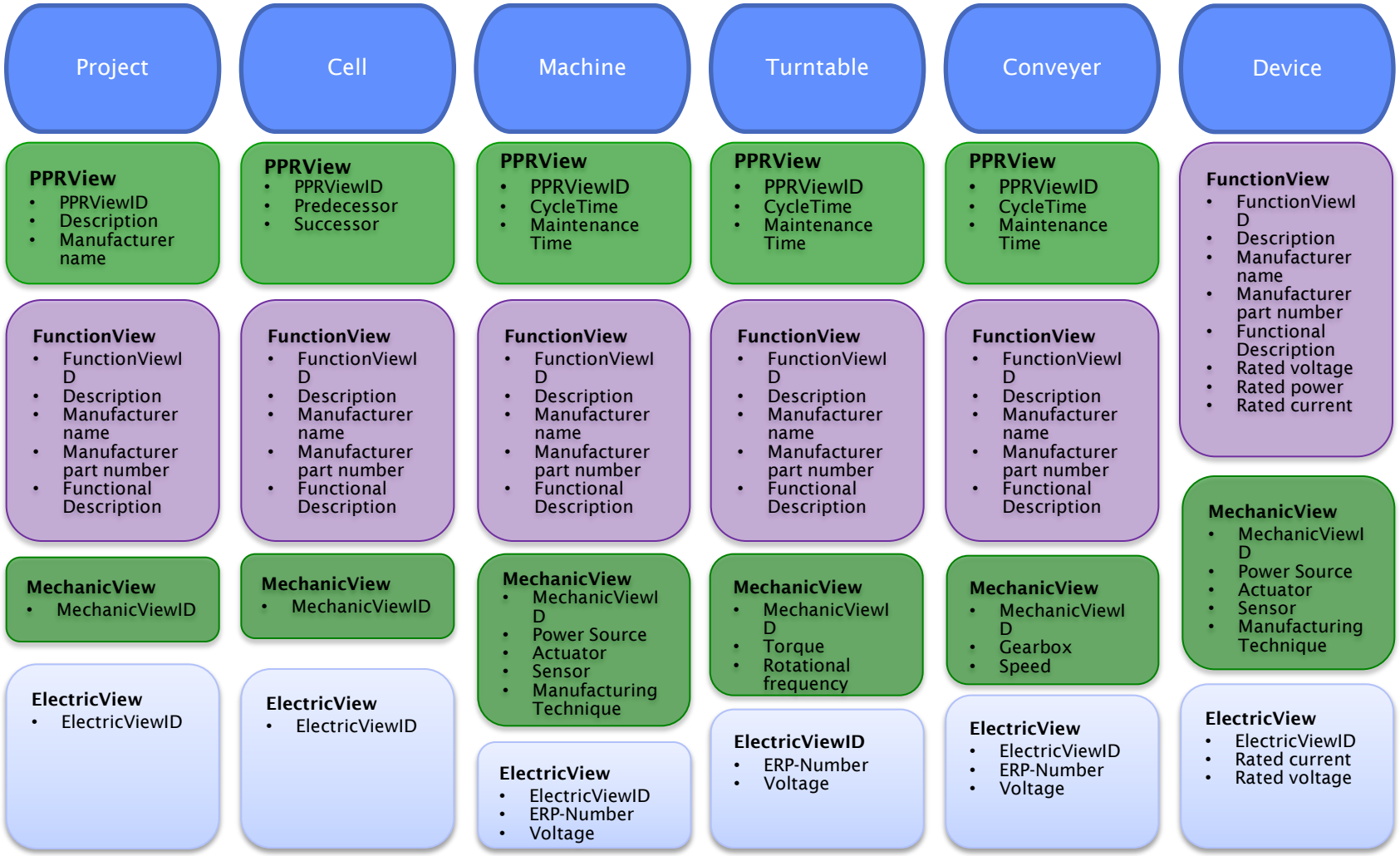




Example – Ultrasonic measurement cell

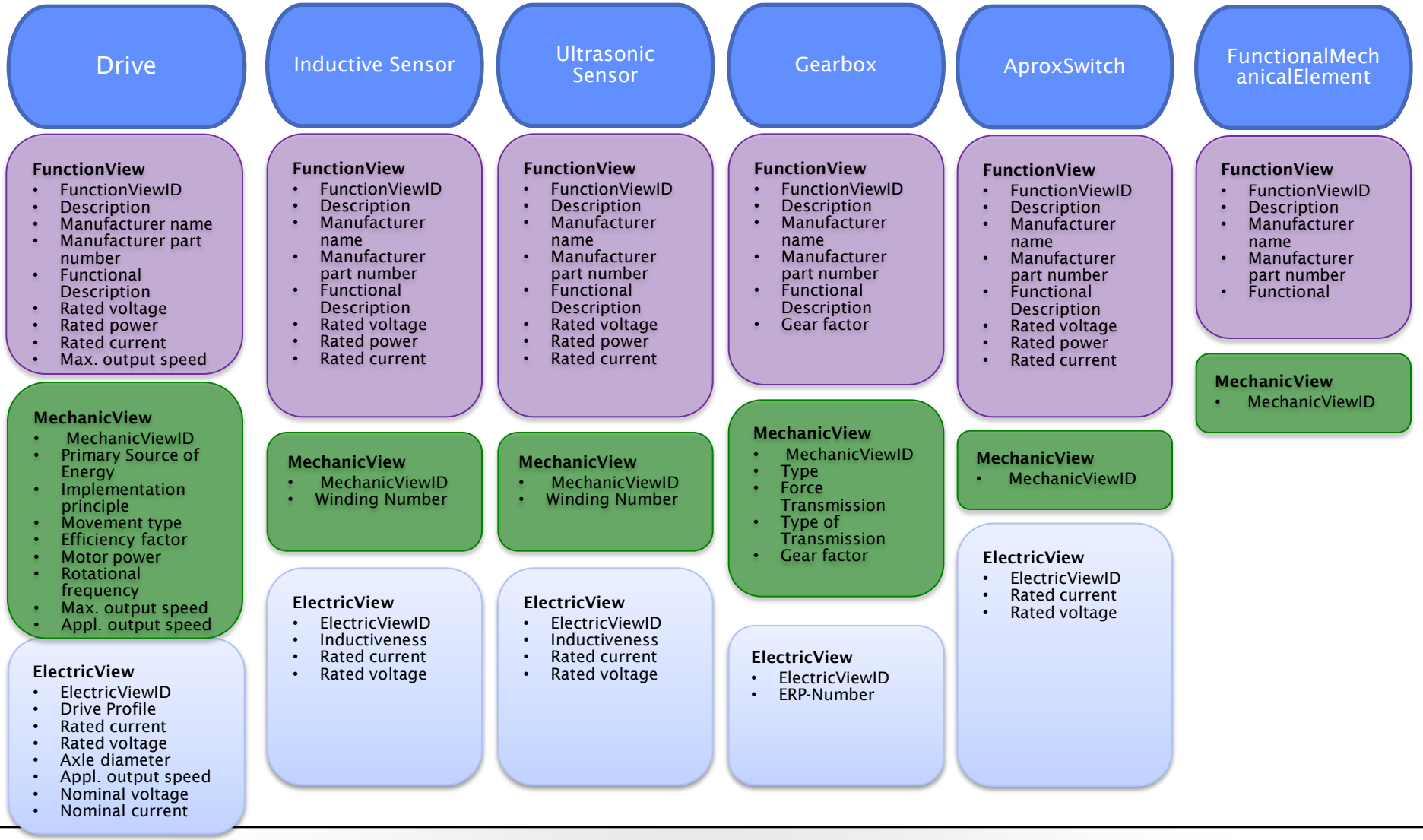
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder



Example – Ultrasonic measurement cell

apl. Prof. Dr.- Ing. habil. A. Lüder

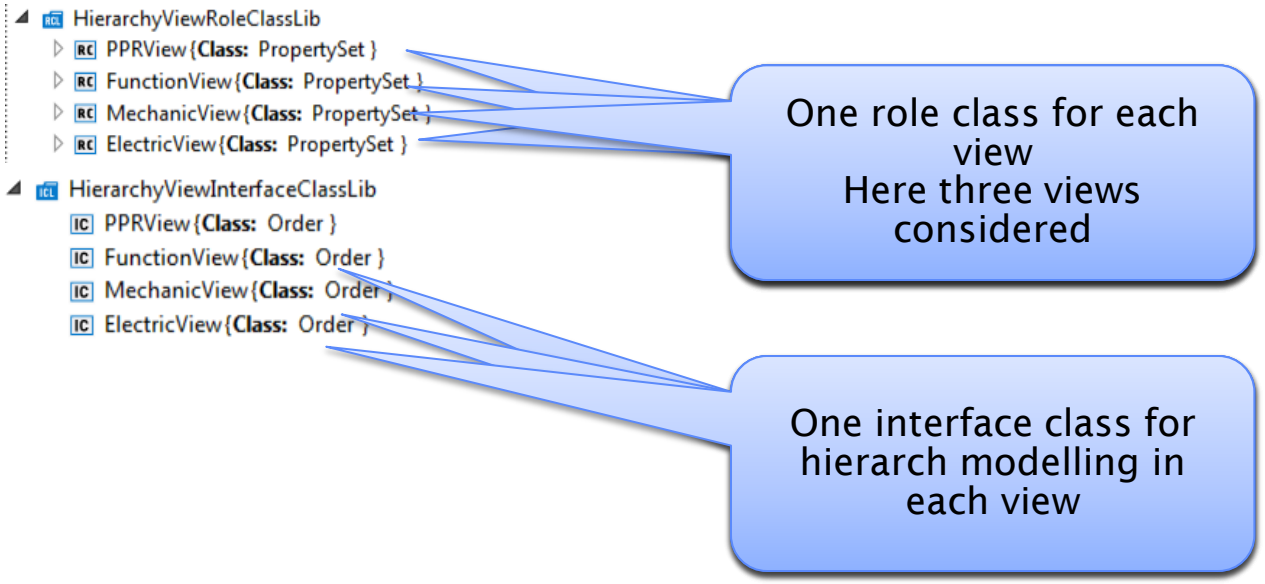


Example – Ultrasonic measurement cell

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

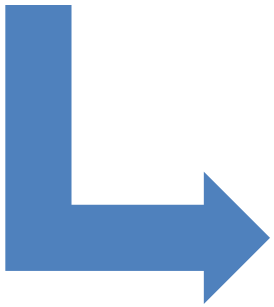
apl. Prof. Dr.- Ing. habil. A. Lüder

Step 1: Identification of views



Step 1

- HierarchyViewRoleClassLib
  - PPRView {**Class:** PropertySet }
  - FunctionView {**Class:** PropertySet }
  - MechanicView {**Class:** PropertySet }
  - ElectricView {**Class:** PropertySet }
- HierarchyViewInterfaceClassLib
  - PPRView {**Class:** Order }
  - FunctionView {**Class:** Order }
  - MechanicView {**Class:** Order }
  - ElectricView {**Class:** Order }



Step 2: Modelling of roles for each concept relevant in each view

- PPRViewRoleClassLib
- FunctionViewRoleClassLib
  - Project {**Class:** Structure }
  - Cell {**Class:** ResourceStructure }
  - Machine {**Class:** Resource }
  - Turntable {**Class:** Resource }
  - Conveyer {**Class:** Resource }
  - Device {**Class:** Resource }
  - Drive {**Class:** Resource }
  - InductivSensor {**Class:** Resource }
  - UltrasonicSensor {**Class:** Resource }
  - Gearbox {**Class:** Resource }
  - AproxSwitch {**Class:** Resource }
  - FunctionalMechanicalElement {**Class:** Resource }
- MechanicViewRoleClassLib
- ElectricViewRoleClassLib
  - Project {**Class:** AutomationProject }
  - Cell {**Class:** DeviceUserFolder }
  - Machine {**Class:** DeviceUserFolder }
  - Turntable {**Class:** DeviceUserFolder }
  - Conveyer {**Class:** DeviceUserFolder }
  - Device {**Class:** Device }
  - Drive {**Class:** Device }
  - InductivSensor {**Class:** Device }
  - UltrasonicSensor {**Class:** Device }
  - Gearbox {**Class:** Device }
  - AproxSwitch {**Class:** Device }

One role class for each concept relevant in the function view

One role class for each concept relevant in the mechanic view

One role class for each concept relevant in the electric view

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Use process exploiting AutomationML

Step 1

- HierarchyViewRoleClassLib
  - PPRView {**Class:** PropertySet }
  - FunctionView {**Class:** PropertySet }
  - MechanicView {**Class:** PropertySet }
  - ElectricView {**Class:** PropertySet }
- HierarchyViewInterfaceClassLib
  - PPRView {**Class:** Order }
  - FunctionView {**Class:** Order }
  - MechanicView {**Class:** Order }
  - ElectricView {**Class:** Order }

Step 2

- PPRViewRoleClassLib
- FunctionViewRoleClassLib
  - Project {**Class:** Structure }
  - Cell {**Class:** ResourceStructure }
  - Machine {**Class:** Resource }
  - Turntable {**Class:** Resource }
  - Conveyer {**Class:** Resource }
  - Device {**Class:** Resource }
  - Drive {**Class:** Resource }
  - InductivSensor {**Class:** Resource }
  - UltrasonicSensor {**Class:** Resource }
  - Gearbox {**Class:** Resource }
  - AproxSwitch {**Class:** Resource }
  - FunctionalMechanicalElement {**Class:** Resource }
- MechanicViewRoleClassLib
- ElectricViewRoleClassLib
  - Project {**Class:** AutomationProject }
  - Cell {**Class:** DeviceUserFolder }
  - Machine {**Class:** DeviceUserFolder }
  - Turntable {**Class:** DeviceUserFolder }
  - Conveyer {**Class:** DeviceUserFolder }
  - Device {**Class:** Device }
  - Drive {**Class:** Device }
  - InductivSensor {**Class:** Device }
  - UltrasonicSensor {**Class:** Device }
  - Gearbox {**Class:** Device }
  - AproxSwitch {**Class:** Device }

Step 3: Modelling of SystemUnitClasses as templates for applicable objects in each view

- AML2PPRViewSystemUnitClassLib
- AML2FunctionViewSystemUnitClassLib
  - Project {**Role:** Project, FunctionView}
  - Cell {**Role:** Cell, FunctionView}
  - Machine {**Role:** Machine, FunctionView}
  - Turntable {**Role:** Turntable, FunctionView}
  - Conveyer {**Role:** Conveyer, FunctionView}
  - Device {**Role:** Device, FunctionView}
  - Drive {**Role:** Drive, FunctionView}
    - Function_PngRef {**Class:** PngRef }
    - FunctionView_toMother {**Class:** FunctionView }
    - FunctionView_toChild {**Class:** FunctionView }
    - FunctionViewRoleClassLib/Drive
    - HierarchyViewRoleClassLib/FunctionView
  - InductivSensor {**Role:** InductivSensor, FunctionView}
  - UltrasonicSensor {**Role:** UltrasonicSensor, FunctionView}
  - Gearbox {**Role:** Gearbox, FunctionView}
  - AproxSwitch {**Role:** AproxSwitch, FunctionView}
  - FunctionalMechanicalElement {**Role:** Functional…
- AML2MechanicViewSystemUnitClassLib
- AML2ElectricViewSystemUnitClassLib
  - Project {**Role:** Project, ElectricView}
  - Cell {**Role:** Cell, ElectricView}
  - Machine {**Role:** Machine, ElectricView}
  - Turntable {**Role:** Turntable, ElectricView}
  - Conveyer {**Role:** Conveyer, ElectricView}
  - Device {**Role:** Device, ElectricView}
  - Drive {**Role:** Drive, ElectricView}
  - InductivSensor {**Role:** InductivSensor, ElectricV…
  - UltrasonicSensor {**Role:** UltrasonicSensor, Elect…
  - Gearbox {**Role:** Gearbox, ElectricView}
  - AproxSwitch {**Role:** AproxSwitch, ElectricView}

One system unit class for each concept relevant in the function view

One system unit for each concept relevant in the mechanic view

One system unit for each concept relevant in the electric view

View related system unit classes are applied by discipline specific tools to export/import proper engineering objects.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

Use process exploiting AutomationML

apl. Prof. Dr.- Ing. habil. A. Lüder

Step 1

- HierarchyViewRoleClassLib
  - PPRView {**Class:** PropertySet }
  - FunctionView {**Class:** PropertySet }
  - MechanicView {**Class:** PropertySet }
  - ElectricView {**Class:** PropertySet }
- HierarchyViewInterfaceClassLib
  - PPRView {**Class:** Order }
  - FunctionView {**Class:** Order }
  - MechanicView {**Class:** Order }
  - ElectricView {**Class:** Order }

Step 2

- PPRViewRoleClassLib
- FunctionViewRoleClassLib
  - Project {**Class:** Structure }
  - Cell {**Class:** ResourceStructure }
  - Machine {**Class:** Resource }
  - Turntable {**Class:** Resource }
  - Conveyer {**Class:** Resource }
  - Device {**Class:** Resource }
  - Drive {**Class:** Resource }
  - InductivSensor {**Class:** Resource }
  - UltrasonicSensor {**Class:** Resource }
  - Gearbox {**Class:** Resource }
  - AproxSwitch {**Class:** Resource }
  - FunctionalMechanicalElement {**Class:** Resource }
- MechanicViewRoleClassLib
- ElectricViewRoleClassLib
  - Project {**Class:** AutomationProject }
  - Cell {**Class:** DeviceUserFolder }
  - Machine {**Class:** DeviceUserFolder }
  - Turntable {**Class:** DeviceUserFolder }
  - Conveyer {**Class:** DeviceUserFolder }
  - Device {**Class:** Device }
  - Drive {**Class:** Device }
  - InductivSensor {**Class:** Device }
  - UltrasonicSensor {**Class:** Device }
  - Gearbox {**Class:** Device }
  - AproxSwitch {**Class:** Device }

Step 3

- AML2PPRViewSystemUnitClassLib
- AML2FunctionViewSystemUnitClassLib
  - Project {**Role:** Project, FunctionView}
  - Cell {**Role:** Cell, FunctionView}
  - Machine {**Role:** Machine, FunctionView}
  - Turntable {**Role:** Turntable, FunctionView}
  - Conveyer {**Role:** Conveyer, FunctionView}
  - Device {**Role:** Device, FunctionView}
  - Drive {**Role:** Drive, FunctionView}
    - Function_PngRef {**Class:** PngRef }
    - FunctionView_toMother {**Class:** FunctionView }
    - FunctionView_toChild {**Class:** FunctionView }
    - FunctionViewRoleClassLib/Drive
    - HierarchyViewRoleClassLib/FunctionView
  - InductivSensor {**Role:** InductivSensor, FunctionView}
  - UltrasonicSensor {**Role:** UltrasonicSensor, FunctionView}
  - Gearbox {**Role:** Gearbox, FunctionView}
  - AproxSwitch {**Role:** AproxSwitch, FunctionView}
  - FunctionalMechanicalElement {**Role:** FunctionalMechanicalElement, Functio
- AML2MechanicViewSystemUnitClassLib
- AML2ElectricViewSystemUnitClassLib
  - Project {**Role:** Project, ElectricView}
  - Cell {**Role:** Cell, ElectricView}
  - Machine {**Role:** Machine, ElectricView}
  - Turntable {**Role:** Turntable, ElectricView}
  - Conveyer {**Role:** Conveyer, ElectricView}
  - Device {**Role:** Device, ElectricView}
  - Drive {**Role:** Drive, ElectricView}
  - InductivSensor {**Role:** InductivSensor, ElectricView}
  - UltrasonicSensor {**Role:** UltrasonicSensor, ElectricView}
  - Gearbox {**Role:** Gearbox, ElectricView}
  - AproxSwitch {**Role:** AproxSwitch, ElectricView}

Overall system unit classes are applied within data logistics to transport and integrate proper engineering objects.

Step 4: Modelling of SystemUnitClasses as templates for applicable objects over all views

- AML1ConceptSystemUnitClassLib
  - Project {**Role:** Project}
  - Cell {**Role:** Cell}
  - Machine {**Role:** Machine}
  - Turntable {**Role:** Turntable}
  - Conveyer {**Role:** Conveyer}
  - Device {**Role:** Device}
  - Drive {**Role:** Drive}
    - FunctionView {**Class:** Drive **Role:** Drive, Func...
    - MechanicView {**Class:** Drive **Role:** Drive, MechanicView}
    - ElectricView {**Class:** Drive **Role:** Drive, ElectricView}
    - FunctionViewRoleClassLib/Drive
    - MechanicViewRoleClassLib/Drive
    - ElectricViewRoleClassLib/Drive
  - InductivSensor {**Role:** InductivSensor}
  - UltrasonicSensor {**Role:** UltrasonicSensor}
  - Gearbox {**Role:** Gearbox}
  - AproxSwitch {**Role:** AproxSwitch}
  - FunctionalMechanicalElement {**Role:** FunctionalMechanicalElement}
  - Product {**Role:** Product}
  - Process {**Role:** Process}
  - RequiredProcess {**Role:** RequiredProcess}
  - ProvidedProcess {**Role:** ProvidedProcess}
  - Skill {**Role:** Skill}
  - Rotate {**Role:** Rotate}
  - Translate {**Role:** Translate}
  - Measure {**Role:** Measure}

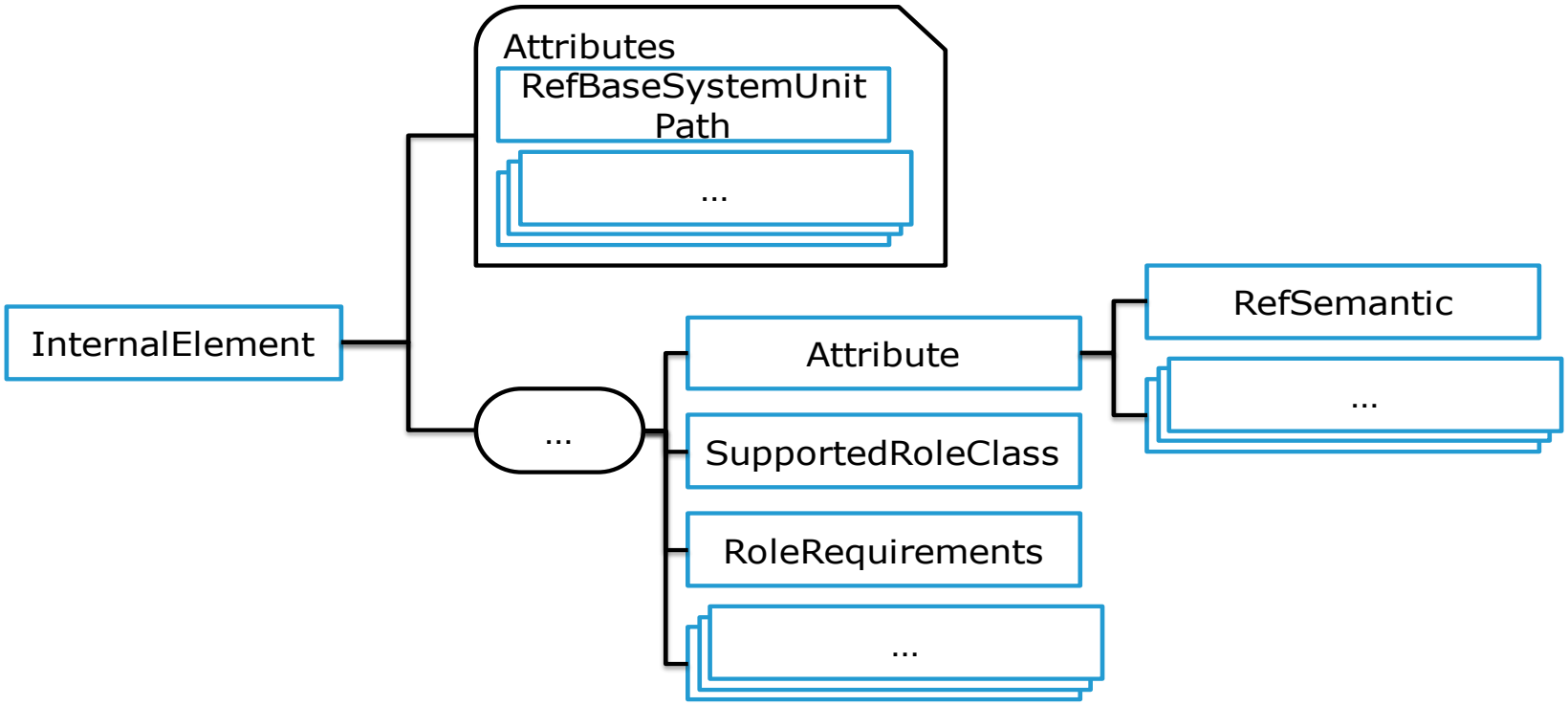
One system unit for each concept crossing all views

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Use process exploiting AutomationML

Means for Semantic Integration in InternalElements and Attributes



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

- No definition of semantics of production system components itself by AutomationML
- But integration of existing semantic definitions
  - Given for example in the eCl@ss classification standard
- eCl@ss
  - Hierarchical semantic system for grouping materials, products, and services
    - » According to a logical structure with a level of detail corresponding to product-specific properties described using standard conform properties
  - Classification of materials, products, and services enabling a unique identification of production system component classes like devices types or installation material types
    - » For each class definition of standardized properties useable for specification of individual characteristics of the class instances

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

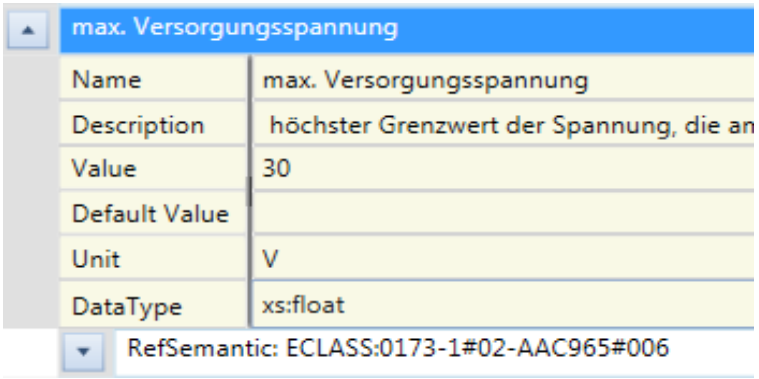
Integration of Object Semantics

- **Key element of the eCl@ss specification:**

  IRDI (International Registration Data Identifier)
  - **Based on the international standards ISO/IEC 11179-6, ISO 29002, and ISO 6532**
  - **Providing unique identification code for each attribute and each class of objects by IRDI**
- **Exploiting the referencing of the IRDI of eCl@ss properties for referencing of semantics of an attribute AutomationML**

➔ **Attribute CorrespondingAttributePath of the CAEX schema element RefSemantic assembled as the string "IRDIPath: " + IRDI of the eCl@ss property defining the semantics of the AutomationML attribute**

- **Example of Semantic Representation for Attributes**
- **Attribute: max. Versorgungsspannung representing the maximal applicable supply voltage for an inductive sensor**
  - **Semantically defined by the IRDI 0173-1#02-AAC962#006.**

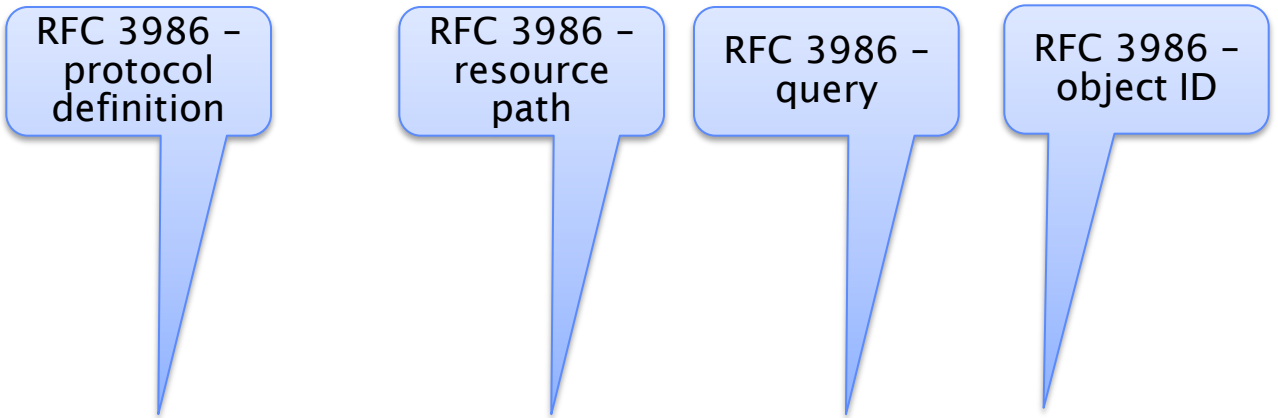



```
<Attribute Name="max. Versorgungsspannung" AttributeDataType="xs:float" Unit="V">
  <Description> höchster Grenzwert der Spannung, die am Versorgungseingang eines elektrischen Betriebsmittels zeitweise oder ständig anliegen muss</Description>
  <Value>30</Value>
  <RefSemantic CorrespondingAttributePath="ECLASS:0173-1#02-AAC965#006" />
</Attribute>
```


OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

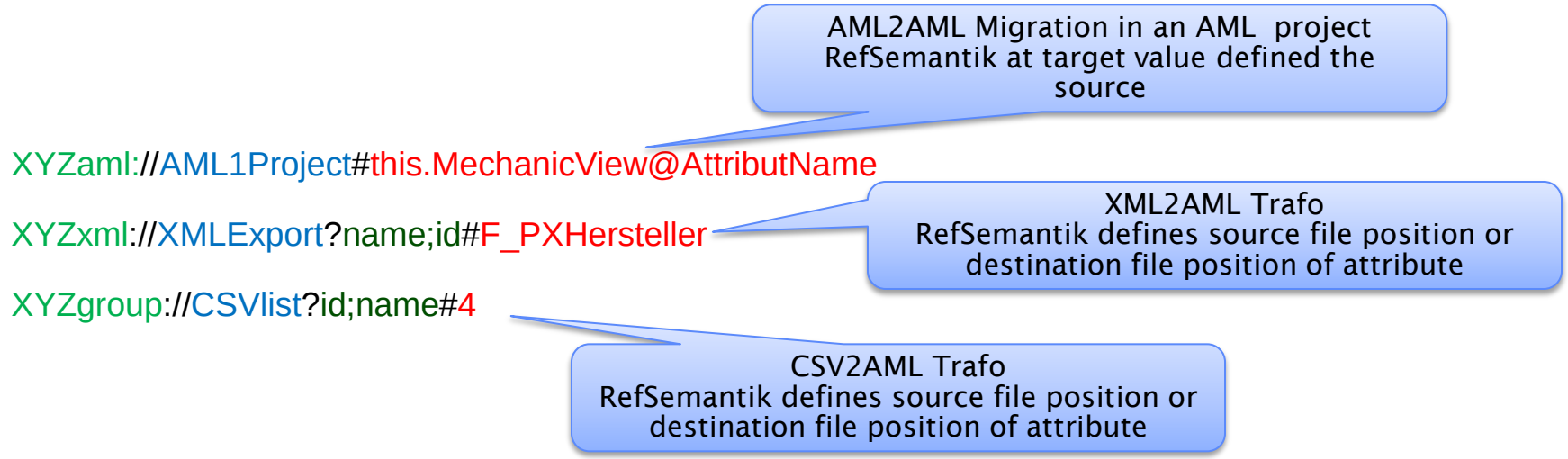
apl. Prof. Dr.- Ing. habil. A. Lüder

- **We can extend this idea by utilizing the RFC 3986 - Uniform Resource Identifier (URI)_ Generic Syntax**



apl. Prof. Dr.- Ing. habil. A. Lüder

Utilization of RefSemantics for configuration of data migration within the data logistics



apl. Prof. Dr.- Ing. habil. A. Lüder

Concept identification and modelling

Reference source and sink data structure

Enable stepwise extension of exchanged data by simple configuration of adapter models

Functionally relevant	OU	User	Project internal ID	Denomination -German-	Denomination -English-	Quantity	Axle ø	Velocity/ rotational speed/ cycle time	Nominal voltage	Nominal current
Funktions-relevant	OE	Nutzer	Projekt interne ID	Benennung -deutsch-	Benennung -englisch-	Anzahl	Achs-ø	Geschwindig-keit / Drehzahl / Zykluszeit	Nominalspannung	Nominalstrom
				Einheit/ Formatierungsvorgabe:		2-stellig	mm	---	V	A
			PS	**Production system**						
			PS_SubS_UMC	**Cell_02**	**Measurement system**					
			PS_SubS_UMC_VAP_MM	**Machine**	**Measurement machine**					
x	AFR	MUTO	PS_SubS_UMC_VAP_MM_CD1_D1	Drive01		1	200	200	24	0.2
x	AFR	MUTO	PS_SubS_UMC_VAP_MM_CD2_D1	Drive02		1	200	200	24	0.2
			PS_SubS_UMC_VAP_CS	**Conveyer**						
x	AFR	MUTO	PS_SubS_UMC_VAP_CS_CD1_D1	Drive01		1	200	200	24	0.1
			PS_SubS_UMC_VAP_TT	**Turntable**						
	AFR	MUTO	PS_SubS_UMC_VAP_TT_CD1_D1	Drive01		1	200	200	24	0.1
		MUTO	PS_SubS_UMC_VAP_TT_CD2_D1	Drive02		1	200	100	24	0.1

- Device {**Class:** Device }
  - Electric_PngRef {**Class:** PngRef }
- Drive {**Class:** Device }
  - Electric_PngRef {**Class:** PngRef }

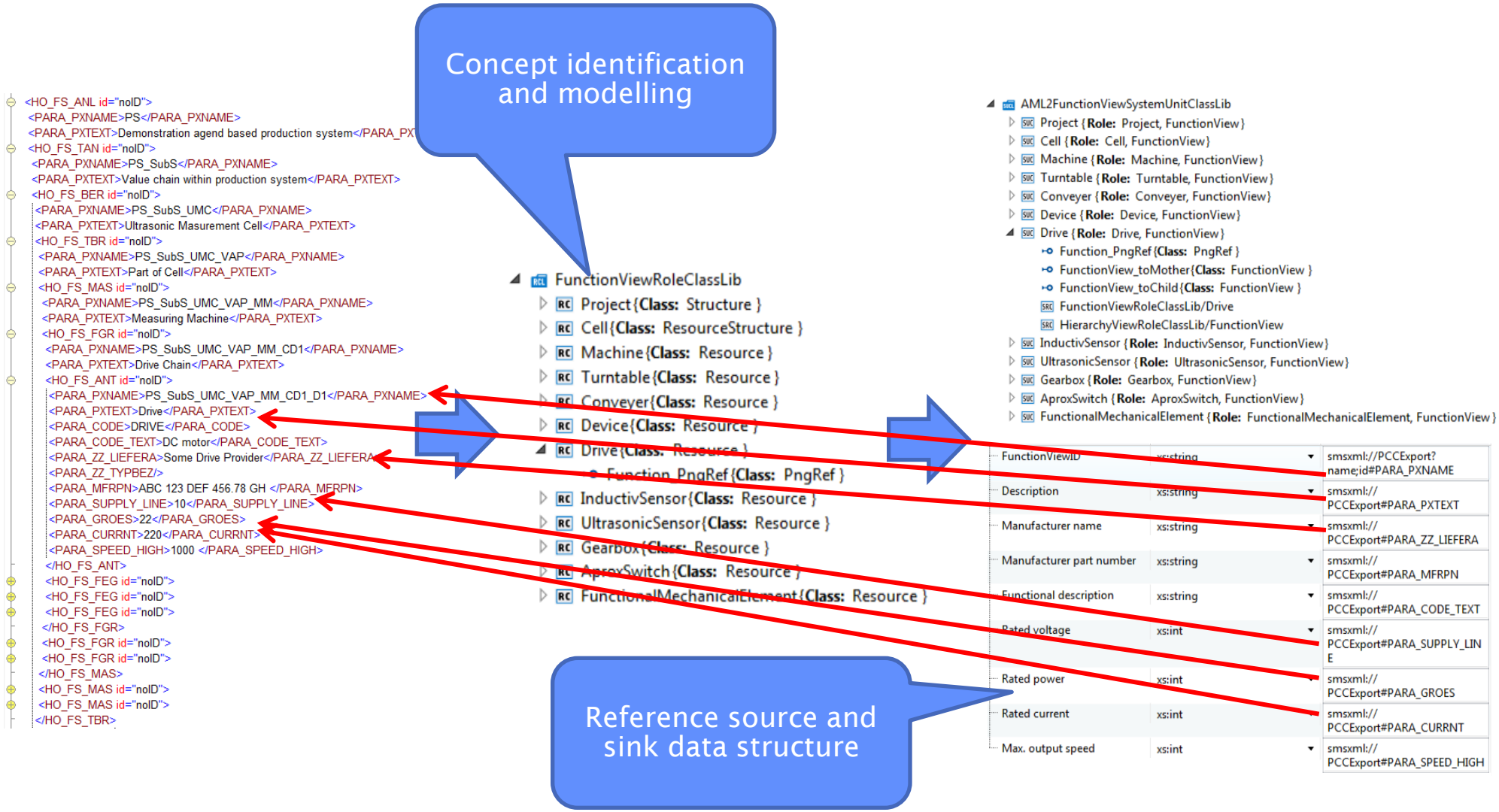
- Drive {**Role:** Drive, ElectricView}
  - Electric_PngRef {**Class:** PngRef }
  - ElectricView_toMother {**Class:** ElectricView }
  - ElectricView_toChild {**Class:** ElectricView }
  - ElectricViewRoleClassLib/Drive
  - HierarchyViewRoleClassLib/ElectricView

ElectricViewID	xs:string	sms:group://drivelist?name;id#4
Drive Profile	xs:string	
Rated current	xs:int	aml://AML1Project#this.FunctionView@Rated current
Rated voltage	xs:int	aml://AML1Project#this.FunctionView@Rated voltage
Axle diameter	xs:int	smsgroup://drivelist#8
Appl. output speed	xs:int	smsgroup://drivelist#9
Nominal voltage	xs:int	smsgroup://drivelist#10
Nominal current	xs:int	smsgroup://drivelist#11

Use process exploiting AutomationML

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder



Use process exploiting AutomationML

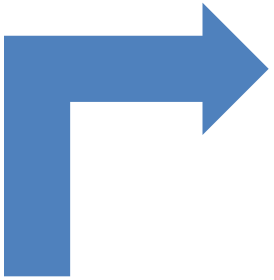
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

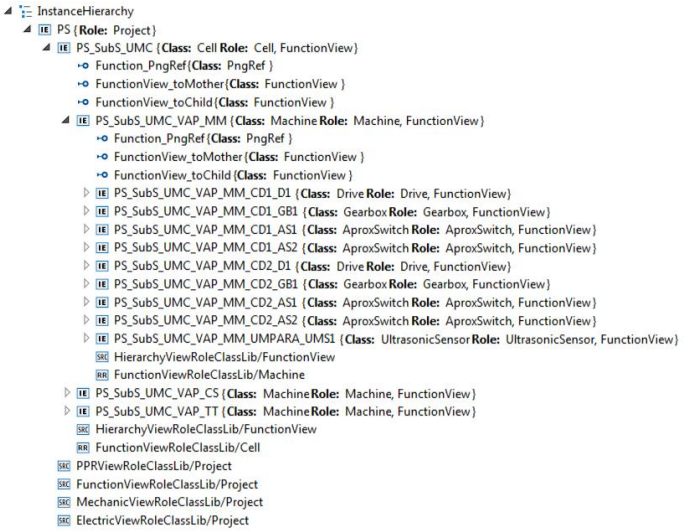
apl. Prof. Dr.- Ing. habil. A. Lüder

Step 1a: Functional engieering

```
<HO_FS_ANL id="noID">
 <PARA_PXNAME>PS</PARA_PXNAME>
 <PARA_PXTEXT>Demonstration agend based production system</PARA_PXTEXT>
 <HO_FS_TAN id="noID">
  <PARA_PXNAME>PS_SubS</PARA_PXNAME>
  <PARA_PXTEXT>Value chain within production system</PARA_PXTEXT>
  <HO_FS_BER id="noID">
   <PARA_PXNAME>PS_SubS_UMC</PARA_PXNAME>
   <PARA_PXTEXT>Ultrasonic Masurement Cell</PARA_PXTEXT>
   <HO_FS_TBR id="noID">
    <PARA_PXNAME>PS_SubS_UMC_VAP</PARA_PXNAME>
    <PARA_PXTEXT>Part of Cell</PARA_PXTEXT>
    <HO_FS_MAS id="noID">
     <PARA_PXNAME>PS_SubS_UMC_VAP_MM</PARA_PXNAME>
     <PARA_PXTEXT>Measuring Machine</PARA_PXTEXT>
     <HO_FS_FGR id="noID">
      <PARA_PXNAME>PS_SubS_UMC_VAP_MM_CD1</PARA_PXNAME>
      <PARA_PXTEXT>Drive Chain</PARA_PXTEXT>
      <HO_FS_ANT id="noID">
       <PARA_PXNAME>PS_SubS_UMC_VAP_MM_CD1_D1</PARA_PXNAME>
       <PARA_PXTEXT>Drive</PARA_PXTEXT>
       <PARA_CODE>DRIVE</PARA_CODE>
       <PARA_CODE_TEXT>DC motor</PARA_CODE_TEXT>
       <PARA_ZZ_LIEFERA>Some Drive Provider</PARA_ZZ_LIEFERA>
       <PARA_ZZ_TYPBEZ/>
       <PARA_MFRPN>ABC 123 DEF 456.78 GH </PARA_MFRPN>
       <PARA_SUPPLY_LINE>10</PARA_SUPPLY_LINE>
       <PARA_GROES>22</PARA_GROES>
       <PARA_CURRNT>220</PARA_CURRNT>
       <PARA_SPEED_HIGH>1000 </PARA_SPEED_HIGH>
      </HO_FS_ANT>
      <HO_FS_FEG id="noID">
      <HO_FS_FEG id="noID">
      <HO_FS_FEG id="noID">
     </HO_FS_FGR>
     <HO_FS_FGR id="noID">
     <HO_FS_FGR id="noID">
    </HO_FS_MAS>
    <HO_FS_MAS id="noID">
    <HO_FS_MAS id="noID">
   </HO_FS_TBR>
```

Step 1c: Setup of overall data model





Step 1b: Translation to AutomationML

Use process exploiting AutomationML

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Step 2a: Mechanical engieering

```
<Assembly name="14100000" simprep="" DescrDE="" DescrEN="ProductionSystem">
  <Assembly name="14101000" DescrDE="Zelle" DescrEN="Cell" position="0.0000 -836.0000 0.0000" rotation="0.00 180.00 0.00" motionplaced="false" durchmesser="5">
    <Assembly name="14101100" DescrDE="Maschine" DescrEN="Machine" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
      <Assembly name="14101110" DescrDE="Antriebsstrang" DescrEN="Drive chain" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
        <Assembly name="14101111" DescrDE="Antrieb" DescrEN="Drive" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101112" DescrDE="Getriebe" DescrEN="Gear box" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101113" DescrDE="Endschalter" DescrEN="Limiting switch" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101114" DescrDE="Endschalter" DescrEN="Limiting switch" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
      </Assembly>
      <Assembly name="14101120" DescrDE="Antriebsstrang" DescrEN="Drive chain" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
        <Assembly name="14101121" DescrDE="Antrieb" DescrEN="Drive" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101122" DescrDE="Getriebe" DescrEN="Gear box" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101123" DescrDE="Endschalter" DescrEN="Limiting switch" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101124" DescrDE="Endschalter" DescrEN="Limiting switch" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
      </Assembly>
      <Assembly name="14101130" DescrDE="Messeinheit" DescrEN="Measurement unit" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
        <Assembly name="14101131" DescrDE="Ultraschallsensor" DescrEN="Ultrasonic sensor" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
      </Assembly>
      <Assembly name="14101140" DescrDE="Gestell" DescrEN="Frame" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
    </Assembly>
    <Assembly name="14101200" DescrDE="Band" DescrEN="Conveyer" position="0.0000 215.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
      <Assembly name="14101210" DescrDE="Antriebsstrang" DescrEN="Drive chain" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
        <Assembly name="14101211" DescrDE="Antrieb" DescrEN="Drive" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
        <Assembly name="14101212" DescrDE="Getriebe" DescrEN="Gear box" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
      </Assembly>
      <Assembly name="14101220" DescrDE="Positionserkennung" DescrEN="Position detection" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
        <Assembly name="14101221" DescrDE="Positionssensor" DescrEN="Position sensor" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
      </Assembly>
      <Assembly name="14101230" DescrDE="Gestell" DescrEN="Frame" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
    </Assembly>
    <Assembly name="14101300" DescrDE="Drehtisch" DescrEN="Turntable" position="640.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
      <Assembly name="14101310" DescrDE="Antriebsstrang" DescrEN="Drive chain" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
      <Assembly name="14101320" DescrDE="Antriebsstrang" DescrEN="Drive chain" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
      <Assembly name="14101330" DescrDE="Positionserkennung" DescrEN="Position detection" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false">
      <Assembly name="14101340" DescrDE="Gestell" DescrEN="Frame" position="0.0000 0.0000 0.0000" rotation="0.00 0.00 0.00" motionplaced="false"/>
    </Assembly>
  </Assembly>
</Assembly>
```

Step 2b: Translation to AutomationML

Step 2c: Integration in overall data model

Use process exploiting AutomationML

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

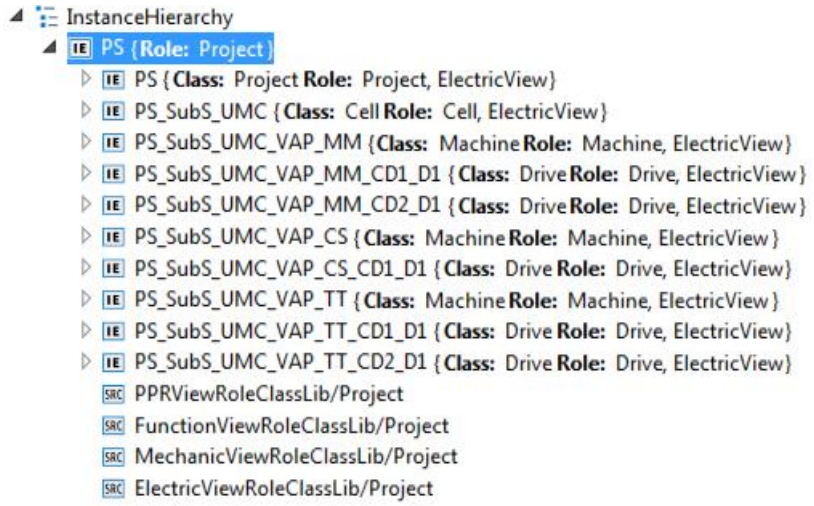
apl. Prof. Dr.- Ing. habil. A. Lüder

Step 3a: Electrical engieering

Functionally relevant	OU	User	Project internal ID	Denomination -German-	Denomination -English-	Quantity	Axle ø	Velocity/ rotational speed/ cycle time	Nominal voltage	Nominal current
Funktions-relevant	OE	Nutzer	Projekt interne ID	Benennung -deutsch-	Benennung -englisch-	Anzahl	Achs-ø	Geschwindig-keit / Drehzahl / Zykluszeit	Nominalspannung	Nominalstrom
				Einheit/ Formatierungsvorgabe:		2-stellig	mm	---	V	A
			PS	**Production system**						
			PS_SubS_UMC	**Cell_02**	**Measurement system**					
			PS_SubS_UMC_VAP_MM	**Machine**	**Measurement machine**					
x	AFR	MUTO	PS_SubS_UMC_VAP_MM_CD1_D1	Drive01		1	200	200	24	0.2
x	AFR	MUTO	PS_SubS_UMC_VAP_MM_CD2_D1	Drive02		1	200	200	24	0.2
			PS_SubS_UMC_VAP_CS	**Conveyer**						
x	AFR	MUTO	PS_SubS_UMC_VAP_CS_CD1_D1	Drive01		1	200	200	24	0.1
			PS_SubS_UMC_VAP_TT	**Turntable**						
x	AFR	MUTO	PS_SubS_UMC_VAP_TT_CD1_D1	Drive01		1	200	200	24	0.2
x	AFR	MUTO	PS_SubS_UMC_VAP_TT_CD2_D1	Drive02		1	200	100	24	0.1

Step 3c: Integration in overall data model

Step 3b: Translation to AutomationML



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG MB

apl. Prof. Dr.- Ing. habil. A. Lüder

Use process exploiting AutomationML

- Method assists responsible staff (*data model integrator*) within identifying information to be exchanged in engineering networks
- Efficiency and speed of data logistics can be improved with automated data integration without change in the engineering habit of the involved engineers
- Implementation of a data logistics can be conducted incrementally.

- Main limitation: Need of a *data model integrator* as a new role within an engineering network that has to have an overview on the different involved engineering disciplines

Reached results