

RUB

Herzlich Willkommen!



LEHRSTUHL FÜR
PRODUKTIONSSYSTEME



LPS LERN- UND
FORSCHUNGSFABRIK

 DIAMOND

AML

AUTOMATION MARKUP LANGUAGE

Schulungsunterlagen für
Einsteiger und Fortgeschrittene

Grundlagen Schulung AutomationML

Aufbau der Schulung

- Dieser Abschnitt richtet sich an Personen ohne oder mit geringen Vorkenntnissen im Bereich AutomationML.
- Vermittelt werden die grundlegenden Prinzipien der Systemmodellierung mit dem AML-Editor sowie die Basis der Objektmodellierung in AML.

Über die Schulung

- Die Schulung kombiniert kurze Theorieeinheiten mit praxisnahen Projektbeispielen des Lehrstuhls für Produktionssysteme (LPS) der Ruhr-Universität Bochum (RUB).
- Ziel ist es, am Ende der Schulung ein konsistentes, realitätsnahes AML-Modell auf Basis der Übungsbeispiele mithilfe des AML-Editors zu erstellen.

Inhaltsübersicht

Einführung

Grundlagen und Definitionen

CAEX/OOP

Der AML-Editor

AML-Trainingsaufgabe

Biepsiellösung

Lessons Learned & Ausblick

Was ist AutomationML ?

<AutomationML/>
Next Level Engineering



Quelle: www.automationml.org

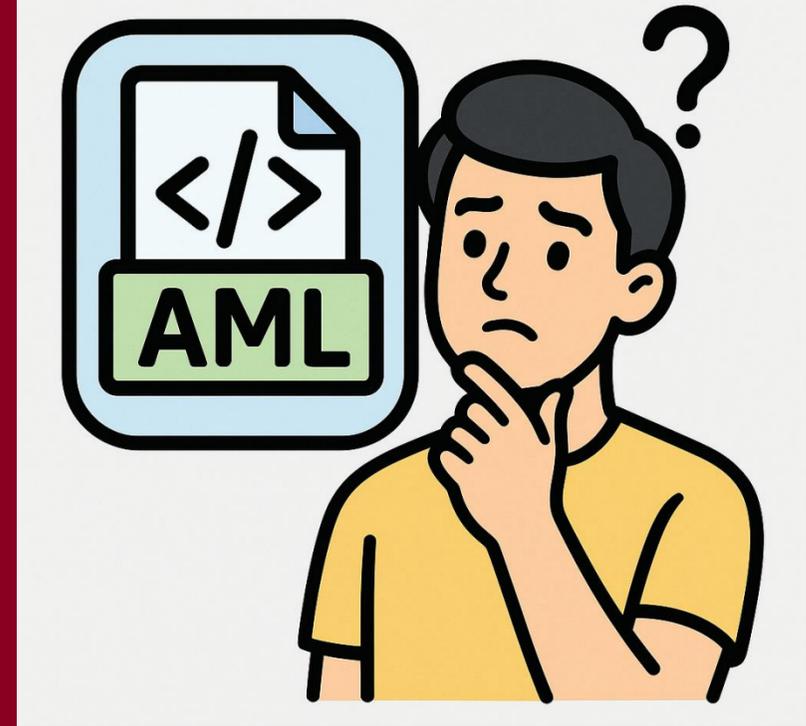
Einführung

Was ist AutomationML (AML) und wofür wird es verwendet?

- AML ist eine flexible, standardisierte Datenmodellierungssprache und ein Dateiformat zur Speicherung und dem Austausch von Informationen zwischen Engineering-Werkzeugen (IEC 62714).
- AML ist ein anwendungsneutraler Ansatz, um Daten in einem maschinenverständlichen Objektmodell abzubilden
- Basiert auf XML (Extensible Markup Language).

Einsatzbereiche von AML

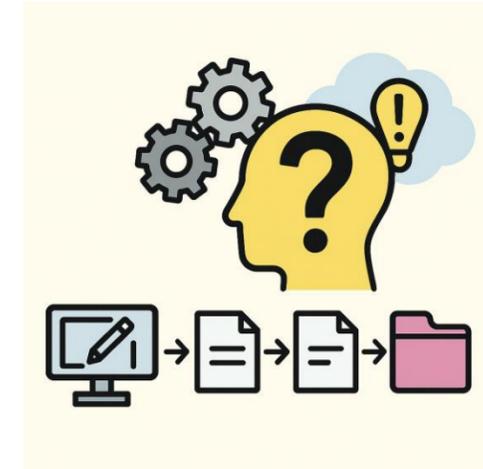
- Informationstransport Prozesse zwischen Engineering Tools
- Einsatzmöglichkeiten für bilateralen oder multilateralen Datenaustausch zeitlich veränderlicher Daten



Herausforderungen aus der Datensicht in Engineering Prozessen

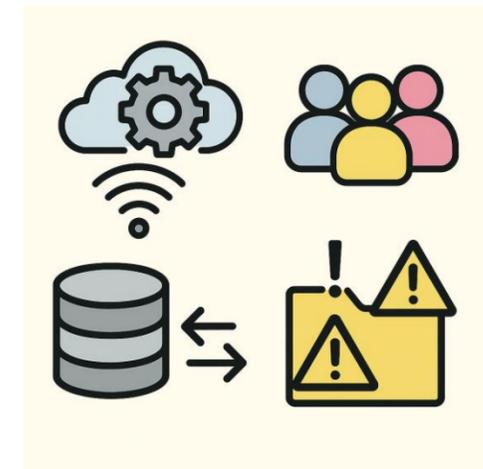
Allgemeine Herausforderungen

- Zentrale Herausforderung ist die Synchronisation stetig veränderlicher Informationen entlang einer Engineering Werkzeugkette



Herausforderungen werkzeugorientierter Ansätze

- Steigende Informationsdichte aufgrund von verstärktem Einsatz von IoT
- Hohe Interdisziplinarität mit starker Diversität der spezifischen Entwicklungswerkzeuge
- Starke Arbeitsteilung in einem auf Phasen basierenden Planungsprozess führt zu:
 - hoher Datendivergenz
 - steigender Dateninkonsistenz



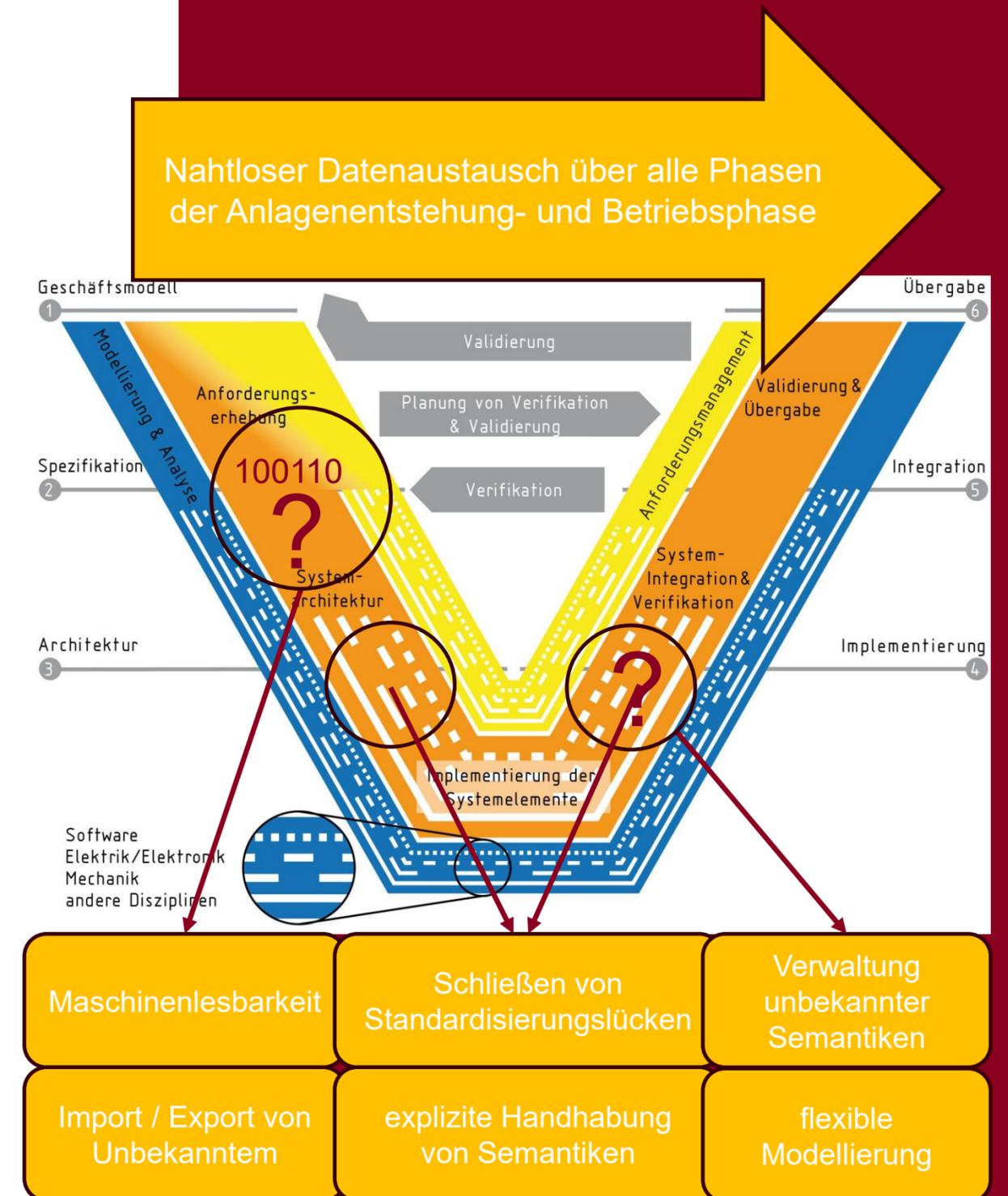
Ziele und Teilziele beim Einsatz von AML

Hauptziel

- Nahtloser, automatisierter Datenaustausch über alle Phasen des Engineerings zwischen verschiedenen Software-Tools

Teilziele

- Maschinen- & menschenlesbare Daten
- Schließen von Standardisierungslücken
- Verwaltung von Im-/Export unbekannter Datensemantiken
- Import/Export unbekannter Daten handhabbar machen
- Flexible Modellierung statt starrer Formate
- Unbekannte Semantiken explizit verwalten

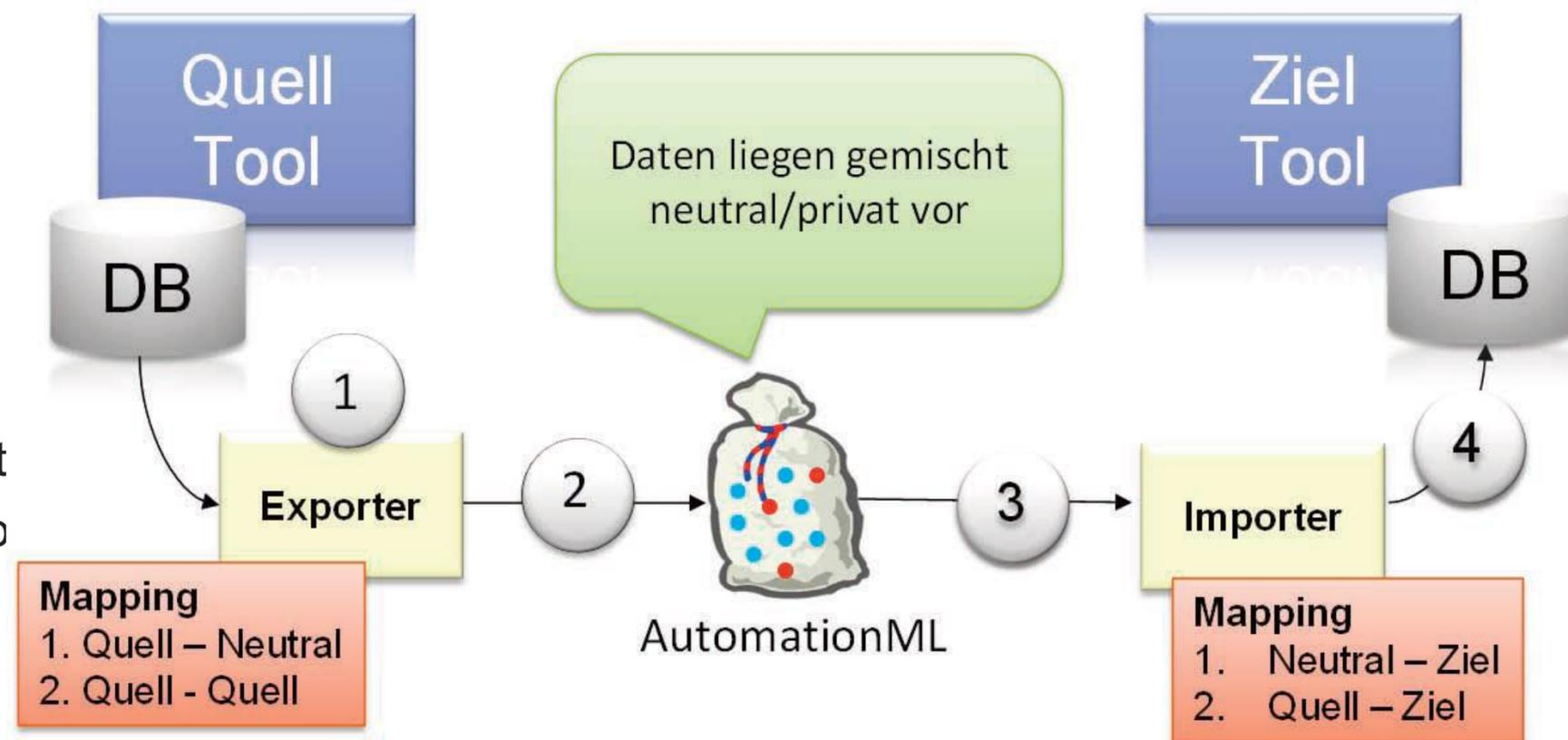


Quelle: V-Modell aus VDI/VDE 2206

Merkmale von AML

Charakteristika von AML

- Metaformat anstatt explizites Datenformat
- objektorientierter Datenmodellierungsansatz
- Trennung von Semantik und Syntax
- Einsatz bestehender Standards (XML)
- Identifikation von Semantik anhand von ..
 - Quellwerkzeugdetektion aus den proprietären Daten
 - Steigerung der Nachvollziehbarkeit durch Quelltools
- Hohe Variabilität bei der Abstraktion



Quelle: A. Lüder et al. AML-Whitepaper veröffentlicht in SPS Magazin



Vor- und Nachteile von AML

Vorteile:

- Einsatz von universellen Datenmodellierungsansätzen und Sprachen führt zu Zeitersparnis im Engineering Prozess
- Effizienzsteigerung durch Wiederverwendung bereits vorhandener Informationen
- Reduktion des Bedarfs an komplexen Import/Exporter Schnittstellen

Nachteile:

- Initialer Aufwand bei der Einführung kann je nach Grad der bereits vorhandenen Standardisierung von Informationen hoch ausfallen
- Fokus liegt auf Engineering Werkzeugen, jedoch keine Betrachtung des gesamten Lebenszyklus eines Produktes

Inhaltsübersicht

Einführung

Grundlagen und Definitionen

XML/CAEX/OOP

Der AML-Editor

AML-Trainingsaufgabe

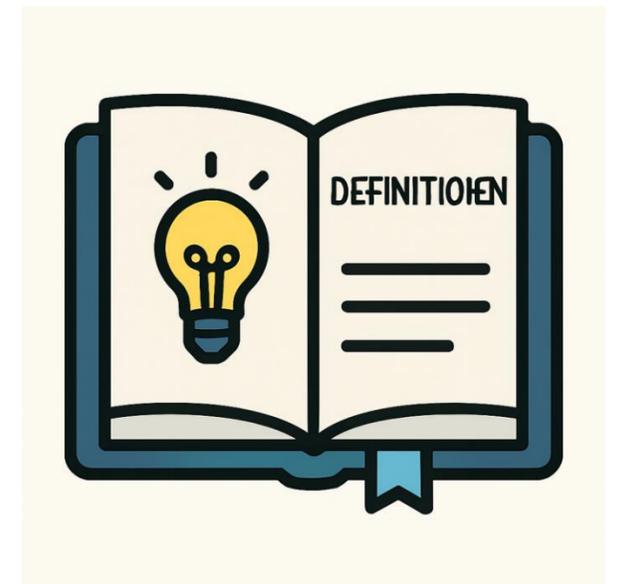
Beispiellösung

Lessons Learned & Ausblick

Allgemeine Definitionen aus OOP/CAEX und AML

Definitionen und Begrifflichkeiten:

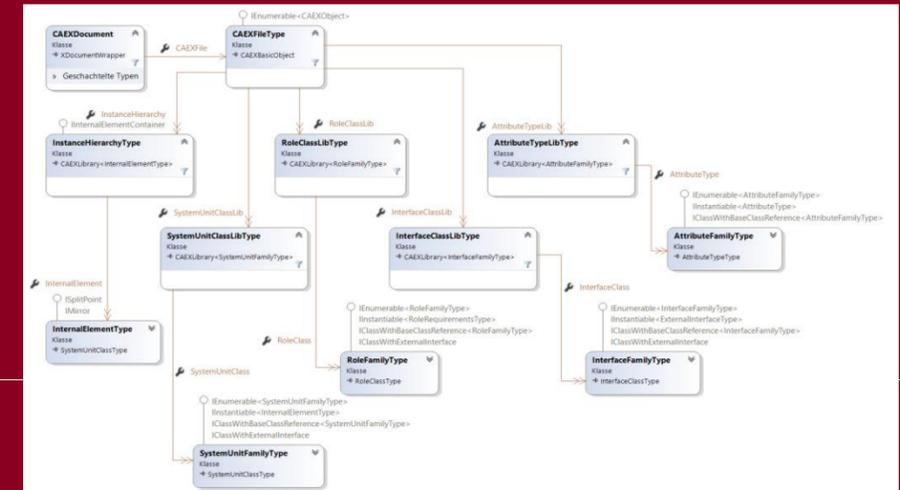
- Daten = Durch Beobachtung/Messung gewonnene unverarbeitete Zahlenwerte oder Objekte.
- Syntax = Beschreibungsform der Zusammenführungsregeln bei der Notation von Daten.
- Semantik = von außen definierter Bedeutung und Interpretationsregel eines Datenelements, die auf Vereinbarungen von Mitwissern basiert.
- Informationen = bilden die Grundlage für Entscheidungen und sind interpretierte Daten deren Kontext und Bedeutung bekannt ist.
- Datenmodell = Ist eine strukturierte Zusammenfassung, zusammengehöriger Daten und Relationen, im Bezug auf einen spezifischen Anwendungszweck
- Metamodell = beschreibt die Sprachebene, mit der die Datenmodelle erzeugt werden können. D.h. es umfasst zusätzlich zur Datenmodellinstanz, die Regeln, Strukturen und Konzepte zur Erstellung einzelner Instanzen.



CAEX = Computer Aided Engineering Exchange

OOP = objektorientierte Programmierung

Gemeinsamkeiten/ Unterschiede von AML / UML / SysML



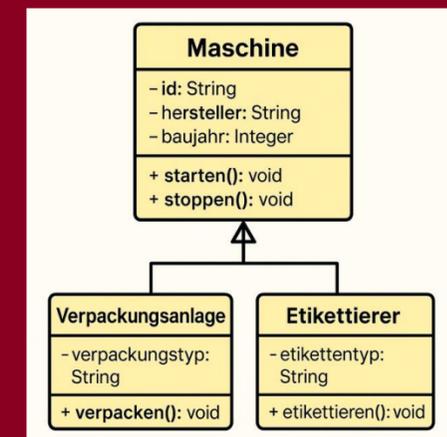
Gemeinsamkeiten

- SysML und UML sind grafische Beschreibungsformen, zur Förderung der Verständlichkeit einer Sache für den Mensch
- AML, UML und SysML zeichnen sich dadurch aus, dass sie durch Diagramme, und den Einsatz bestehender Modelle verständnisfördernd für den Menschen sein können

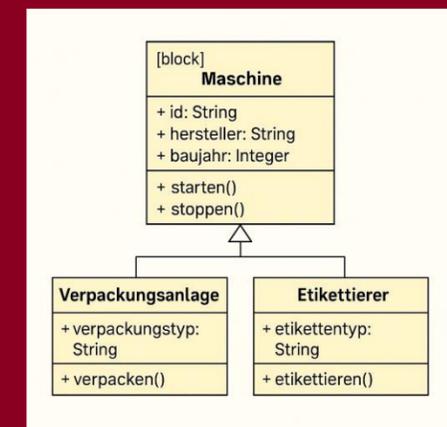
Unterschiede

- AML soll Maschinenverständlichkeit erhöhen
- AML ist Objektbeschreibungssprache, die bestehende zu überführende, proprietäre Modelle umfasst.

UML



SysML



Lernkontrolle

Kurzfragen

1. Was versteht man unter AML und wofür wird es verwendet?
2. Erläutern Sie welche Herausforderungen in digitalisierten Engineering Prozessen entstehen können. Welche Rolle spielt AML in diesem Zusammenhang?
3. Nennen Sie 4 Ziele, die bei dem Einsatz von AML adressiert werden.
4. Was zeichnet AML aus? Beschreiben Sie 4 Charakteristika von AML.
5. Nennen Sie jeweils 2 Vor- und Nachteile, die mit dem Einsatz von AML einhergehen und erläutern Sie diese Anhand eines geeigneten Beispiels.
6. Was unterscheidet AML von SYSML und UML?
7. Welche Gemeinsamkeiten weisen diese Beschreibungsmittel auf?
8. Definieren Sie die Begriffe Daten, Semantik und Datenmodell.
9. Was unterscheidet ein Datenmodell von einem Metadatenmodell? Welche Gemeinsamkeiten weisen beide Modellarten auf?

Inhaltsübersicht

Einführung

Grundlagen und Definitionen

XML/CAEX/OOP

Der AML-Editor

AML-Trainingsaufgabe

Beispiellösung

Lessons Learned & Ausblick

```

<?xml version="1.0" encoding="UTF-8"?>
<CAEXFile xmlns="http://www.dke.de/CAEX" SchemaVersion="3.0">
  <InstanceHierarchy Name="ExamplePlant">
    <InternalElement Name="Machine01" ID="m01">
      <Attribute Name="Manufacturer">
        <Value>ExampleCorp</Value>
      </Attribute>
      <Attribute Name="ModelNumber">
        <Value>XC-200</Value>
      </Attribute>
    </InternalElement>
  </InstanceHierarchy>

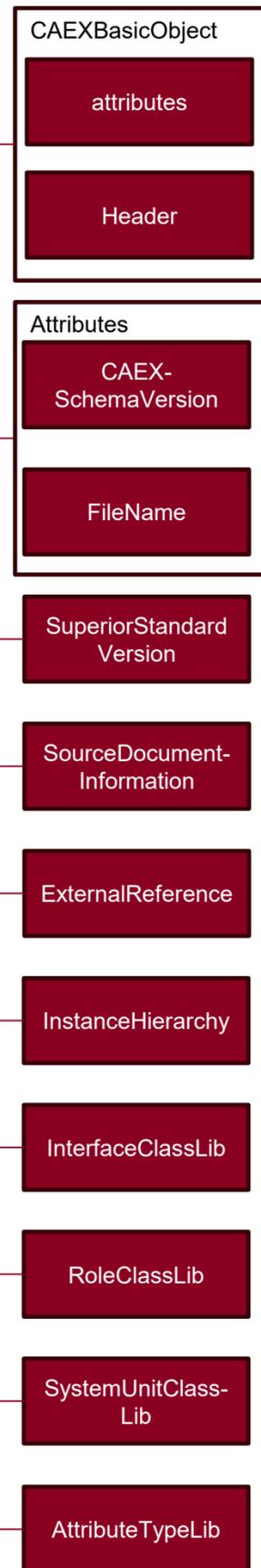
  <SystemUnitClassLib Name="StandardComponents">
    <SystemUnitClass Name="MachineBase">
      <Attribute Name="PowerRating" />
    </SystemUnitClass>
  </SystemUnitClassLib>

  <InterfaceClassLib Name="StandardInterfaces">
    <InterfaceClass Name="ElectricalInterface">
      <Attribute Name="Voltage" />
    </InterfaceClass>
  </InterfaceClassLib>
</CAEXFile>

```

Vertiefung XML in
Schulungsteil 2 sowie AML-
Engine in C#

CAEX-File



1 Das Metadatenmodell in CAEX

Aufbau des CAEX-Schemas

1. Elemente zur Modellierung von organisatorischen Informationen des AML-Dokuments
2. Wurzelknoten verschiedener Bibliotheken (Kernelement der AML basierten Objekt Modellierung)
3. External Reference ermöglicht Splitten von CAEX-Dokumenten, durch Verweis auf externe CAEX-Objekte

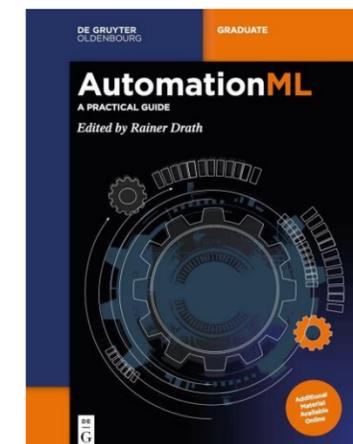
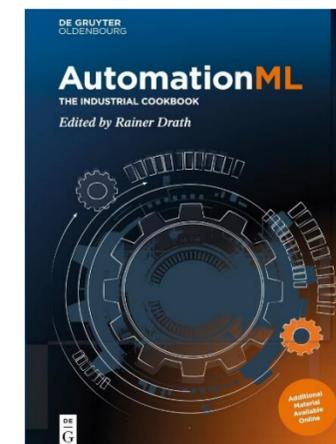
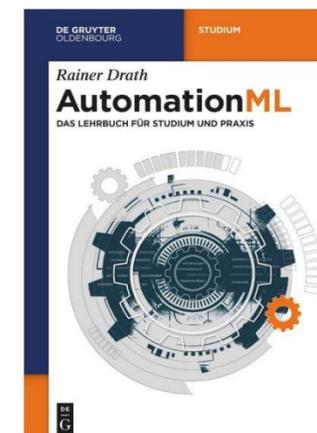
2 Metadatenmodell

- Die Persistenz im CEAX-Schema wiederum basiert auf XML
- Ermöglicht automatische Prüfung auf Fehler
- Schemadatei beinhaltet Blaupause

Das CAEX-Datenmodell

CAEX-Schema als Ausgangspunkt eines AML-Dokuments

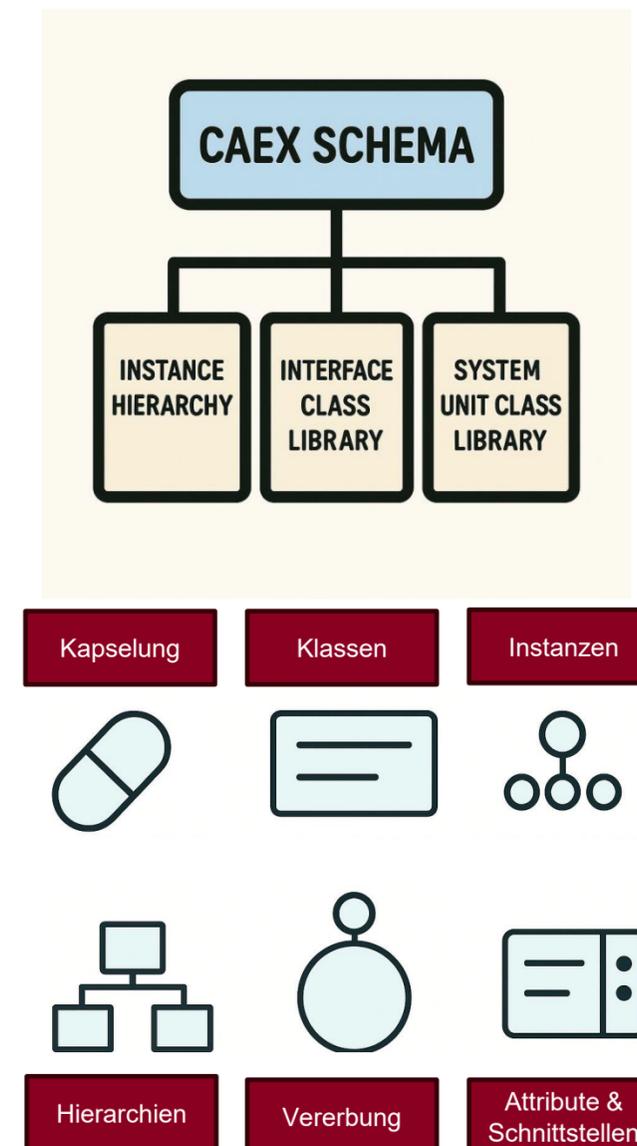
- Das AML-Datenmodell setzt sich zusammen aus der *.aml Datei und den in ihr definierten Regeln, sowie aus dem Aufbau und Regeln des CAEX-Schemas.
- Der CAEX- und AML-Quellcode wird durch XML beschrieben.
- Das CAEX-Schema wird über den AML-Editor oder mithilfe der AML-Engine gespeichert und ermöglicht die .aml Dokumentvalidierung.
- Weiterführende Litertatur:
<https://www.automationml.org/about-automationml/publications/amlbook/>



CAEX Sprachelemente und XML als Basis von AML-Modellen

CAEX als Grundlage für AML

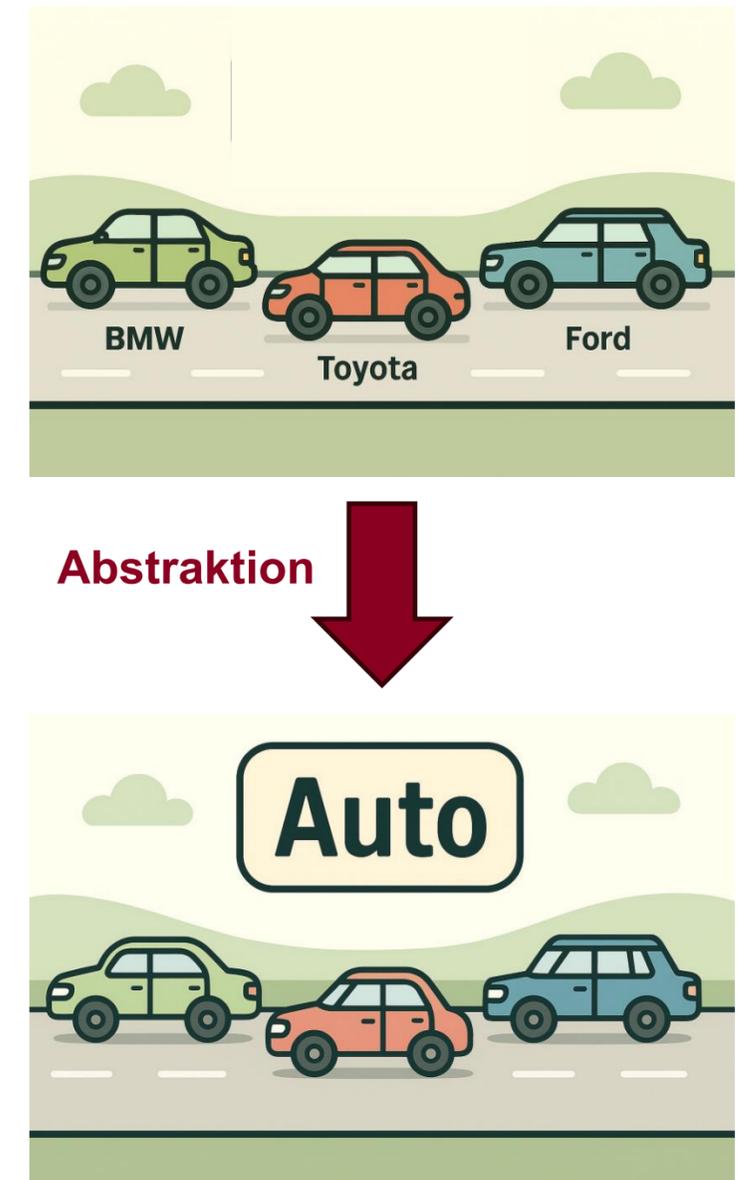
- Modellierung in AML basiert auf dem Aufbau des CAEX-Schemas,
- Für CEAX und AML gelten gängige Grundkonzepte aus der objektorientierten Programmierung:
 - Kapselung
 - Klassen & Klassenbibliotheken
 - Instanzen und Instanzierung
 - Hierarchien und Vererbung
 - Attribute
 - Schnittstellen



Das Rollenkonzept in CAEX und AML

Abstraktion realer Objekte

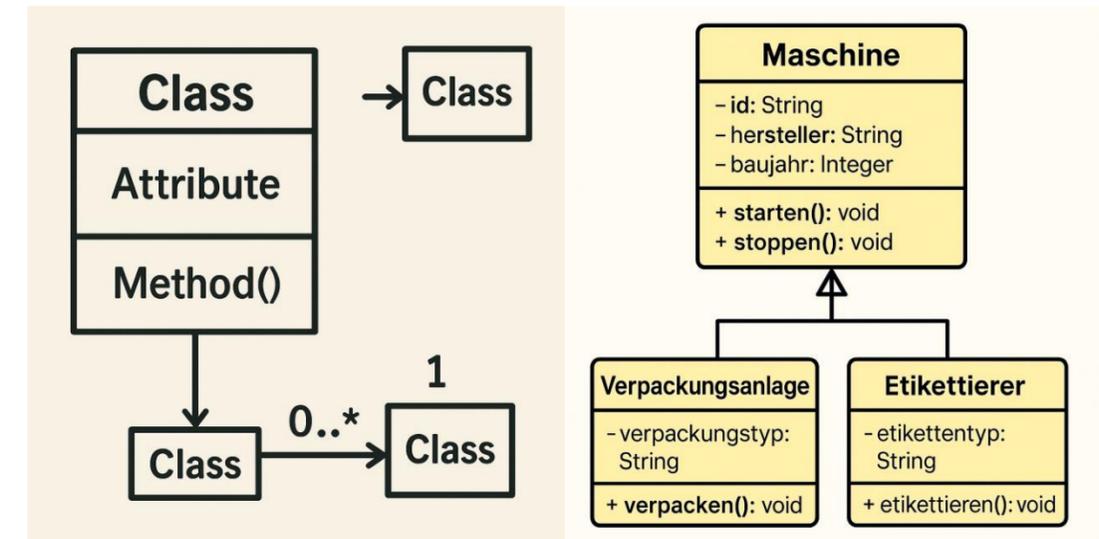
- Jedes Objekt innerhalb eines Systems kann abstrahiert und ohne Spezifikation der technischen Details modelliert werden
- Objekte können mehrere Rollen gleichzeitig haben
- Engineering Prozesse in der Systementwicklung
 - Denken in Rollen ermöglicht Modellierung komplexer Anlagen und Systeme bevor eine Komponentenauswahl stattfindet
 - Vereinfachung von Änderungs- und Versionsmanagement durch Abstrakte Platzhalter Objekte im Datenmodell (Rollen)



Modellierung und Analyse von Anlagen

Vorgehensweise bei der Anlagen Modellierung mit AML

- Systemanalyse zunächst unabhängig von AML-Struktur möglich
- Vorgehen folgt Methodik der objektorientierten Analyse
- Notationsmöglichkeiten bspw. via UML-Klassendiagramm oder tabellarisch

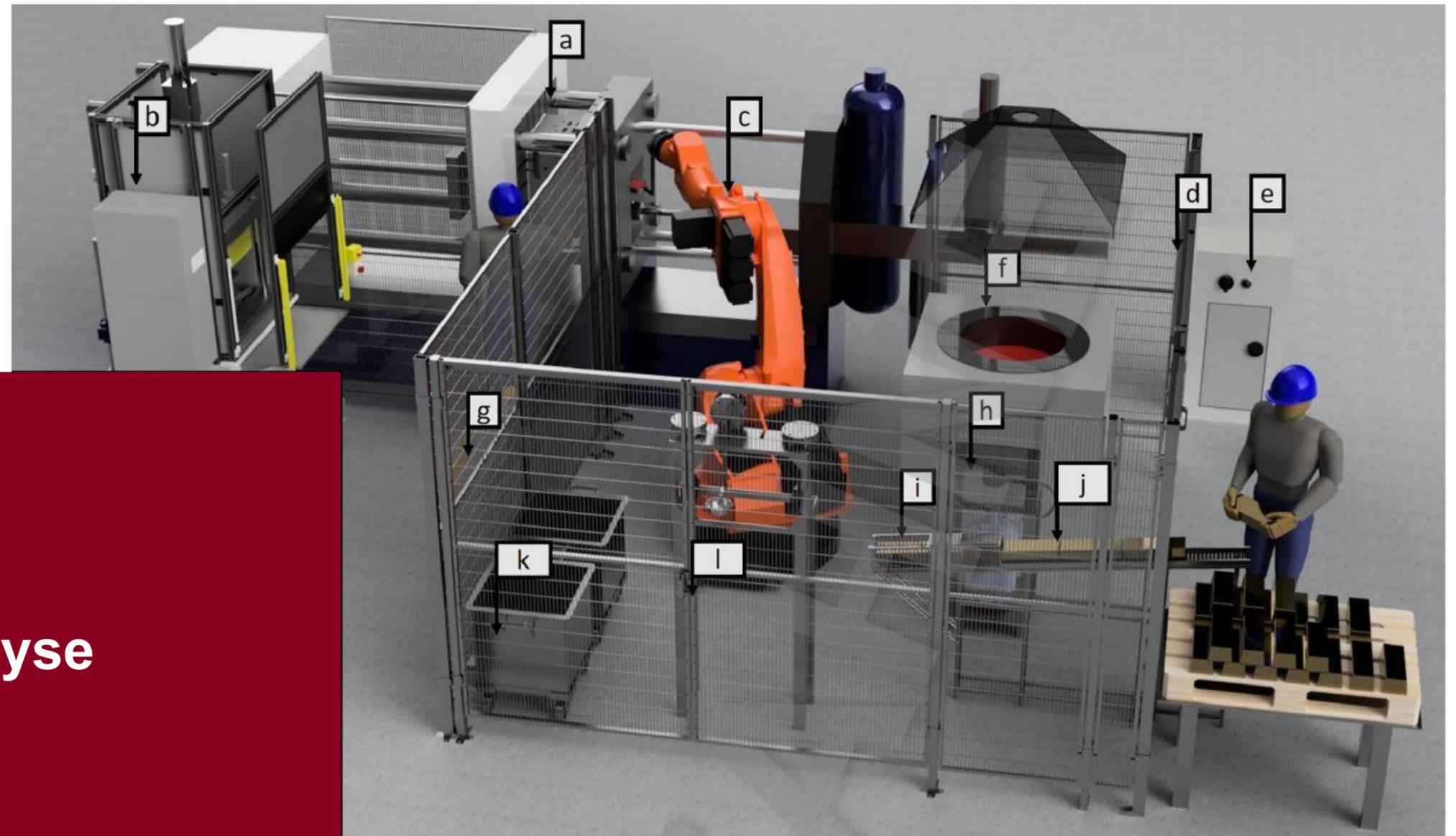


Vorgehensweise objektorientierte Analyse

Schritt:	Beschreibung	Abbildungsform in CAEX /AML	Beispiele
1	Identifikation der Systemakteure	InternalElement	konkrete Objekte bspw. Roboter, Endeffektor etc.
2	Abstraktion auf Funktion und Rolle der Akteure im Gesamtsystem	RoleClass	Roboter, Steuerung, Materialbereitstellung etc.
3	Identifikation der individuellen Eigenschaften	Attribute / AttributeType	Traglast, Reichweite, Preis ID etc.
4	Identifikation von Schnittstellen	InterfaceClass / ExternalInterface	USB, Digitaler In-/Output etc.
5	Identifikation von Anforderungen der Akteure	RoleRequirements	Max. Traglast, Mindestabstand, Volumstrom im Intervall zwischen x & y
6	Identifikation von Klassen, Eigenschaften und Schnittstellen der Akteure	SystemUnitClass / AttributeType / InterfaceClass	Robotertyp, Preisklasse, USB Typ C

Schulungsbeispiel: 1

HiRoCast-Produktionszelle



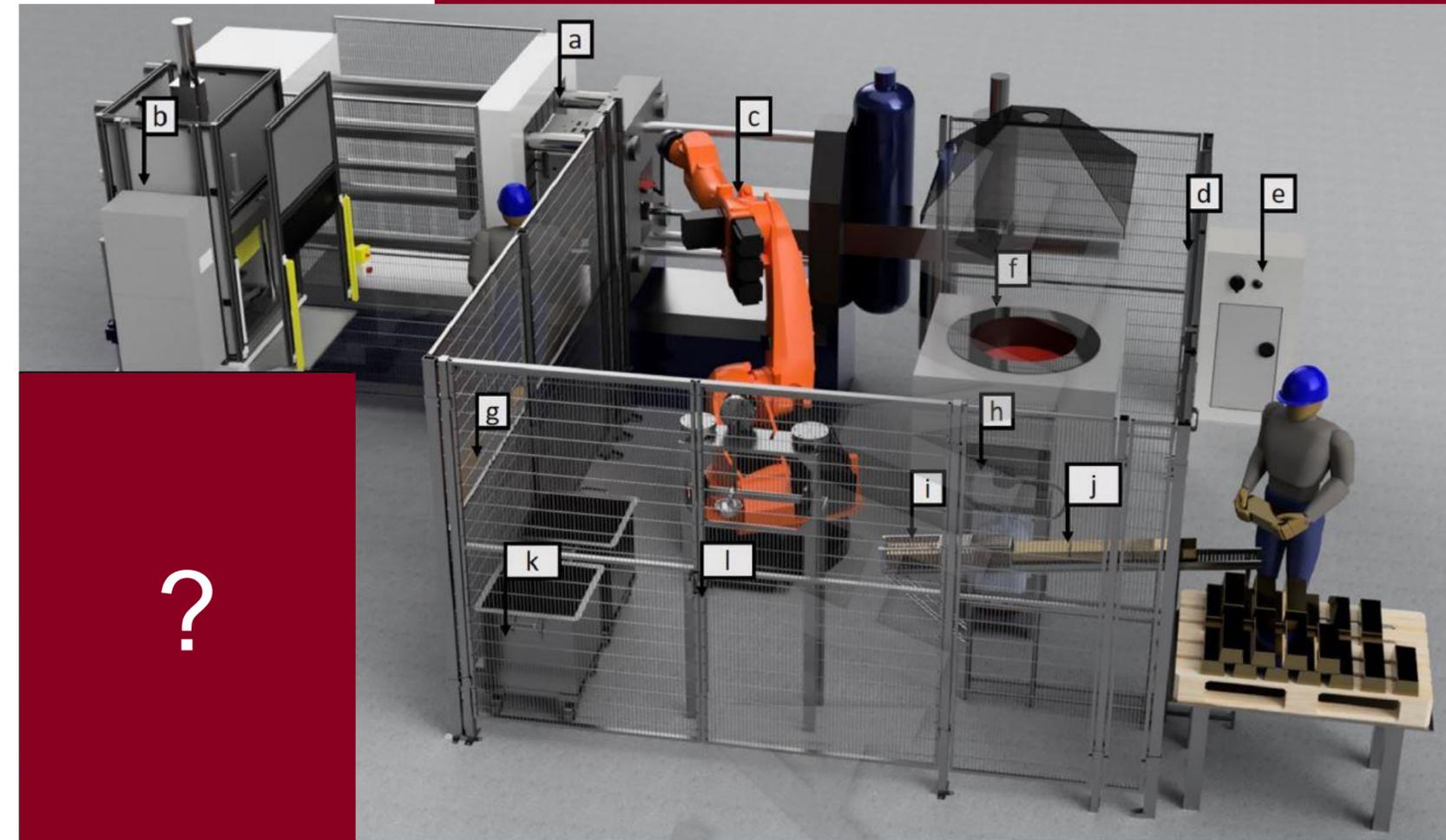
Systemaufbau und
objektorientierte Analyse

Bestimmen der Systemelemente

Eine kleine Aufwärmübung...

Die HiRoCast-Anlage besteht aus....

- Druckgussmaschine (Gießereiprozess)
- Abknackpresse (mechanische Station)
- Kuka Kr 200 (Roboter)
- Sicherheitszaun (Sicherheitsvorrichtung)
- Sicherheitstüren (Sicherheitsvorrichtung)
- Steuergeräte/Peripherie (Steuerungseinheit)
- Leitsystem / Beckhoff SPS (Steuerungseinheit)
- Robotersteuerung KRC 2 (Steuerungseinheit)
- Vorwärmstationen B & K (Gießereiprozess)
- Barrenmagazinen (Materialhandhabung)
- Schlackebehälter (Materialhandhabung)
- Nabertherm Ofen + Abdeckung (Gießereiprozess)
- Endeffektor/Gießkelle (Roboter)



Lernkontrolle

Kurzfragen

1. Wie hängen XML, AML und das CEAX-Schema zusammen?
2. Wodurch wird die Persistenz und automatisierbare Überprüfung in AML ermöglicht?
3. Was versteht man unter den Prinzipien Kapselung und Vererbung?
4. Nennen Sie 3 weitere Konzepte aus der OOP welche für AML ebenfalls gültig sind.
5. Erläutern Sie die 6 Schritte der objektorientierten Analyse eines Systems am Beispiel einer Kaffeemaschine.
6. Welche CAEX/AML Abbildungsform wird verwendet um die Funktion einer Systemkomponenten zu Modellieren?

Inhaltsübersicht

Einführung

Grundlagen und Definitionen

XML/CAEX/OOP

Der AML-Editor

AML-Trainingsaufgabe

Beispiellösung

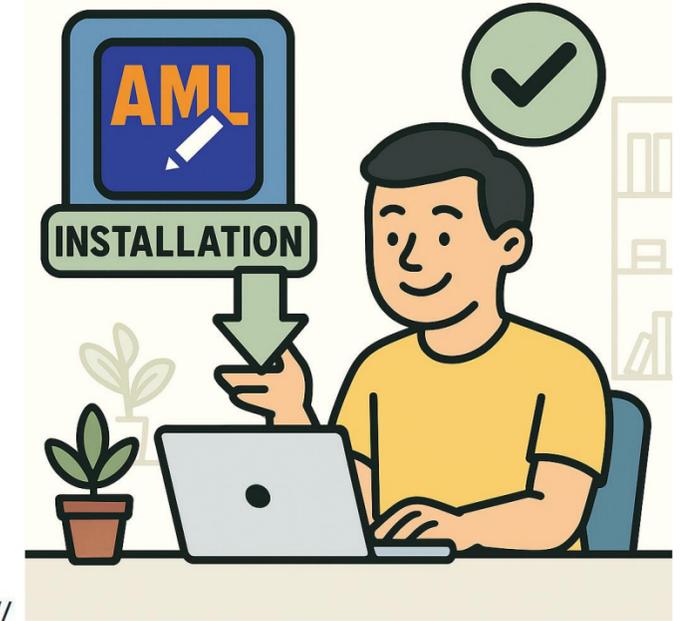
Lessons Learned & Ausblick

Installation und Download des AML-Editors

Downloads:

- Download Möglichkeiten des AML-Editors sind auf der offiziellen Seite des AML Ev. zu finden unter: <https://www.automationml.org/download-archive/>
- Alle im Rahmen dieser Schulung vorgestellten Inhalte sind mit Version 6.4 erstellt worden und lösbar
- Ebenfalls auf der Seite zu finden sind die AML-Standard Bibliotheken
- Zusätzlich sinnvoll ist die Dokumentvalidierung mithilfe der CAEX-Schemadatei: Speichern erfolgt aus dem AML-Editor heraus

Extras -> Save CAEX Schema to -> Speicherverzeichnis festlegen



Software and Tools //

AutomationML Editor V.6.4.
The latest version of the AutomationML Editor, which is provided under MIT license

AMLEngine 2.1 V.1.7.0
The AutomationML Engine supporting CAEX2.15 and CAEX3.0 and is distributed under MIT licence

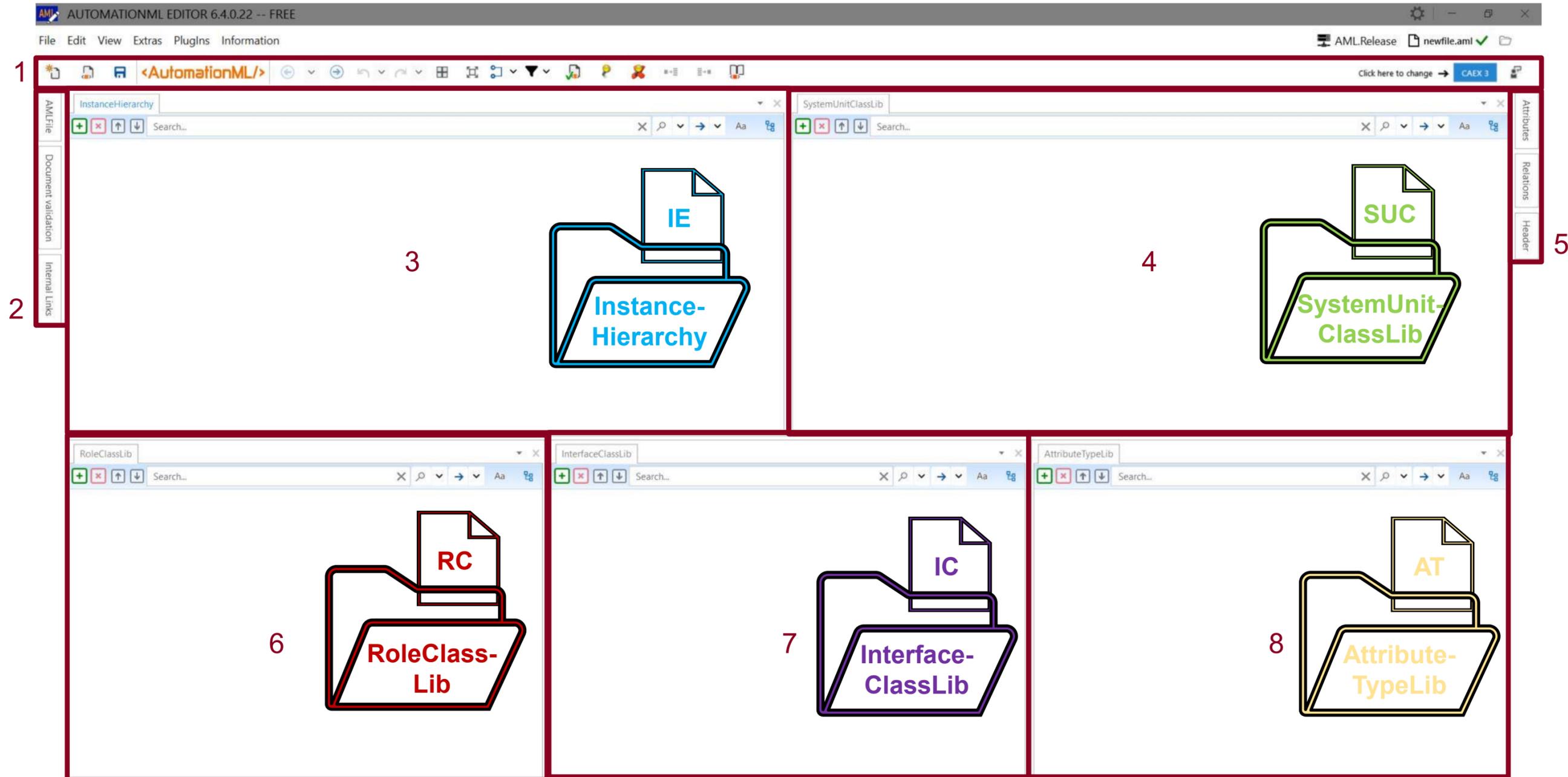
>> Download Archive //

Standard 1 download found.

Standard AutomationML communication libraries
The standard AutomationML communication libraries according to Whitepaper Part 5

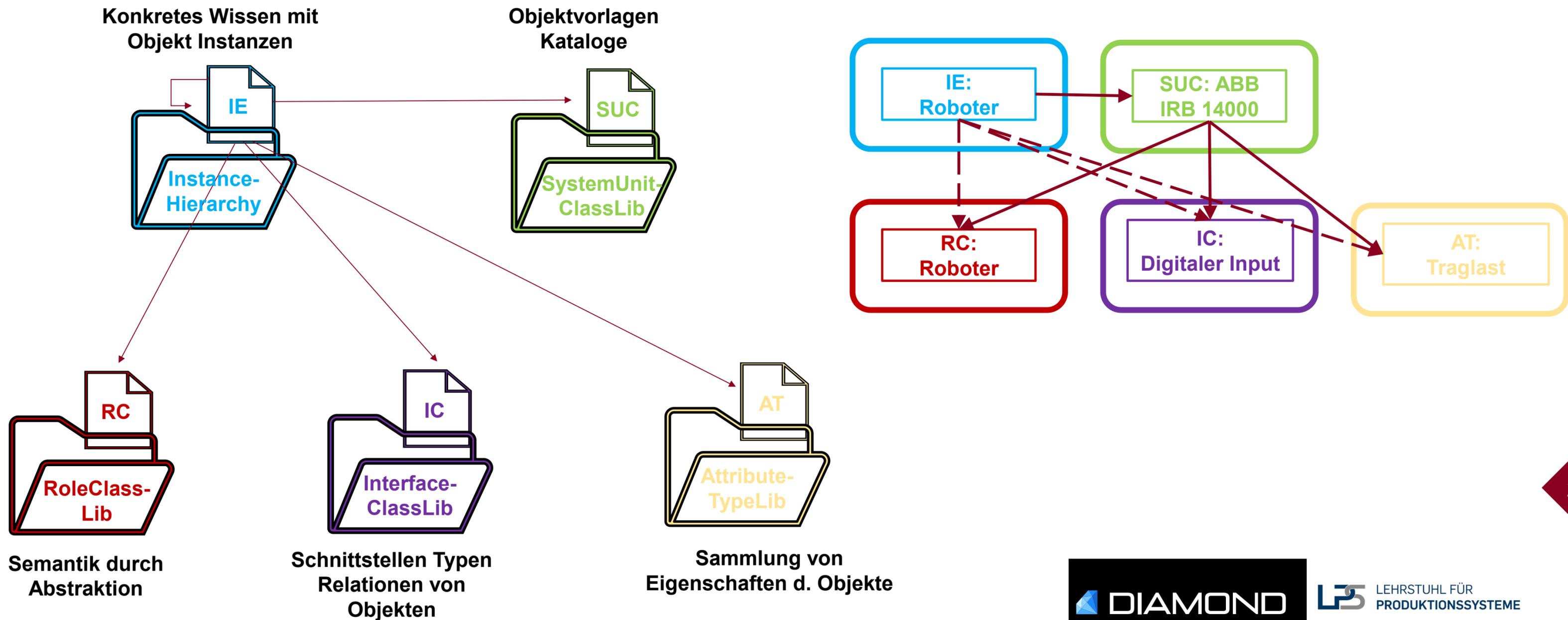


Die Benutzungsoberfläche des AML-Editors



Modellierung in CAEX/AML

Minimalbeispiel eines ABB IRB 1400



Allgemeine Hinweise zur Arbeit mit dem AML-Editor

Hinweise:

- Jede Bibliothek kann über die Schaltelemente gleichermaßen editiert werden, wobei Instanzen oder Klassen der jeweiligen Kategorie erzeugt, gelöscht, verschoben oder gefiltert werden können

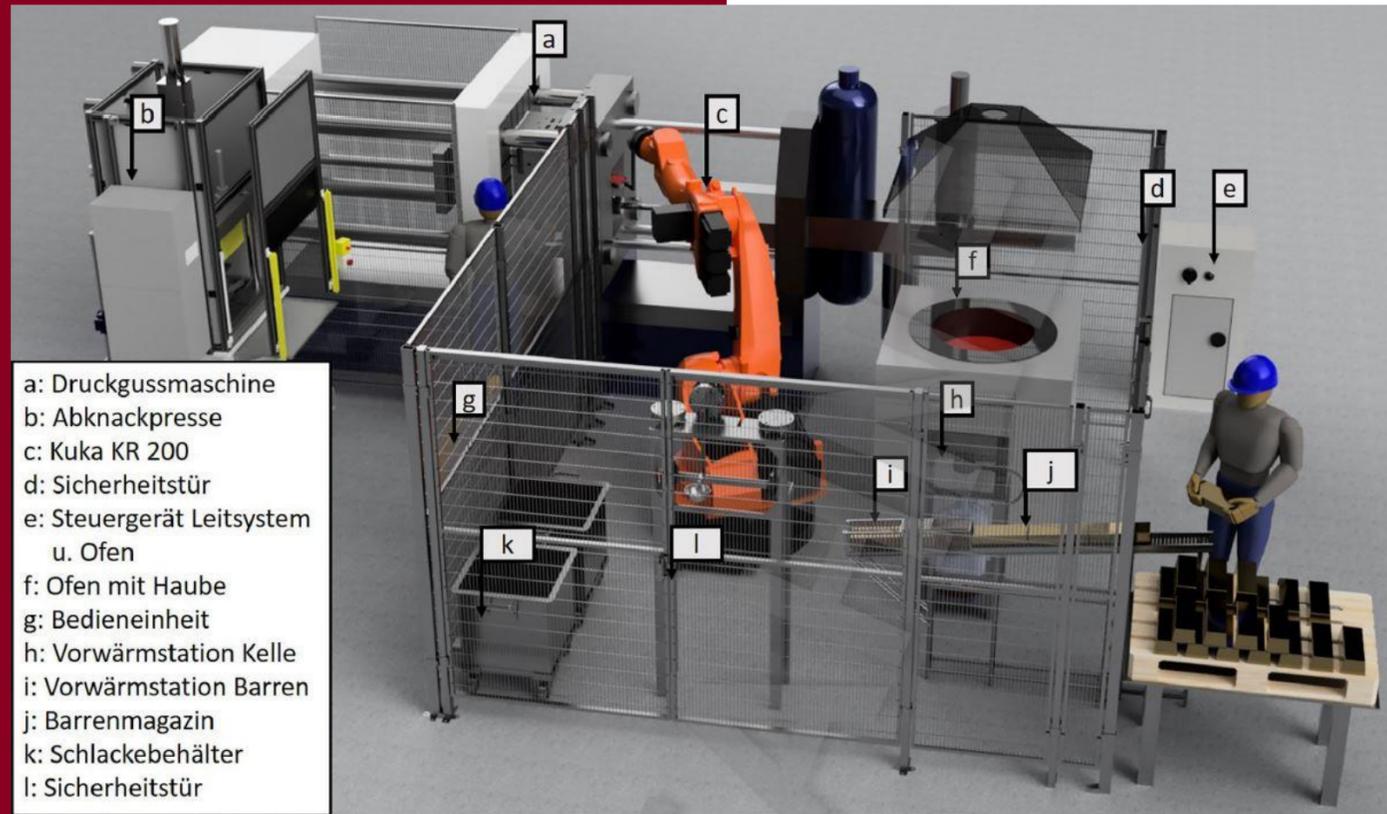


- Modellierungsreihenfolge Praxis:



- Reihenfolge in der Schulung zur Vermittlung der Grundlagen:





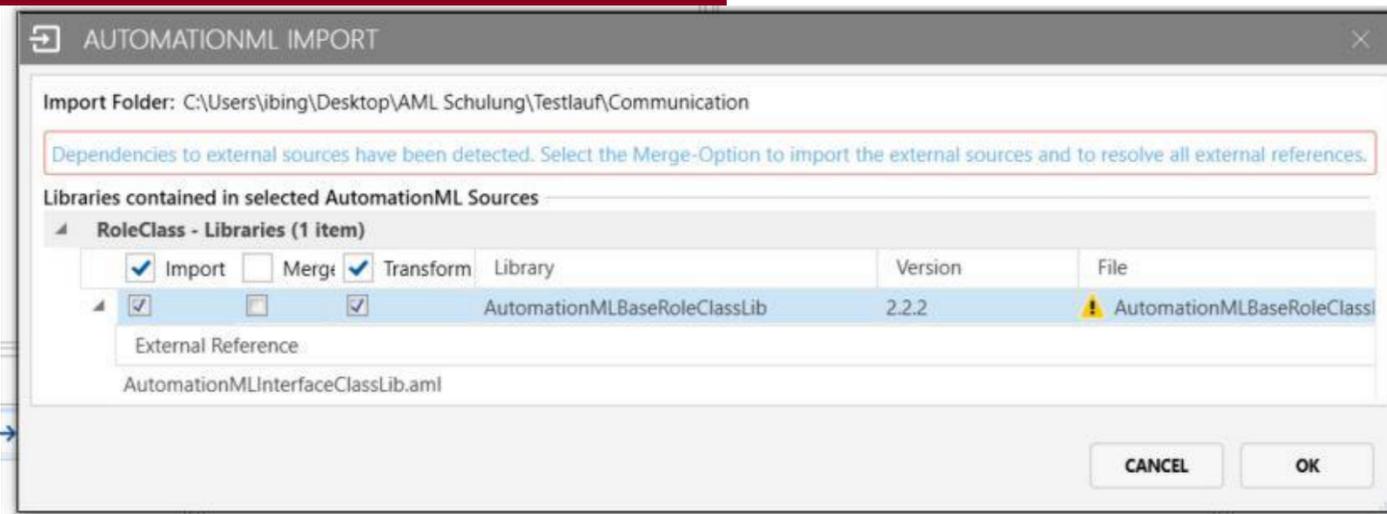
Erstellung eines AML-Metamodell's

Allg. Vorbereitung:

- Zu modellierenden Prozess definieren und System-Komponenten identifizieren (UML/OOA o.Ä.) ✓
- Installation AML-Editor ✓
- Download Allg. AML-Bibliotheken (optional) ✓

Erste Schritte im AML-Editor

- Einfügen benötigter Standard AML Bibliotheken
 File -> Import from files -> Ordner in dem AML-Standardbibliotheken abgelegt sind auswählen -> .aml Files auswählen (keine Mehrfachselektion möglich)
 - AutomationMLBaseRoleClassLib.aml
 - AutomationMLInterfaceClassLib.aml
- Anwählen der Import und Transform Option -> Bestätigen mit OK



Das leere AML-File mit importierten Standardklassen

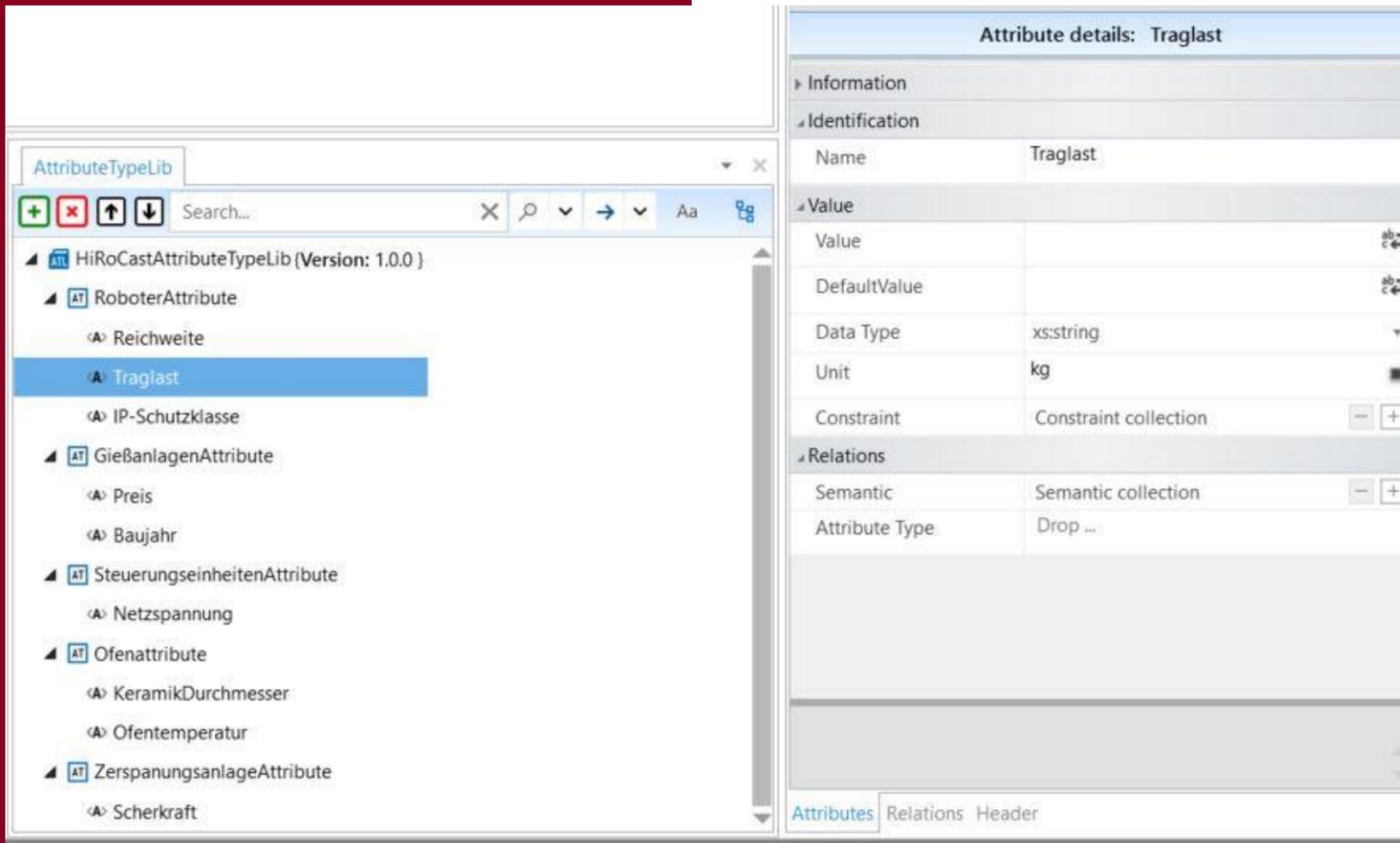
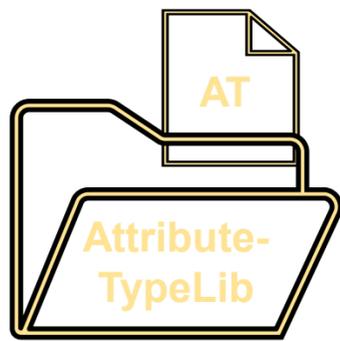
The screenshot displays a software interface with a project structure on the left and several empty windows on the right. The project structure is organized as follows:

- AutomationMLBaseRoleClassLib (Version: 2.2.2)
 - AutomationMLBaseRole
 - Group (Class: AutomationMLBaseRole)
 - Facet (Class: AutomationMLBaseRole)
 - Port (Class: AutomationMLBaseRole)
 - Resource (Class: AutomationMLBaseRole)
 - Product (Class: AutomationMLBaseRole)
 - Process (Class: AutomationMLBaseRole)
 - Structure (Class: AutomationMLBaseRole)
 - PropertySet (Class: AutomationMLBaseRole)
 - Frame (Class: AutomationMLBaseRole)

The right side of the interface features five empty windows, each with a search bar and navigation icons:

- InstanceHierarchy
- SystemUnitClassLib
- RoleClassLib
- InterfaceClassLib
- AttributeTypeLib

On the far right, there are three vertical tabs: Attributes, Relations, and Header.



AttributeType/Attribute

Die *AttributeTypeLib* (ATL) Bibliothek ...

- beschreibt und definiert Datenstruktur aller Attribute.
- Dient der Modellierung von Eigenschaften mitsamt Werten, Einheiten und Limitierungen.
- kann Objekte vom Typ *AttributeType* (AT) oder *Attribute* <A> umfassen.

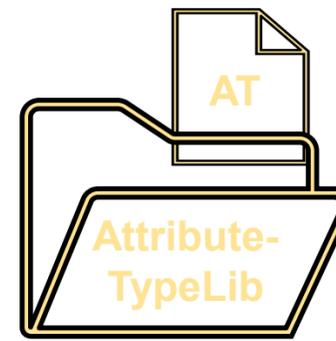
Constraints:

- Über Constraints lassen sich für einzelne Attribute Randbedingungen und Anforderungen definieren
 - OrdinalScaled = min./max. Werte
 - NominalScaled = Spezifikation diskrete Wertebereiche
 - UnknownType = beliebige Spezifikation der Syntax

Element	Beschreibung	Beispiel
Name	Name des Attributes	Gewicht
Unit	Einheit des Attributes	kg
AttributeDataType	Datentyp des Attributes nach XML-Notation	xs:string, xs:int
RefAttributeType	Pfad (Referenz) zu einem Eltern-Attributtyp	SomeStandardlib/Weight
DefaultValue	Vordefinierter Standardwert eines Attributes	40
Value	Wert des Attributes	50
RefSemantic	Erlaubt die semantische Definition eines Attributes, gespeichert als Referenz auf die Semantik	IRDI:0112/2///61360_4# AAH011
Constraint	Schränkt den Wertebereich oder die Grenzwerte eines Attributes ein	Max: 500 Min: 5
Attribute	Element zur Definition geschachtelter Attribute, auf diese Weise können Attribute selbst Attribute enthalten und komplexe Attributstrukturen bilden	Ein Attribut Gewicht könnte als Kind eine Liste von Messwerten aufnehmen, um eine Historie abzubilden

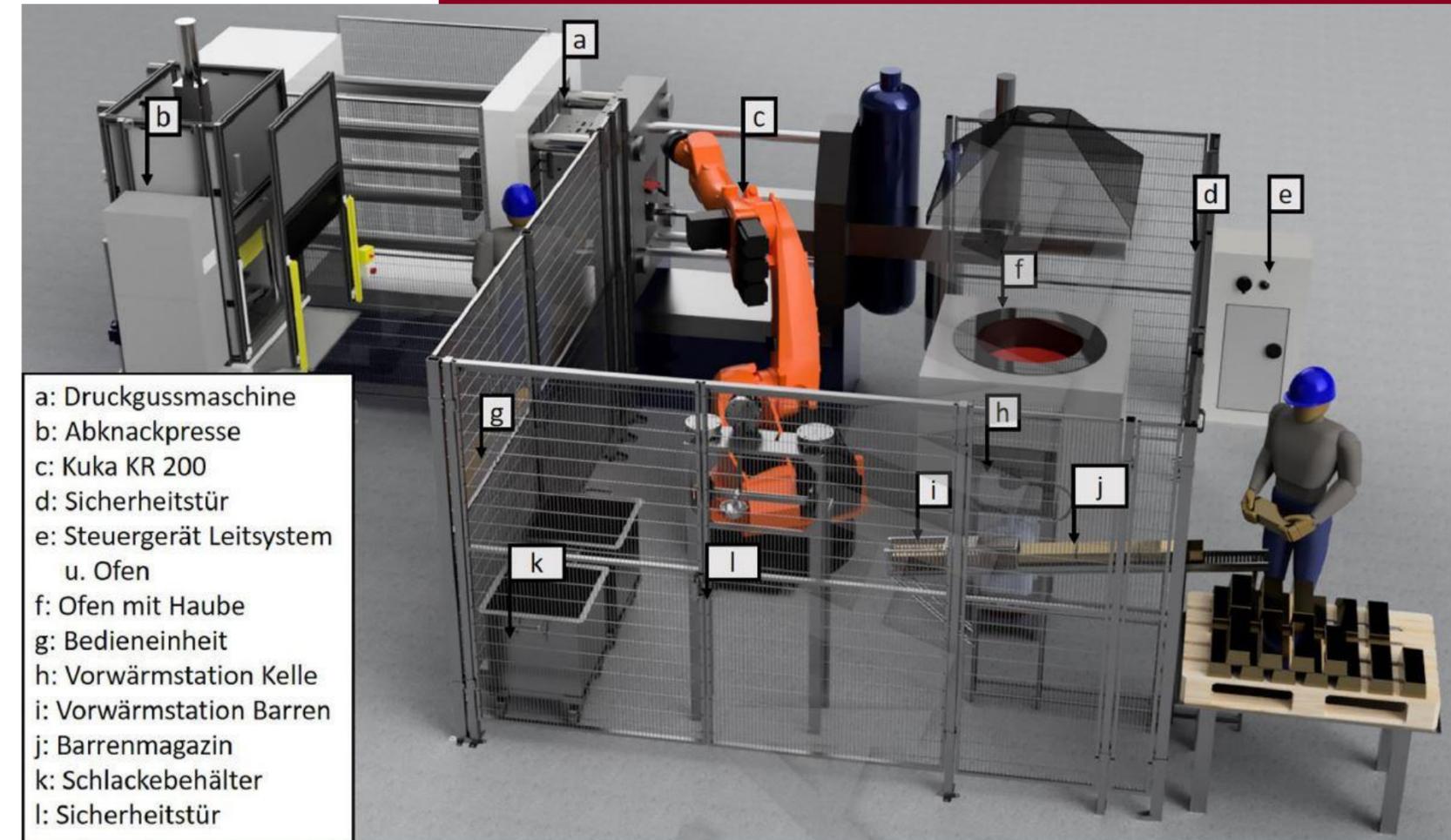
Quelle: R.Drath, AutomationML 2022, S.72 Tabelle 3.3

Lehrbeispiel: HiRoCast

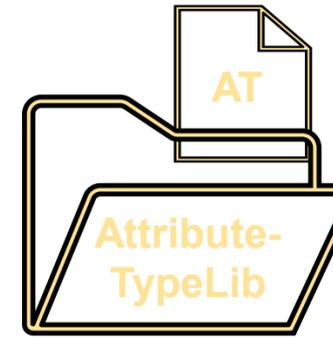


AttributeType und Attribute

Akteur:	AttributeType:	Attribute:
Druckgussmaschine	GießanlagenAttribute	Preis, Baujahr
Abknackpresse	ZerspanungsanlageAttribute	Scherkraft
Kuka KR 200	RoboterAttribute	Reichweite, Traglast, IP-Schutzklasse
Sicherheitszaun	SicherheitAttribute	Abmaße LxBxH
Sicherheitstüren	SicherheitAttribute	Abmaße LxBxH
Steuergeräte/Peripherie	SteuerungseinheitenAttribute	Netzspannung
Leitsystem/SPS	SteuerungseinheitenAttribute	Netzspannung
Robotersteuerung KRC2	RoboterAttribute	VersionsNr.
Vorwärmstationen B & K	-	-
Barrenmagazin	-	-
Schlackebehälter	-	-
Ofen mit Abdeckung	OfenAttribute	KeramikDurchmesser, Ofentemperatur
Endeffektor/Gießkelle	-	-



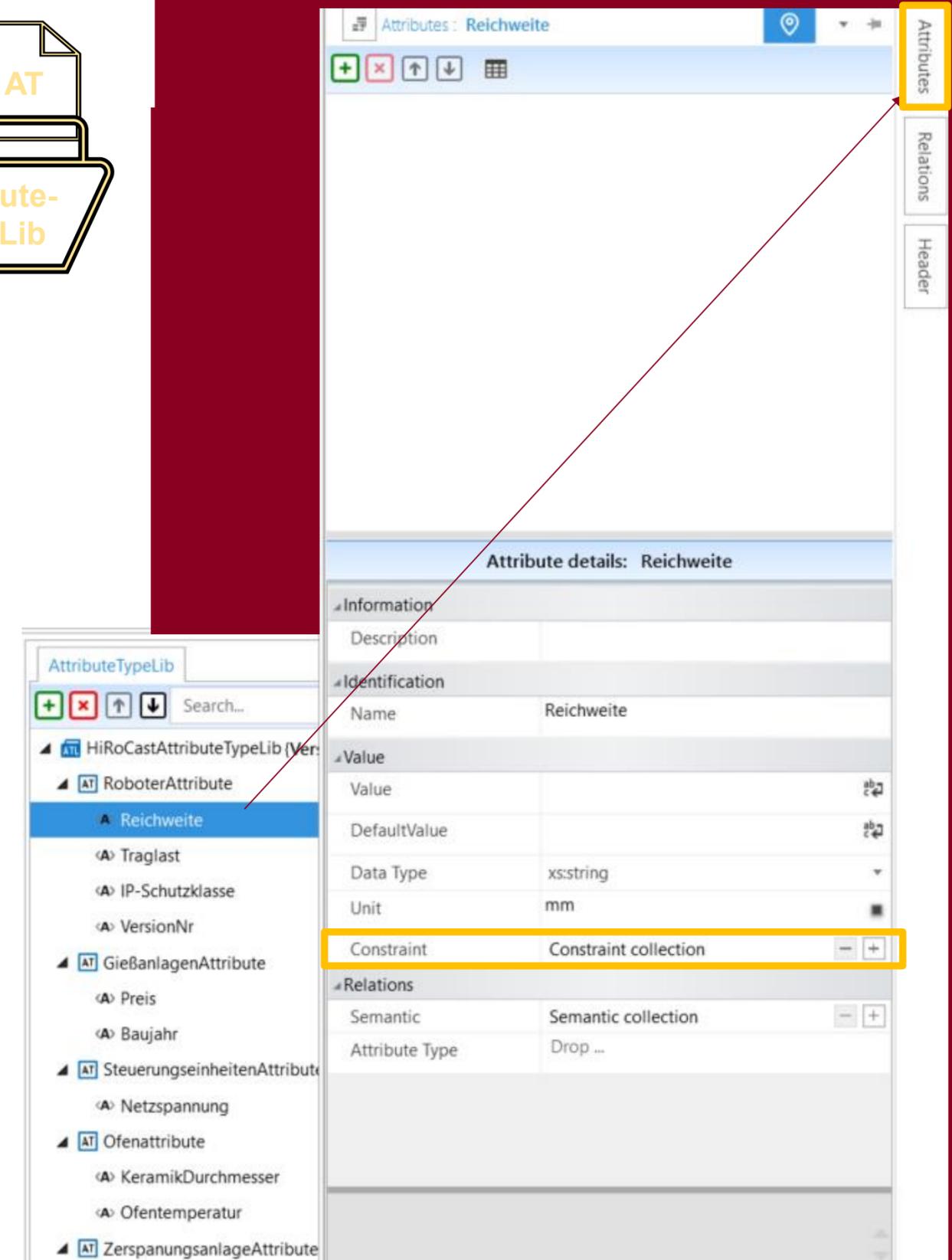
Lehrbeispiel: HiRoCast

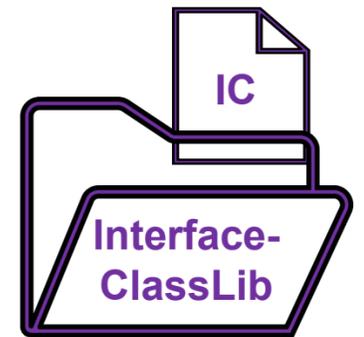


Constraints und Einheiten in AML

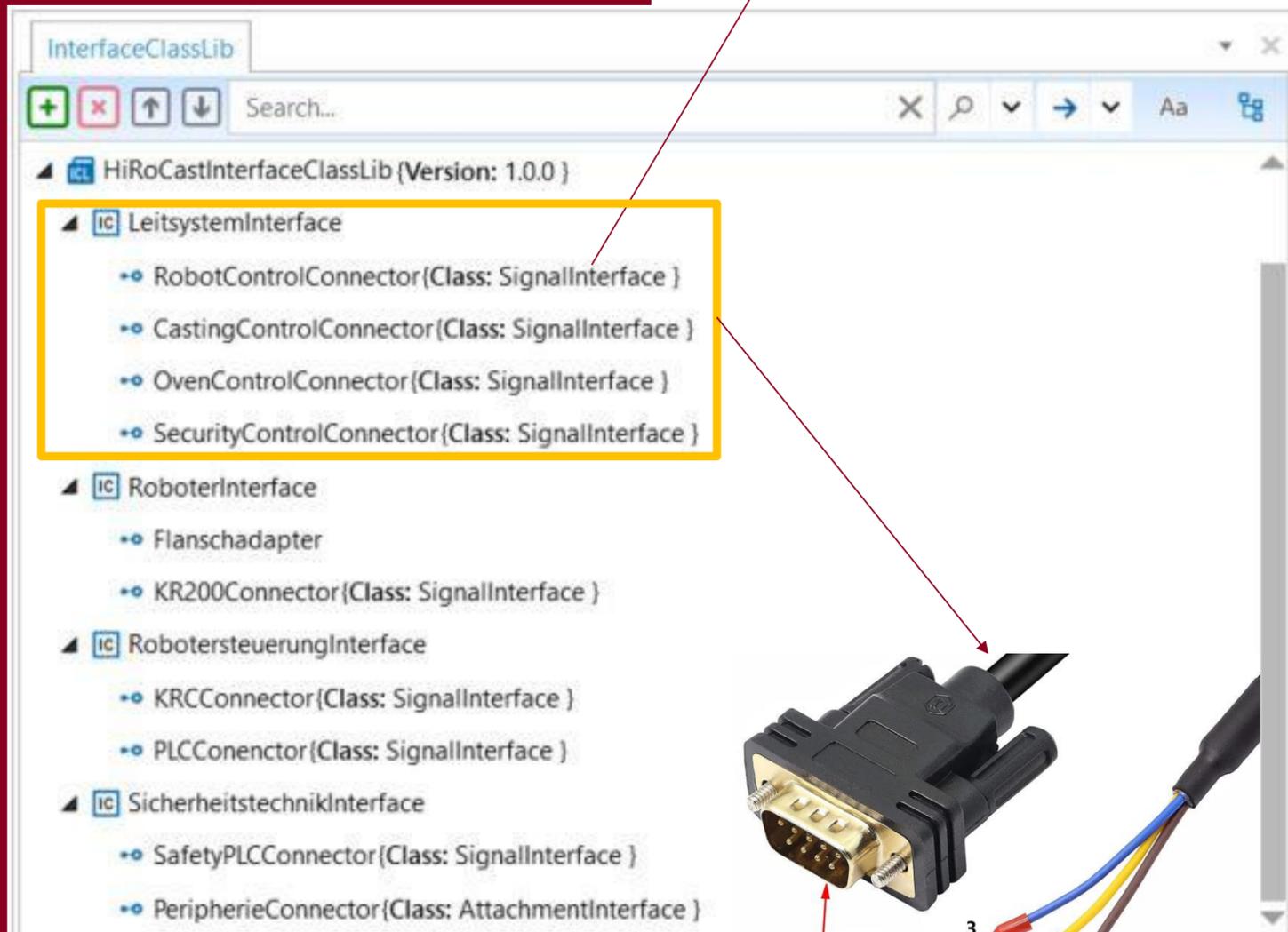
- Randbedingungen können über die Attribute Schaltfläche (rechts im AML-Editor) für die aktiv angewählte Instanz definiert werden
 - OrdinalScaled = min./max. Werte
 - NominalScaled = Spezifikation diskrete Wertebereiche
 - UnknownType = beliebige Spezifikation von Syntax

Attribute:	Constraint:	Einheit:
Preis, Baujahr	-	€; JJ/MM/TT
Scherkraft	max. 5kN	N
Reichweite, Traglast, IP-Schutzklasse	2 m; mind. 5 kg	mm;kg;-
Netzspannung	230 V	V
VersionsNr.	höher als 1.6	-
KeramikDurchmesser, Ofentemperatur	mind. 80 cm; mind. 1100°C	mm;°C





Klassenverweise und Vererbung sind erstmal zu vernachlässigen



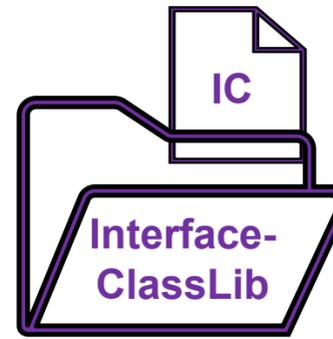
InterfaceClass

Die *InterfaceClassLib (ICL)* Bibliothek ...

- beschreibt Klassen von Schnittstellentypen.
- Ermöglicht Abbildung von Beziehungen zwischen realen Objekten.
- Kann mittels Hierarchisierung ein Port-Konzept Abbilden. (Erläuterung im fortgeschrittenen Kapitel der Schulung)

Modellierungsregeln:

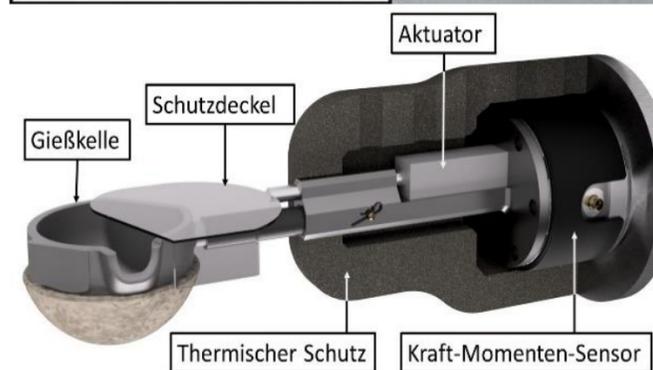
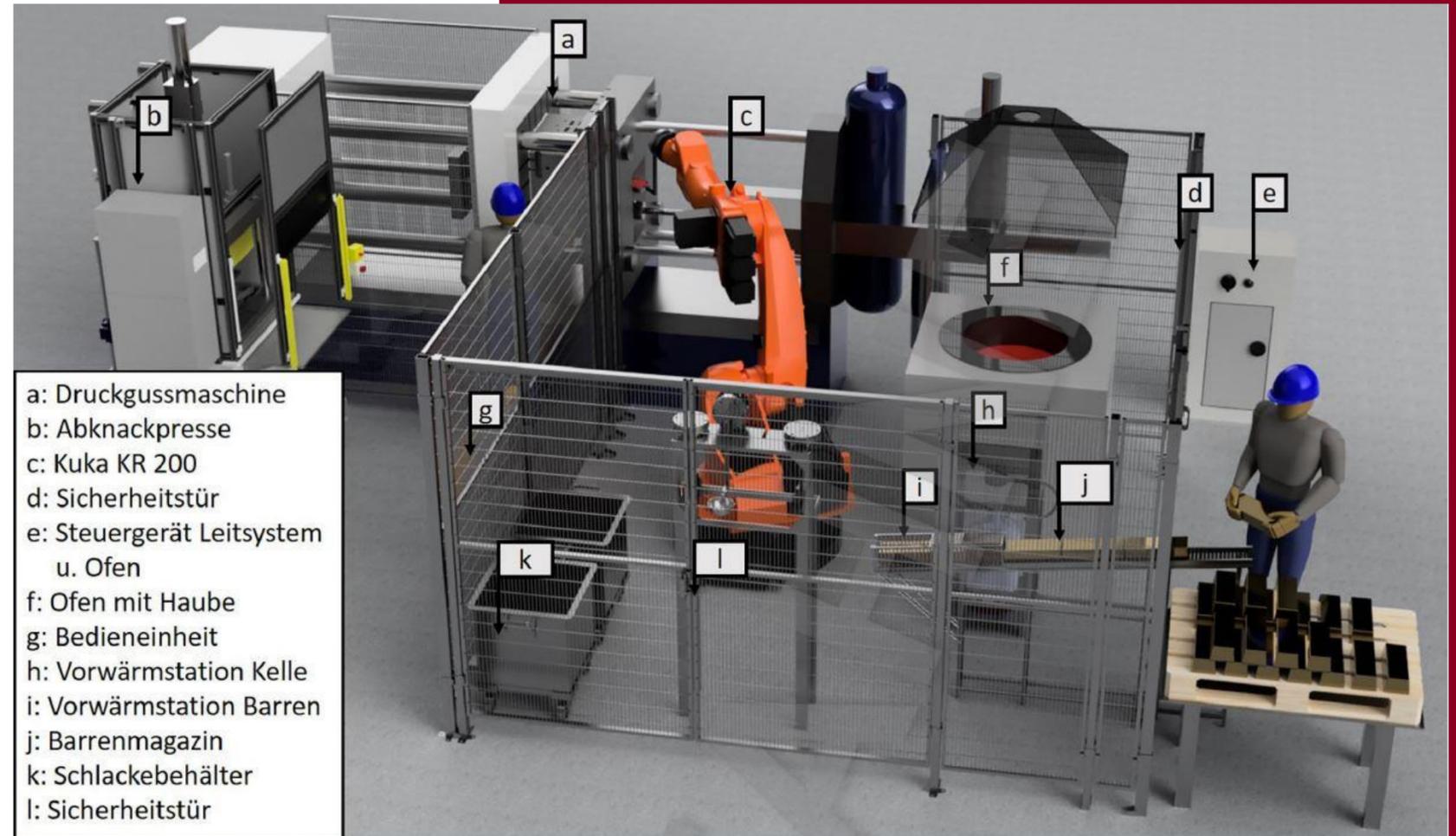
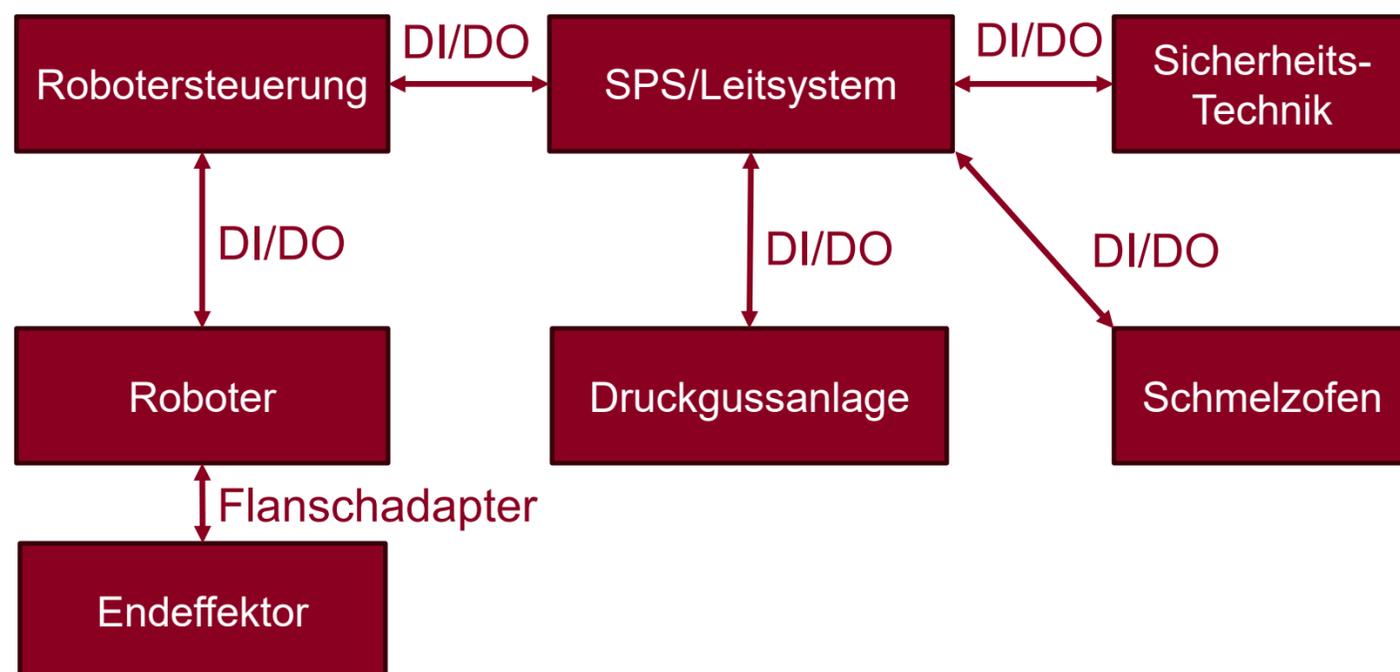
- Schnittstellenklassen sollten von Standard AML-Schnittstellenklassen abgeleitet werden.
- Schnittstellen haben keine Information über Richtungen. Sie sind universell ausgerichtet.
- Hierarchie in Schnittstellen ist rein organisatorisch und hat keine semantische Bedeutung
- Vererbung erfolgt explizit durch Elternklassen.



Lehrbeispiel: HiRoCast

Vereinfachte Schnittstellen Übersicht:

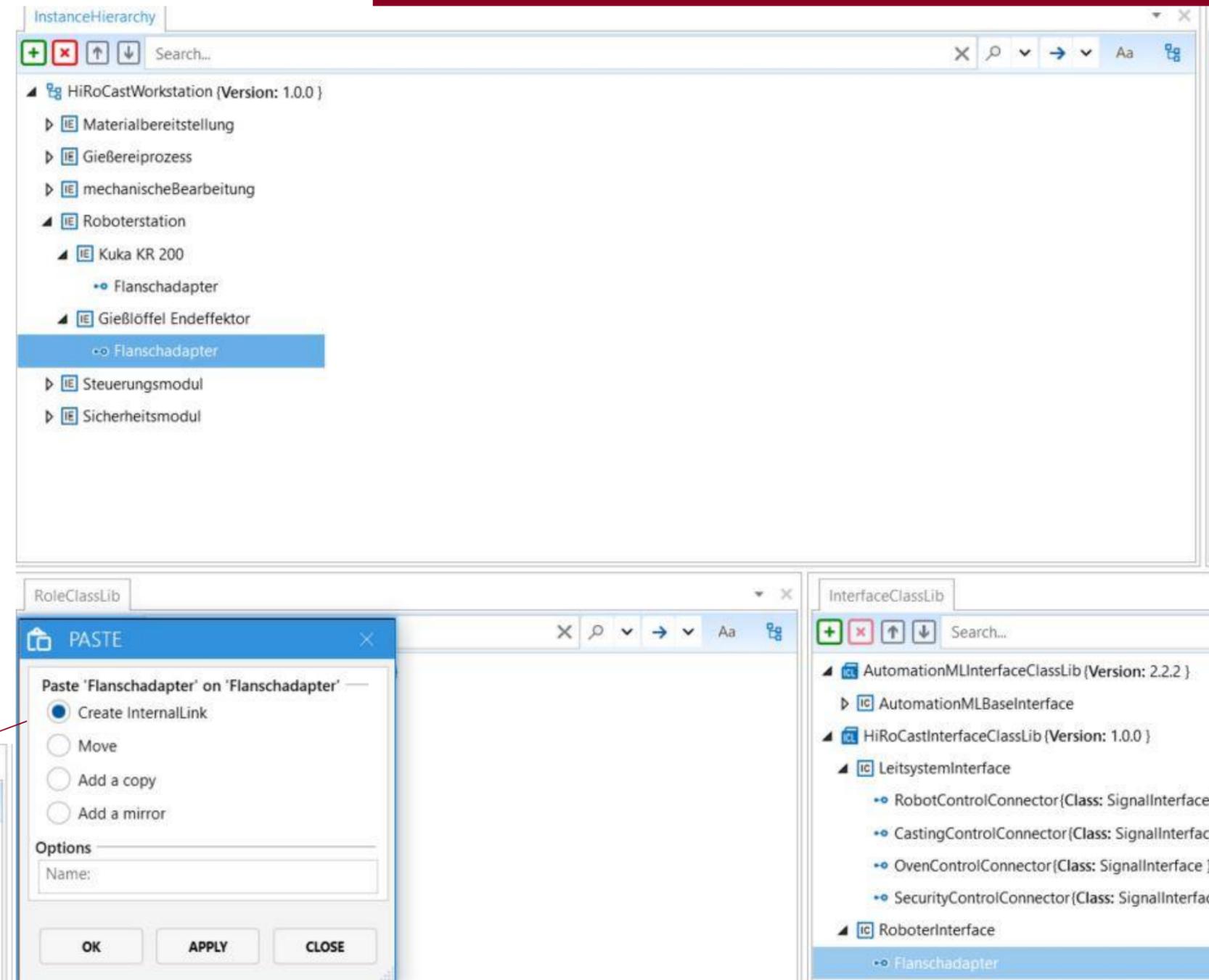
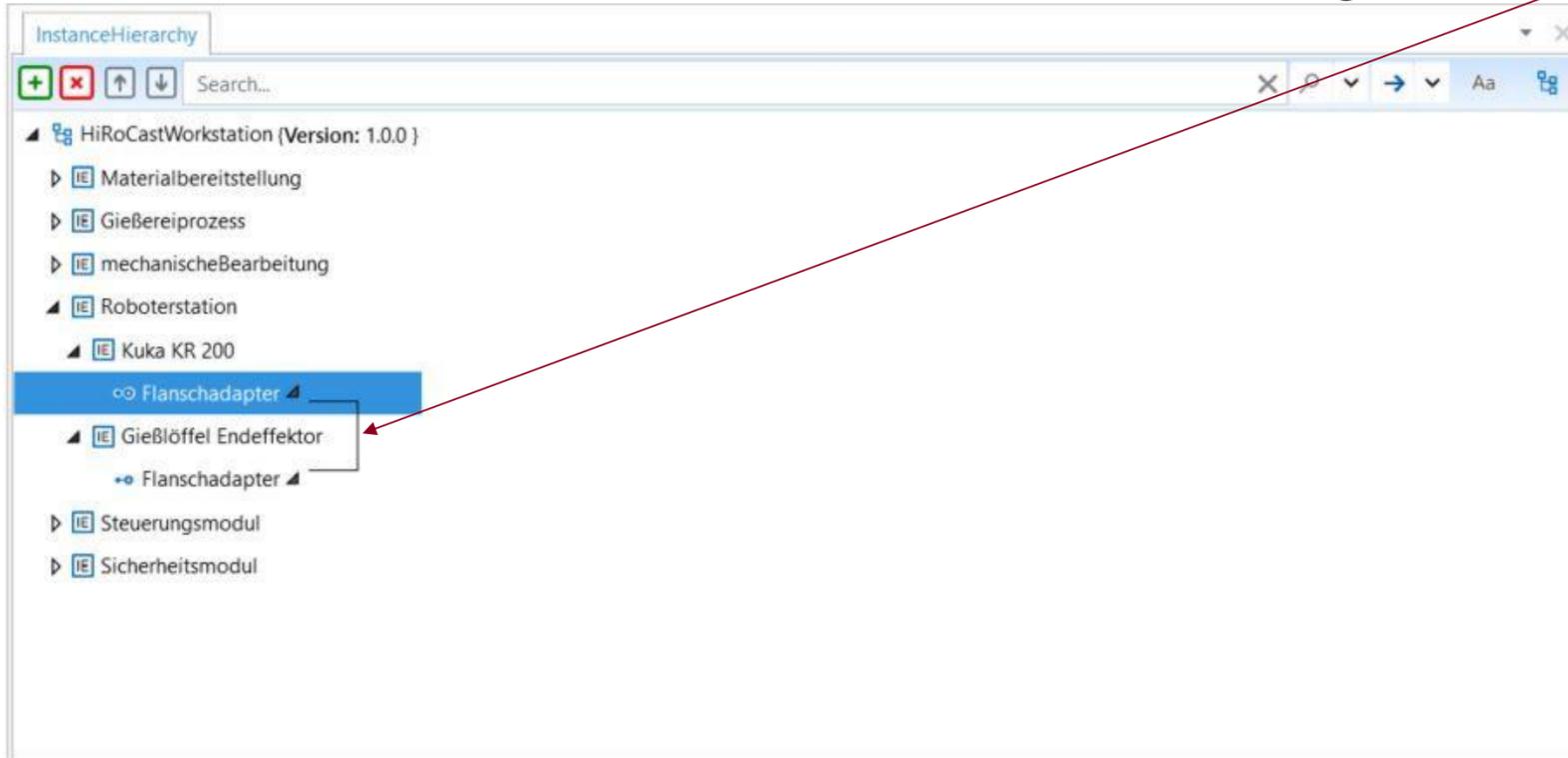
- DI / DO = Digitale In-/Output Schnittstelle
- Schnittstellen nicht signaltechnischer Natur (Bspw. mechanische Verbindungen, Ventile, etc.) können als *ExternalInterface* modelliert werden.

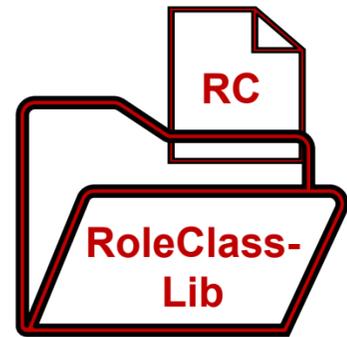


Verknüpfungen von Interfaces

InternalLink und Schnittstellen Verknüpfung

- Erstellen von Schnittstellen Verbindungen im AML-Editor erfolgt über *Internallinks*
- *InternalLinks* können per Drag & Drop der Schnittstellen erstellt werden und werden visuell im Editor dargestellt





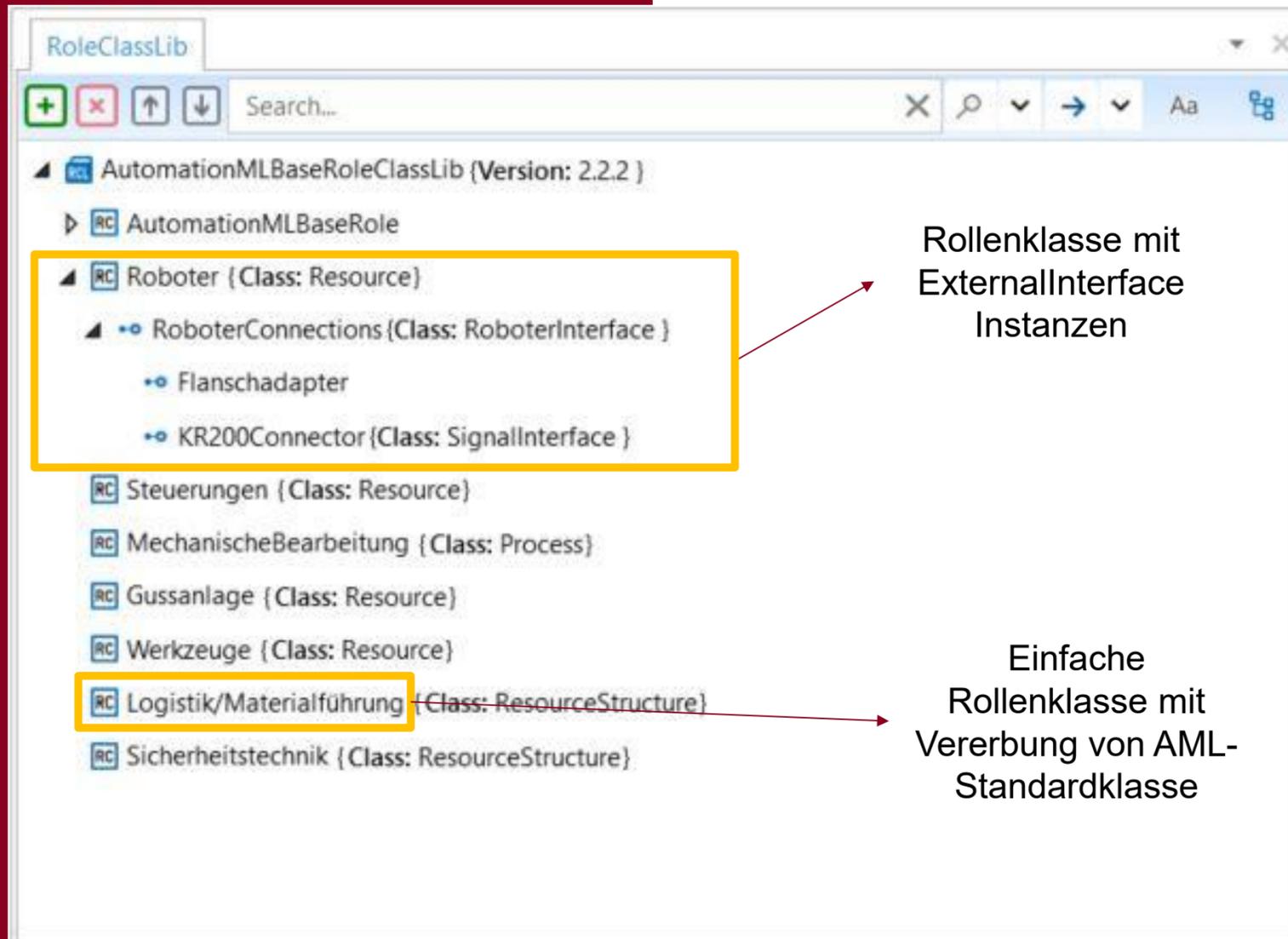
RoleClass

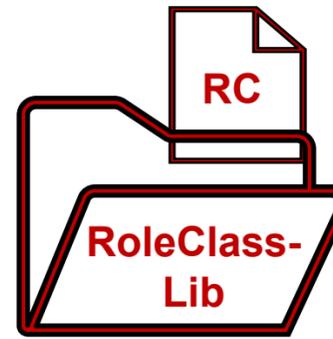
Die *RoleClassLib* (RCL) ist ...

- eine Abbildung einer Funktion oder abstrakten Bedeutung eines Systemakteurs.
- Rollenklassen (RC) können *ExternalInterfaces* oder *Attribute* beinhalten.

Modellierungsregeln:

- Rollenklassen enthalten keine Verschachtelungen weiterer Rollen
- Hierarchisches Modell hat keine Semantik im Datenmodell und ist rein organisatorischer Art (Analogie zu InterfaceClass)
- Vererbung wird explizit durch Verweis auf Elternklasse modelliert

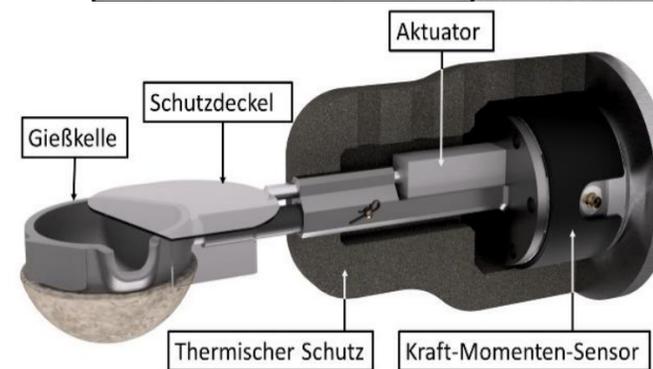
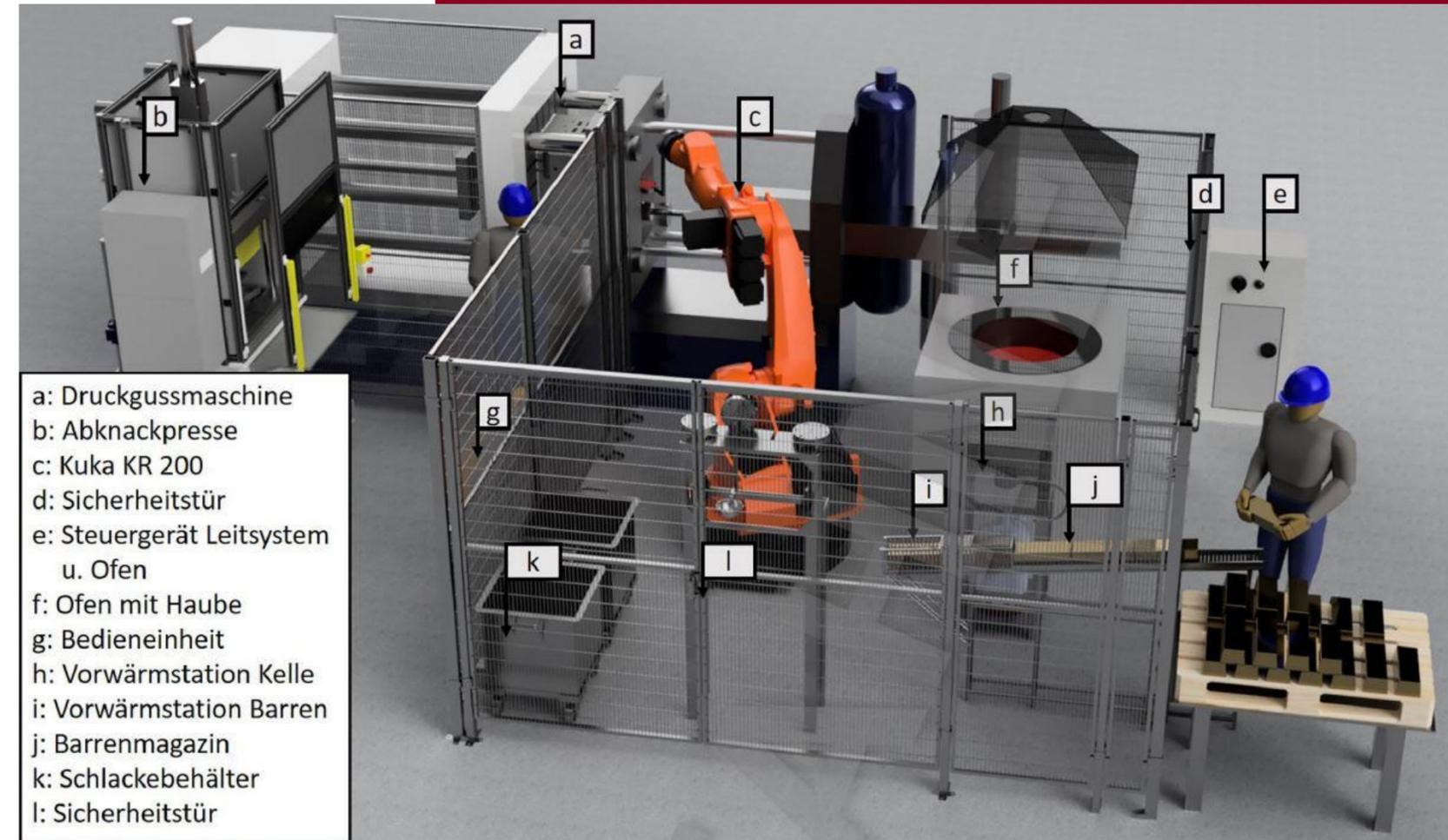




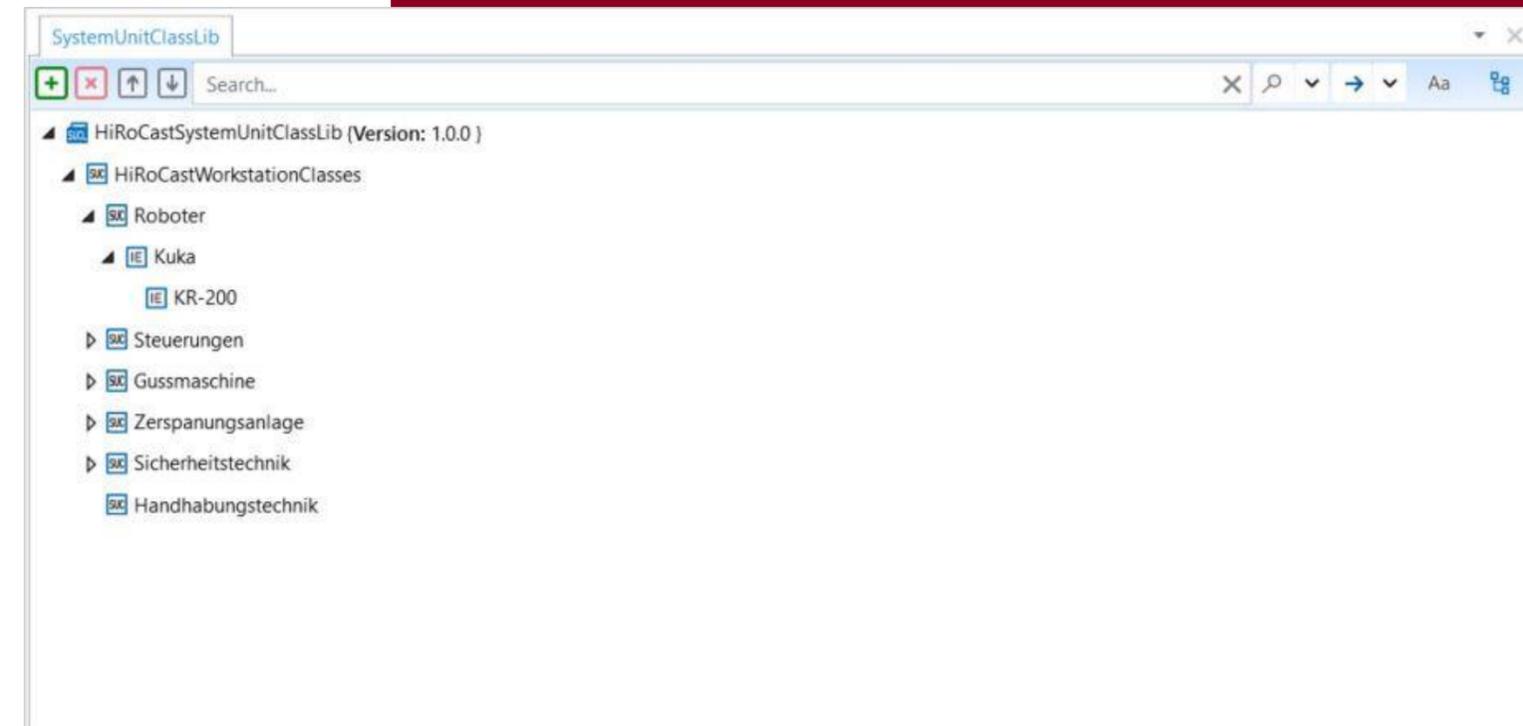
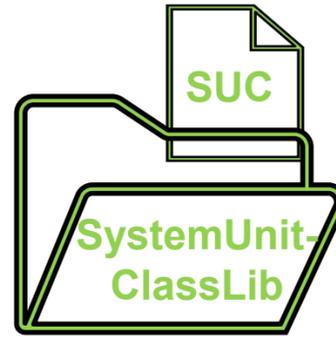
Lehrbeispiel: HiRoCast

Abstraktion der Systemakteure

- | | |
|-------------------------------|-----------------------------|
| • Druckgussmaschine | -> Gussanlage |
| • Abknackpresse | -> mechanische Bearbeitung |
| • Kuka KR 200 | -> Roboter |
| • Sicherheitstür & Zaun | -> Sicherheitstechnik |
| • Leitsystem & Bedieneinheit | -> Steuerungen |
| • Vorwärmstation B & K | -> Logistik/Materialführung |
| • Barrenmagazin | -> Logistik/Materialführung |
| • Tiegelschmelzofen mit Haube | -> Gussanlage |
| • Endeffektor/Gießkelle | -> Werkzeuge |



SystemUnitClass

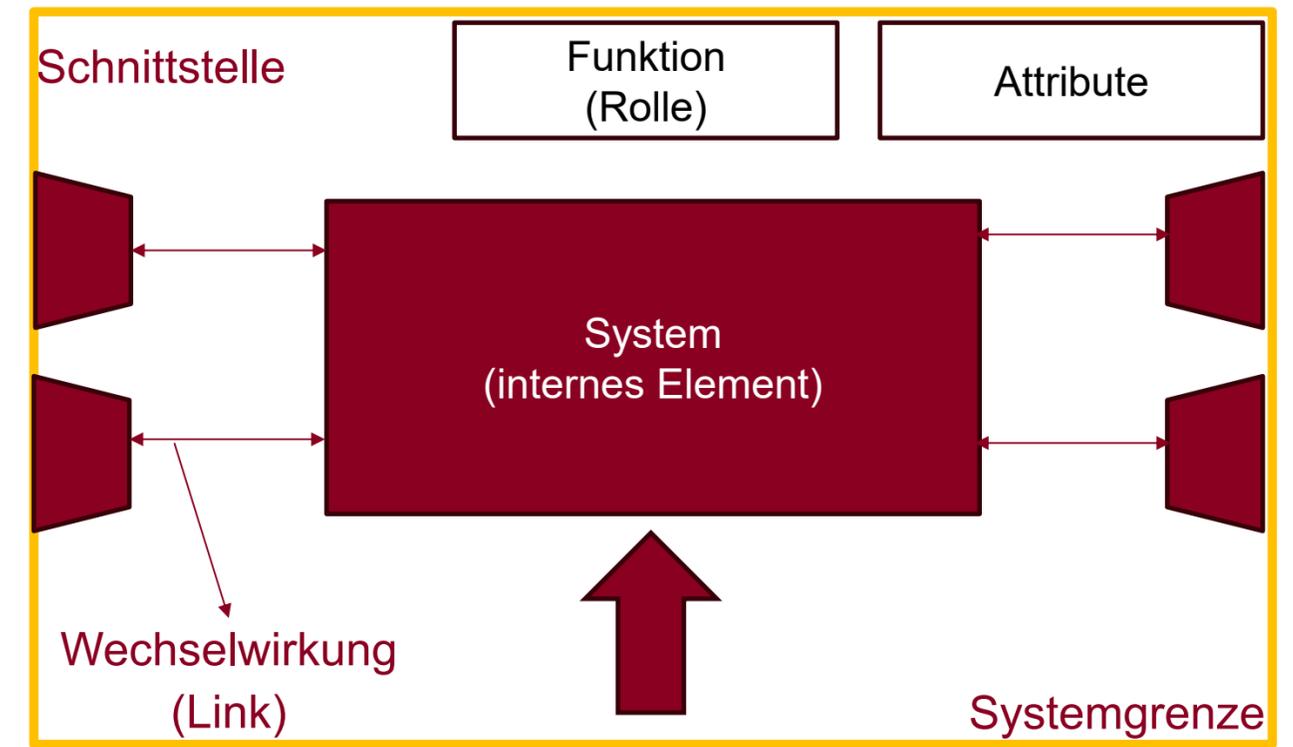


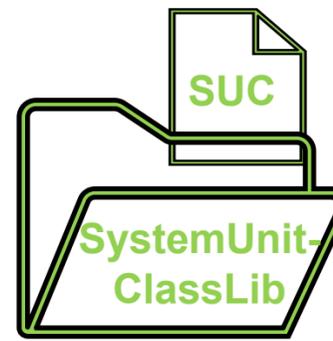
Die *SystemUnitClassLib* (SUC)...

- ist Bibliothek von SystemUnitKlassen (SUC).
- beschreibt Typen von physischen oder logischen Anlagen-Bestandteilen (Units) und deren Kombinationen.
- ermöglicht hierarchische Modellierung von Units.
- folgt der allg. Definition des Systembegriffs.

Modellierungsregeln:

- Eine *SystemUnitClass* kann beliebig viele *RoleClasses* unterstützen
- Jede *RoleClass* kann eigenes Mapping Objekt haben (relevant für Vererbung und Verweise).
- Hierarchie der *SystemUnitClass* hat keine eigene Semantik und ist im AML-Editor rein organisatorischer Art.
- Vererbung muss explizit durch Elternklasse zugewiesen werden.



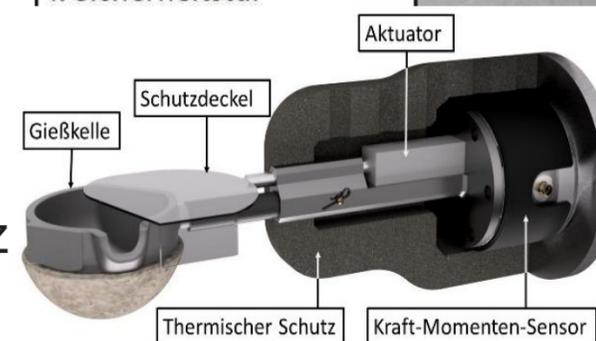
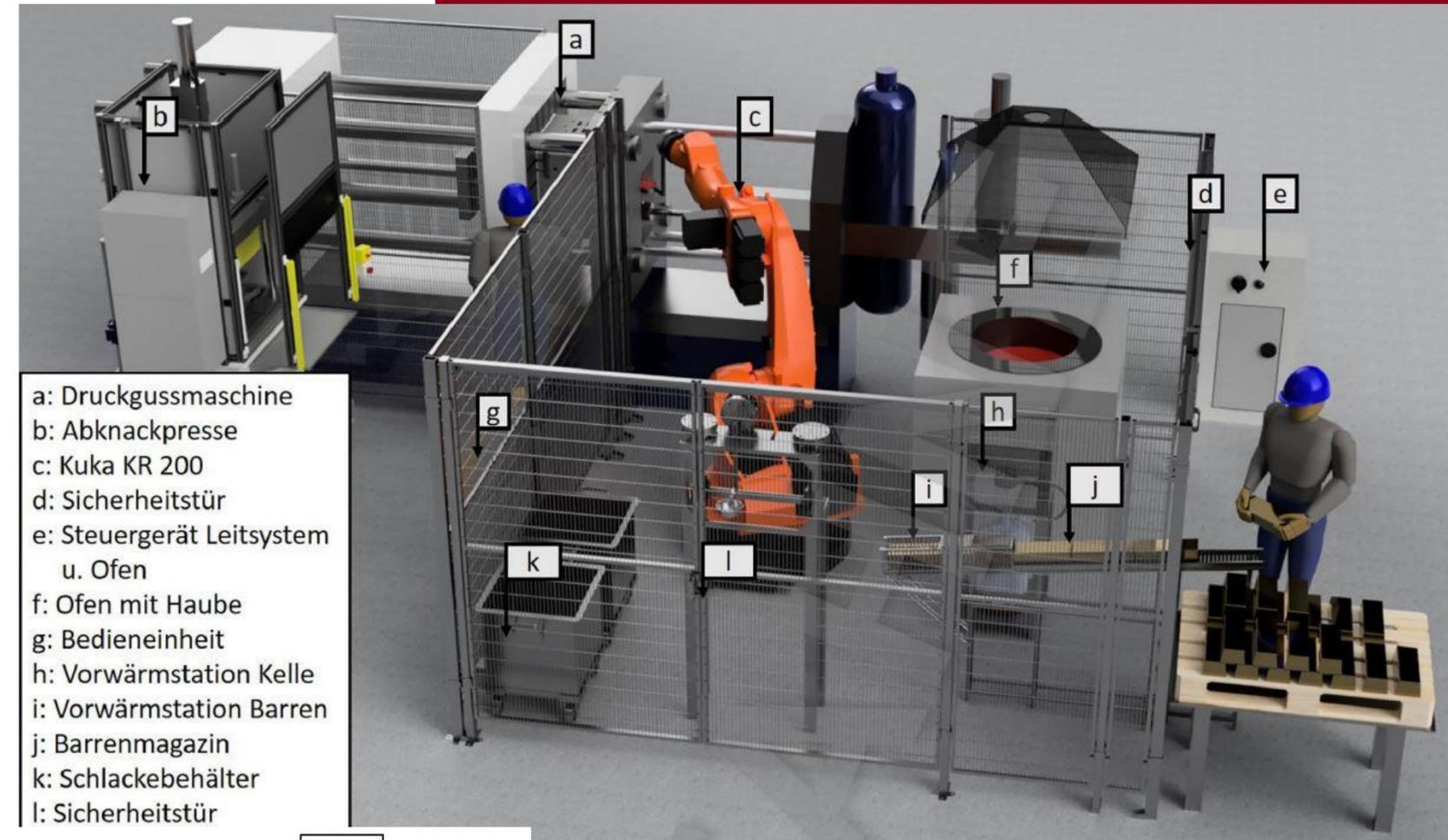


Lehrbeispiel: HiRoCast

Klassifikationskriterien:

- Kriterien der Klassifikation sind abhängig von der Perspektive des Entwicklers und der Ebene des Datenmodells.
- Klassifikation ist auf Basis von Technologie, Hersteller, Verfahren, Datenstruktur, etc.möglich.
- Klassifikation kann auf Sichten basieren Produktionssicht, Managementsicht, Finanzsicht, IT-Sicht etc.
- Beispielhafte Klassifikation nach Funktion und Hersteller:

Kuka KR 200: Roboter -> Hersteller -> Instanz
 Abknackpresse: Zerspanungsanlage->Hersteller->Instanz
 Leitsystem: Steuerungen->Hersteller->Instanz
 Druckguss: Gussmaschine->Hersteller->Instanz
 usw.





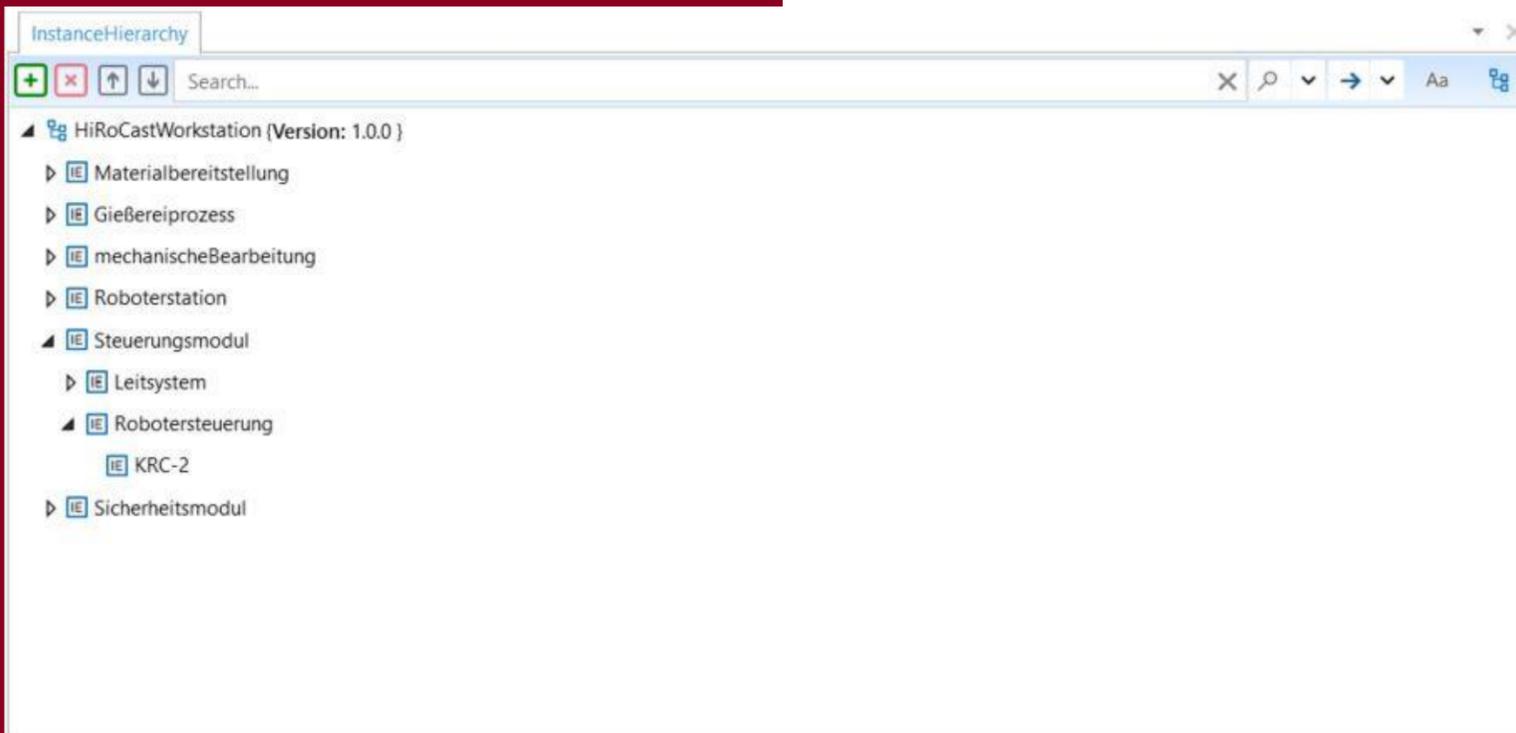
InstanceHierarchy

Die *InstanceHierarchy* ist ...

- Bibliothek von Objektinstanzen des Typ *InternalElement*.
 - *InternalElement* ist ein Datenmodell, das ein konkretes Exemplar eines physischen oder logischen Objekts beschreibt
- beschreibt hierarchisches Gesamtmodell projektbezogener Engineering-Informationen, das aus mehreren Komponenten und Sichten bestehen kann.
- durch alle weiteren Bibliotheken zu Vereinfachen und dient der globalen Systemmodellierung.

Modellierungsregeln:

- Hierarchietiefe ist unbeschränkt
- CAEX lässt freie Wahl für Hierarchie Architekturregeln
- CAEX gibt keine Namenskonventionen für Hierarchien vor





InstanceHierarchy

Modellierungsstrategien:

- 3 Arten der Modellierungsstrategie möglich
 - Ohne Klassen, alle Objekte als *InternalElement*
 - Nur mit Klassen, 1 *InternalElement* das *SystemUnitClass* instanziiert, Klasse enthält wiederum gesamte Anlagenhierarchie
 - Hybride Methode (Kombination von beidem)

Strategie 1: alle Elemente als IE in der InstanceHierarchy

Strategie 2: ein IE in der InstanceHierarchy das auf SUCL für Anlage verweist

AML-File mit dem Datenmodell der HiRoCast Fertigungszelle (ohne Vererbung)

The screenshot displays four panels from a software development environment, likely Siemens Teamcenter, showing the data model for a HiRoCast manufacturing cell. The panels are:

- InstanceHierarchy:** Shows a tree structure of HiRoCastWorkstation (Version: 1.0.0) components, including Materialbereitstellung, Gießereiprozess (Druckgussmaschine, Tiegelschmelzofen, Vorwärmstation), mechanischeBearbeitung, Roboterstation (Kuka KR 200, Gießelöffel Endeffektor, Flanschadapter), Steuerungsmodul (Leitsystem, Robotersteuerung), and Sicherheitsmodul.
- SystemUnitClassLib:** Shows a tree structure of HiRoCastSystemUnitClassLib (Version: 1.0.0) components, including HiRoCastWorkstationClasses (Roboter, Steuerungen (SPS, Beckhoff, Siemens), Gussmaschine (Nabertherm, Schüssler), Zerspanungsanlage, Sicherheitstechnik (Rohde, Schutzzaun A1, Schutztür B5)).
- RoleClassLib:** Shows a tree structure of AutomationMLBaseRoleClassLib (Version: 2.2.2) components, including AutomationMLBaseRole, Roboter (Class: Resource), RoboterConnections (Class: RoboterInterface) (Flanschadapter, KR200Connector (Class: SignalInterface)), Steuerungen (Class: Resource), MechanischeBearbeitung (Class: Process), Gussanlage (Class: Resource), Werkzeuge (Class: Resource), Logistik/Materialführung (Class: ResourceStructure), and Sicherheitstechnik (Class: ResourceStructure).
- InterfaceClassLib:** Shows a tree structure of AutomationMLInterfaceClassLib (Version: 2.2.2) components, including HiRoCastInterfaceClassLib (Version: 1.0.0) (LeitsystemInterface (RobotControlConnector (Class: SignalInterface), CastingControlConnector (Class: SignalInterface), OvenControlConnector (Class: SignalInterface), SecurityControlConnector (Class: SignalInterface)), RoboterInterface (Flanschadapter, KR200Connector (Class: SignalInterface)), RobotersteuerungInterface (KRCCconnector (Class: SignalInterface), PLCConector (Class: SignalInterface)), and SicherheitstechnikInterface).
- AttributeTypeLib:** Shows a tree structure of HiRoCastAttributeTypeLib (Version: 1.0.0) components, including RoboterAttribute (Reichweite, Traglast, IP-Schutzklasse, VersionNr), GießanlagenAttribute (Preis, Baujahr), SteuerungseinheitenAttribute (Netzspannung), and Ofenattribute (KeramikDurchmesser, Ofentemperatur). ZerspanungsanlageAttribute is also listed.

Inhaltsübersicht

Einführung

Grundlagen und Definitionen

XML/CAEX/OOP

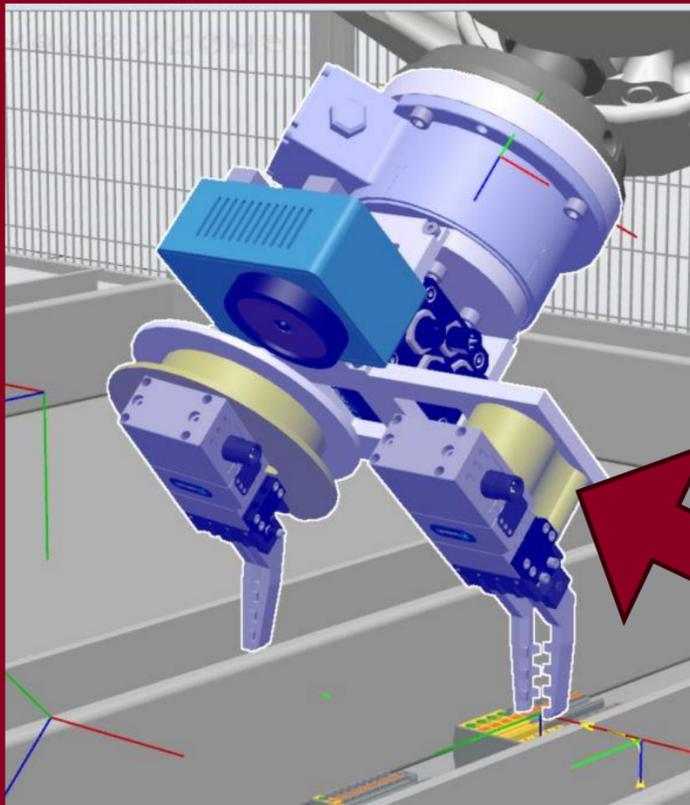
Der AML-Editor

AML-Trainingsaufgabe

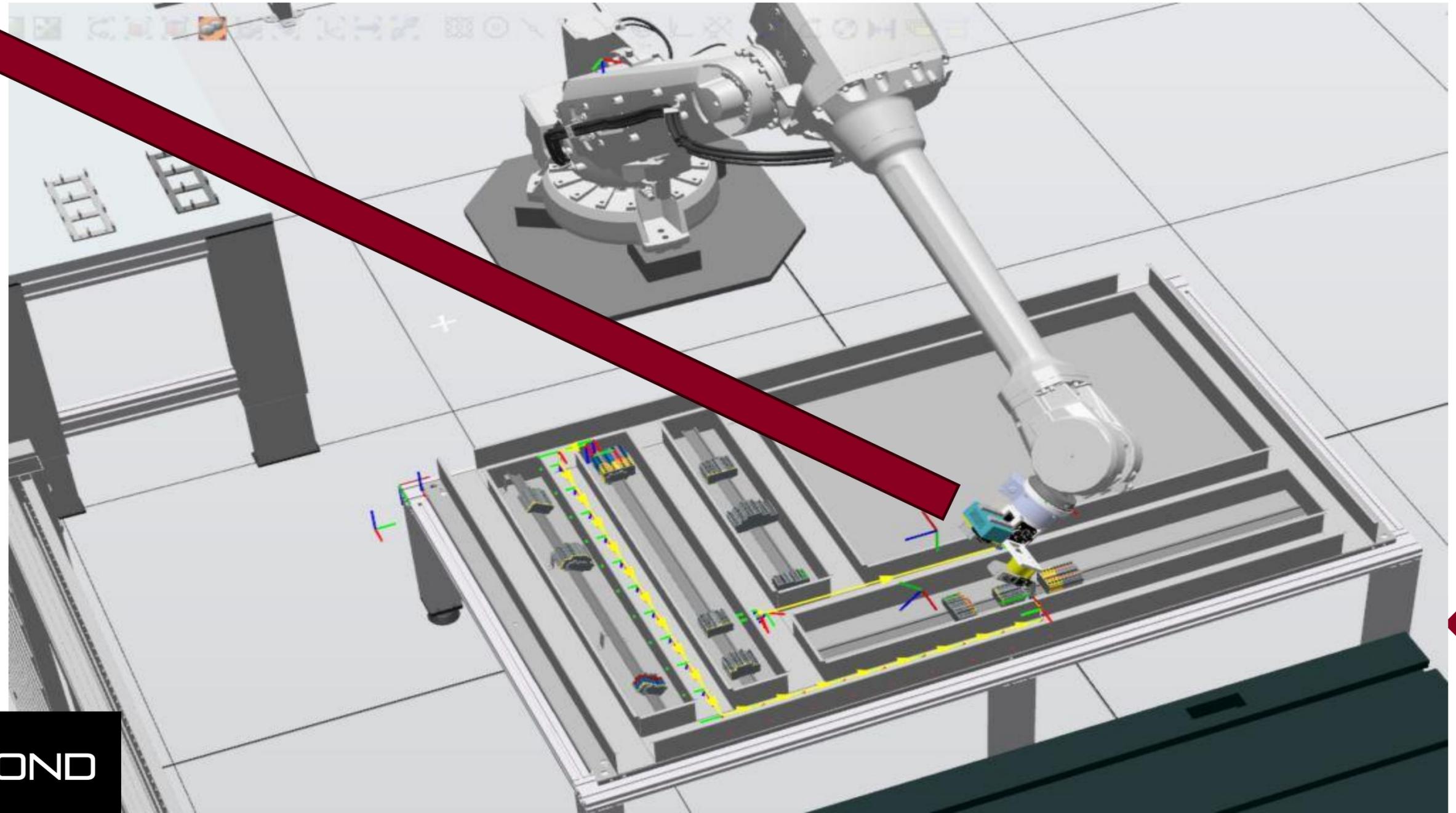
Beispiellösung

Lessons Learned & Ausblick

Trainingsaufgabe

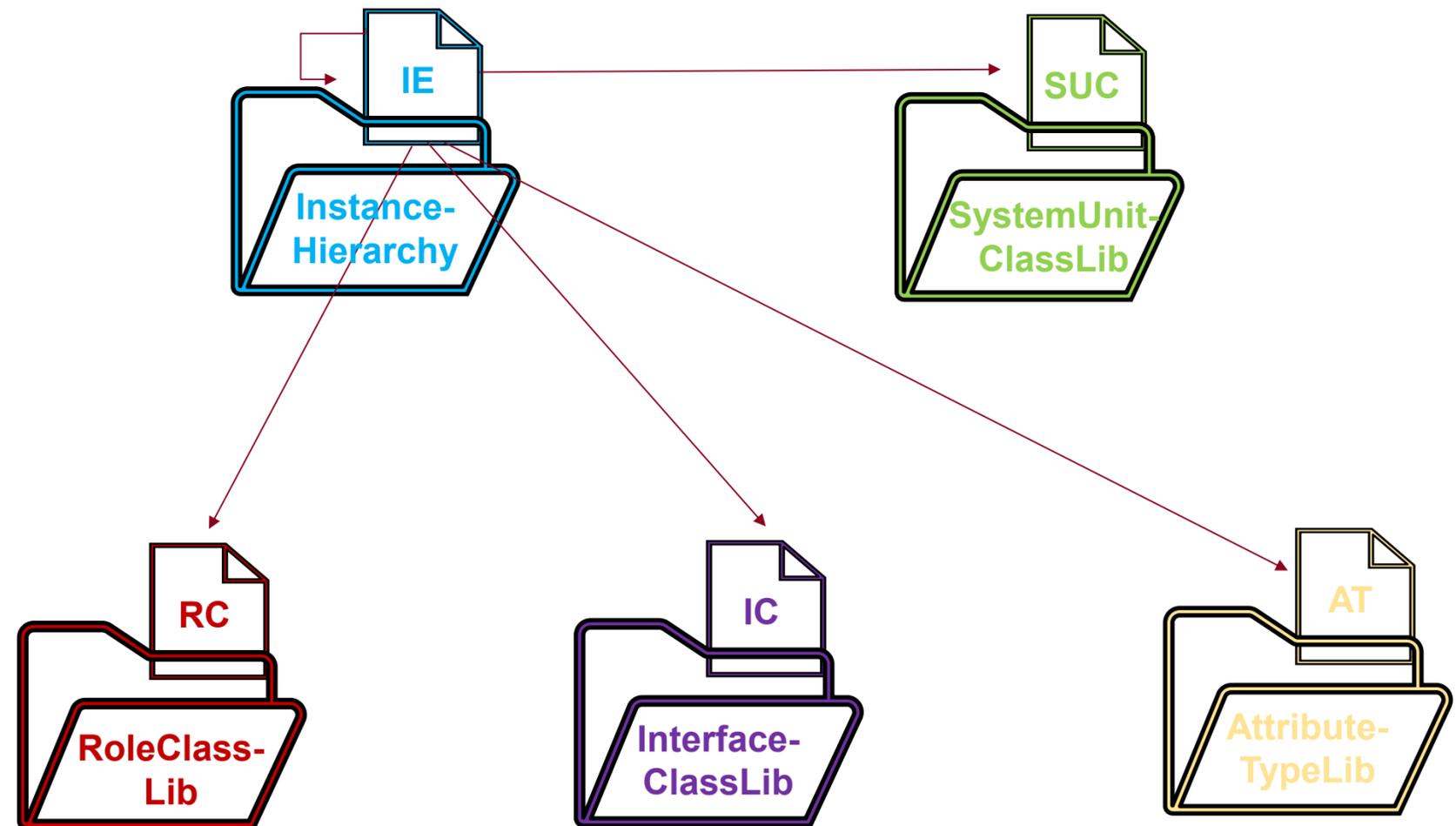


**Modellierung der
Automationszelle und des
Endeffektors aus dem
Projekt RoboScht in AML**



Aufgabenstellung RoboSchaft Fertigungszelle

1. Erstellen Sie für die Roboterzelle und den Endeffektor aus dem Projekt RoboSchaft, jeweils die erforderlichen Bibliotheken und das zugehörige AML-Datenmodell.
 - a) Führen Sie eine objektorientierte Analyse der Bestandteile beider Systeme (Zelle, Werkzeug) durch.
 - b) Erstellen Sie die erforderlichen Bibliotheken (SUC, IC ,ATL, RCL), Instanzen (IE) und Eigenschaften (Attribute).
 - c) Erweitern Sie ihr Modell um die notwendigen Semantiken (InternalLink,Ref) zur Abbildung der Systemstrukturen beider Systeme.
 - d) Erweitern Sie ihre Attribute und Bibliotheken um die Informationen aus dem Anforderungskatalog der Systemkomponenten



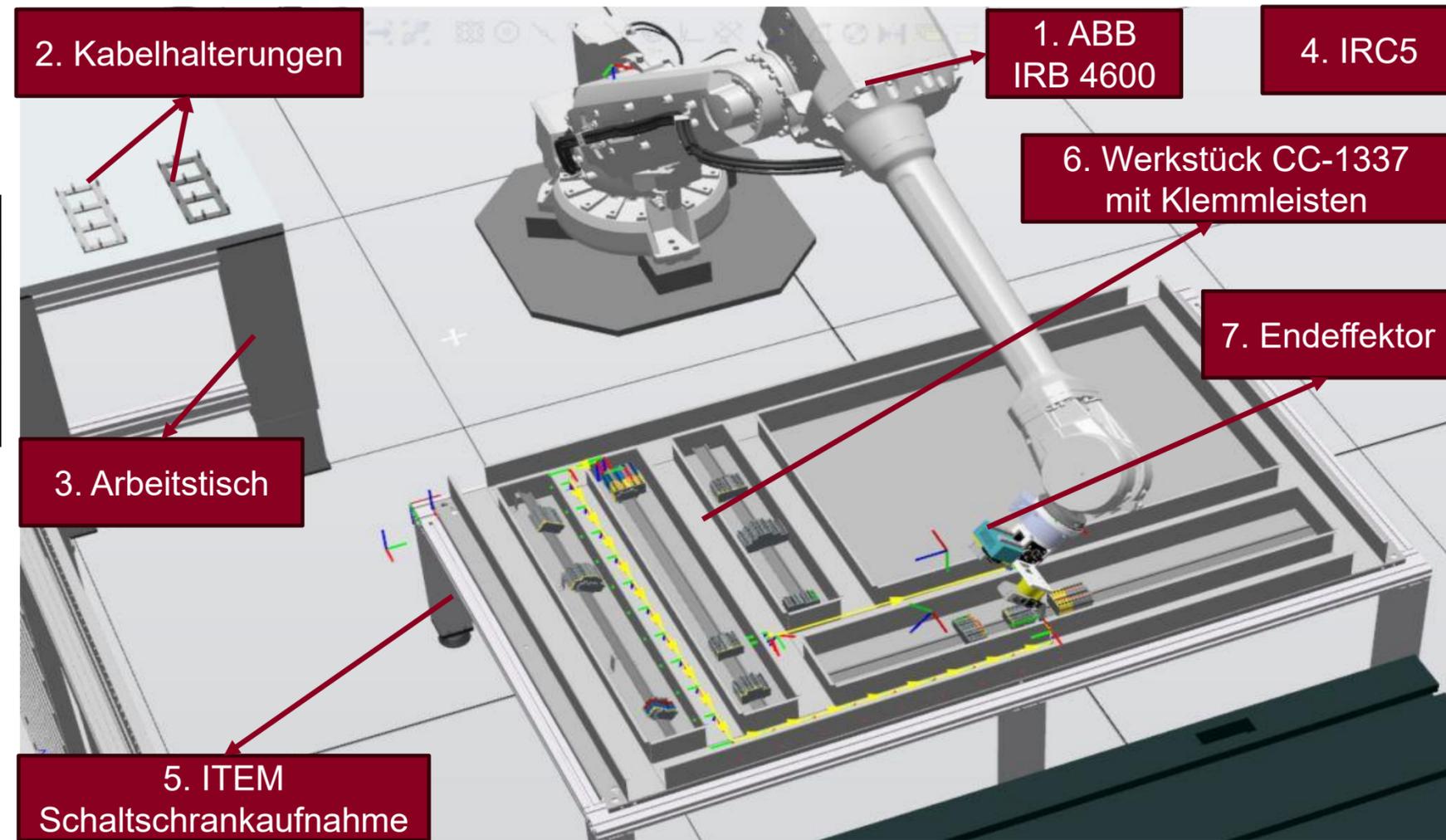
Hinweise und Informationen: zur RoboScht Zelle

• Anforderungsliste:

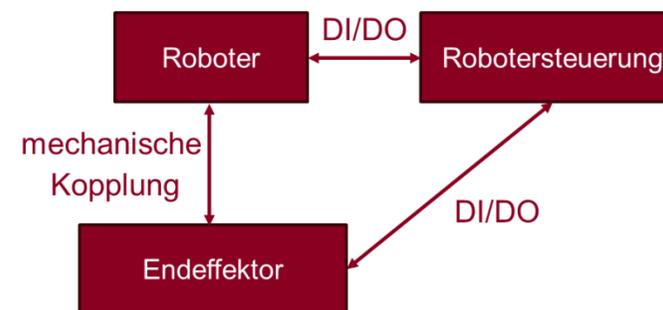
Attribute:	Constraint:	Einheit:
Reichweite	> 2	m
Positionsgenauigkeit	Abweichung < 10	mm
Standard IP Adressenbereich	zwischen XX.XXX.130.000 und XX.XXX.134.000	-

Attribute der Zellenkomponenten:

Akteur:	AttributeType:	Attribute:
Arbeitstisch	MaterialhandhabungAttribute	Abmaße LxBxT, Traglast
Kabelhalterung	MaterialhandhabungAttribute	Abmaße LxBxT, Koordinaten x, y, z
ABB IRB 4600	RoboterAttribute	Reichweite, Traglast
ITEM Schaltschrankaufnahme	MaterialhandhabungAttribute	Abmaße LxBxT, rel. Position x, y, z
Robotersteuerung IRC5	SteuerungAttribute	Netzwerk IP:
Muster CC-1337	ProduktAttribute	Anzahl Verdrahtungen, Lage xx, yy, zz
Roboscht Effektor	WerkzeugAttribute	Greifkraft

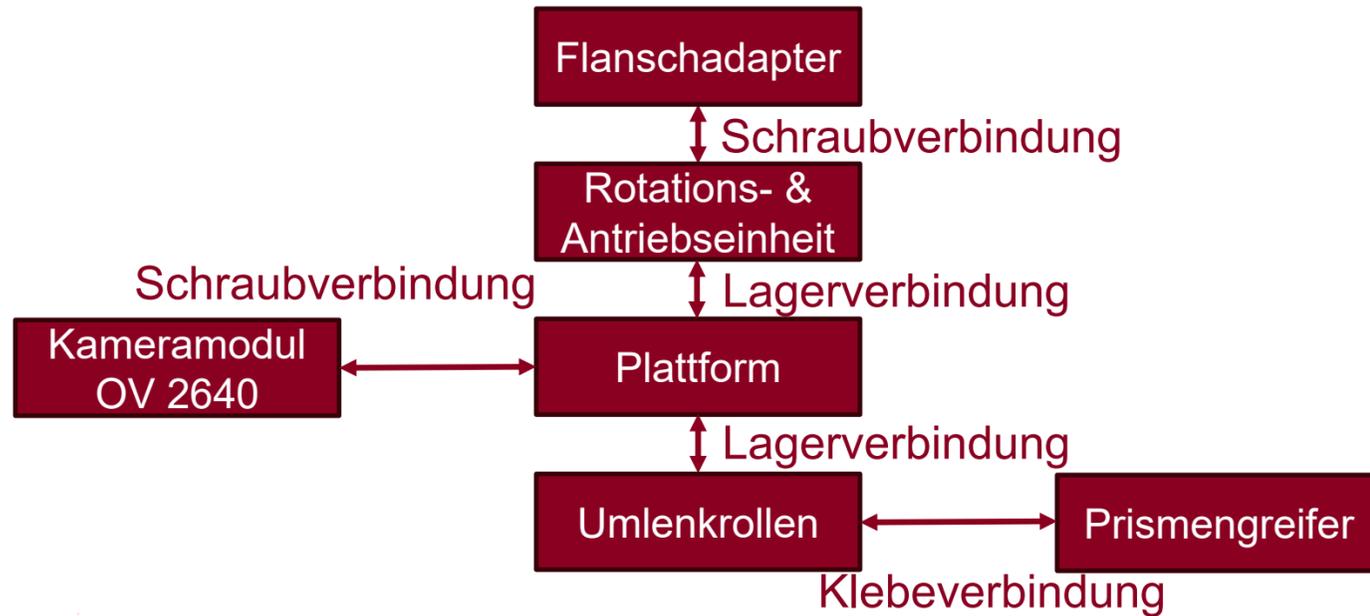


Mechanische und Signal-Schnittstellen:



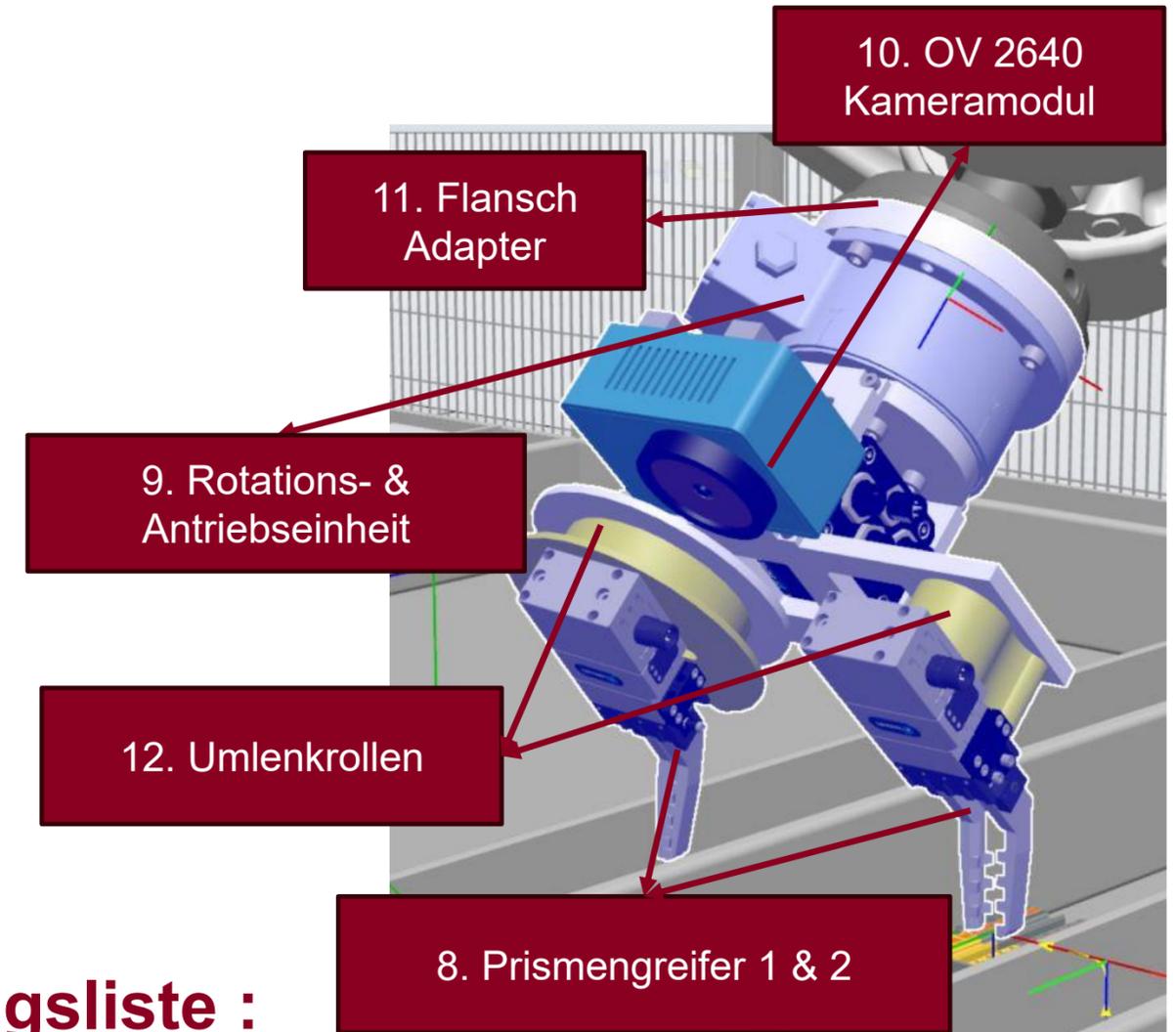
Hinweise und Informationen: zum RoboSchaft Werkzeug

Mechanische und Signal-Schnittstellen:



Attribute der Zellenkomponenten:

Akteur:	AttributeType:	Attribute:
Prismengreifer 1	HandhabungsAttribute	Greifkraft
Prismengreifer 2	HandhabungsAttribute	Greifkraft
Rotations- & Antriebseinheit	HandhabungsAttribute	Drehzahl, Motormoment
OV 2640 Kameramodul	SensorikAttribute	Auflösung, FPS
mech. Flanschadapter	KonstruktionAttribute	Schraubverbindung Isometrisch, Passung
Umlenkrollen	HandhabungsAttribute	Durchmesser, Umfang, Tiefe



Anforderungsliste :

Attribute:	Constraint:	Einheit:
FPS	mind. 30	1/s
Schraubverbindungen	mind. ISO-M4 ; max. ISO-M8	-
Greifkraft	F < 100	N

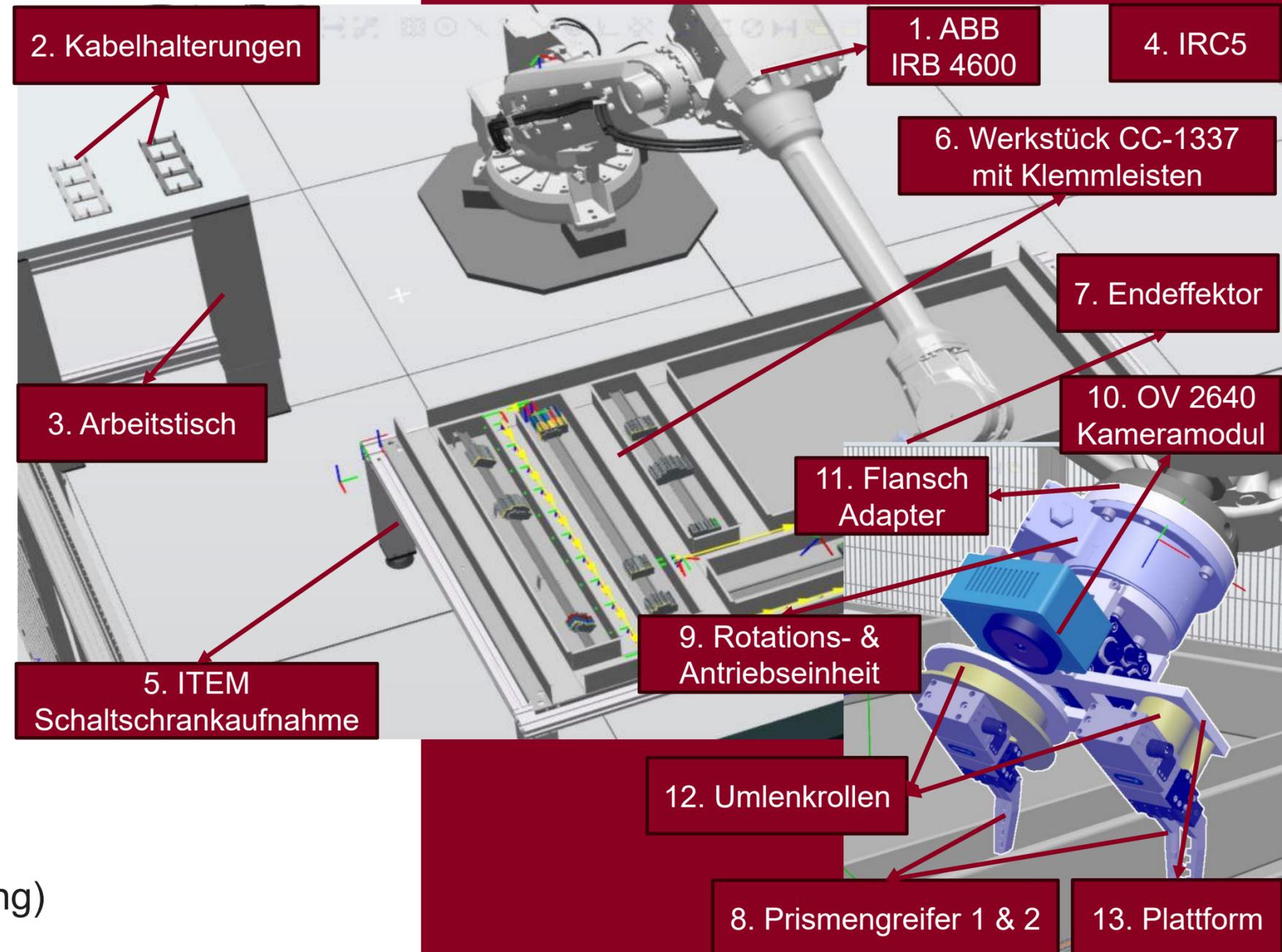
Komponenten RoboScht Zelle

Die RoboScht-Anlage besteht aus....

1. ABB IRB 4600 (Roboter)
2. Kabelhalterungen (Materialhandhabung)
3. Arbeitstisch (Materialhandhabung)
4. Robotersteuerung IRC5 (Steuerungseinheit)
5. ITEM Schaltschrank-Aufnahme (Materialhandhabung)
6. Werkstück CC-1337 mit Klemmleisten (Produktmuster)
7. Endeffektor (Werkzeug)

Der Endeffektor besteht aus....

8. Prismengreifer 1 & 2 (Handhabungssystem)
9. Rotations- & Antriebseinheit (Handhabungssystem)
10. OV2640 Kameramodul (Sensorsystem)
11. Flansch Adapter (mechanische Verbindung)
12. Umlenkrollen/Kabelführung (Materialführung)
13. Plattform/Rahmen (mechanische Verbindung)



Inhaltsübersicht

Einführung

Grundlagen und Definitionen

XML/CAEX/OOP

Der AML-Editor

AML-Trainingsaufgabe

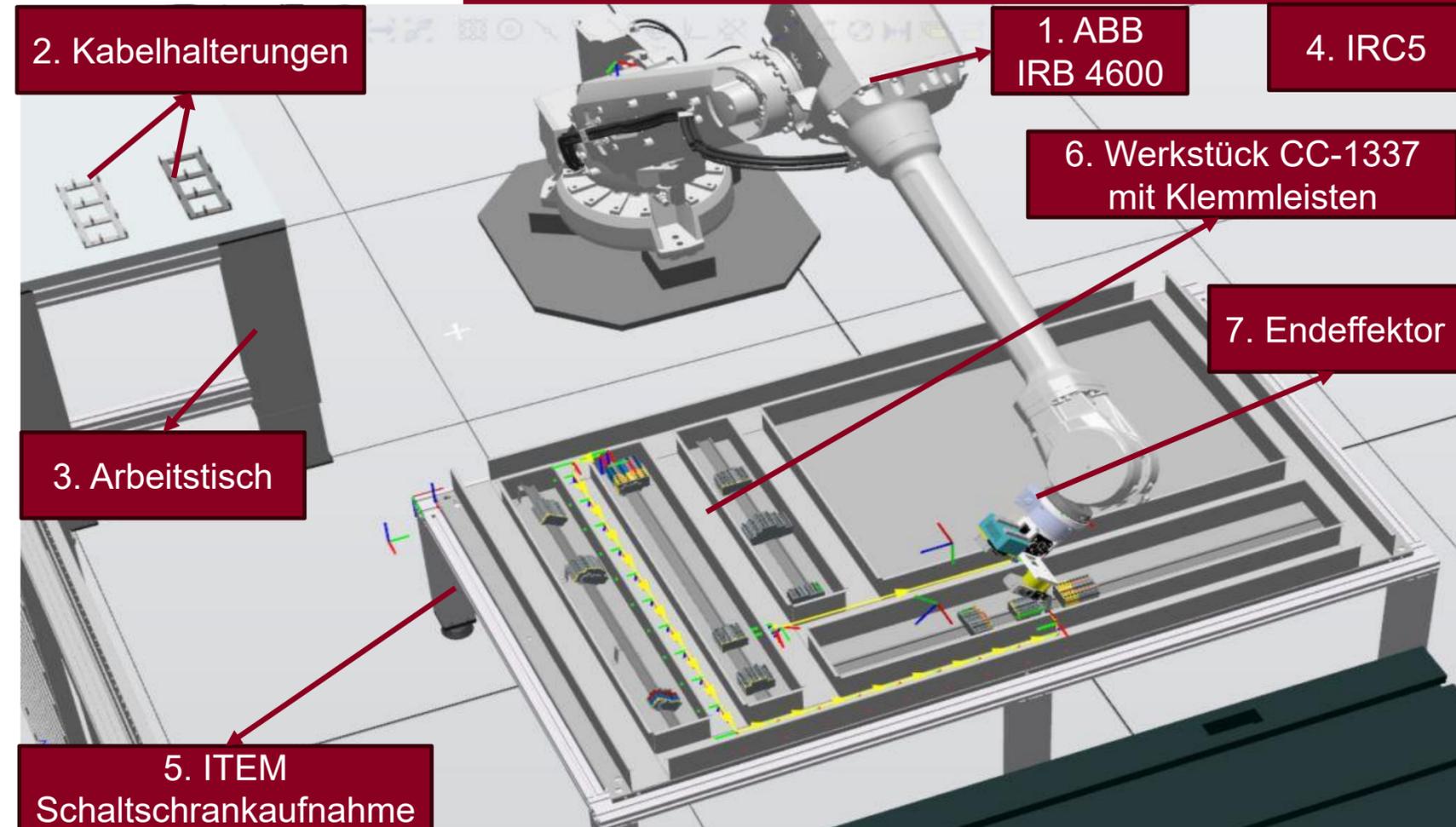
Beispiellösung

Lessons Learned & Ausblick

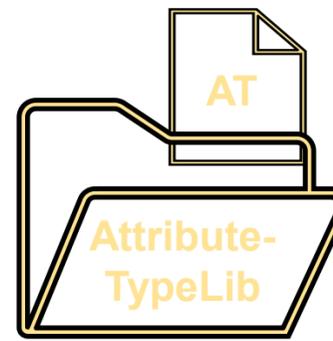
Komponenten RoboScht Zelle

Die RoboScht-Anlage besteht aus....

- | | |
|--|----------------------|
| 1. ABB IRB 4600 | (Roboter) |
| 2. Kabelhalterungen | (Materialhandhabung) |
| 3. Arbeitstisch | (Materialhandhabung) |
| 4. Robotersteuerung IRC5 | (Steuerungseinheit) |
| 5. ITEM Schaltschrank-
Aufnahme | (Materialhandhabung) |
| 6. Werkstück CC-1337
mit Klemmleisten | (Produktmuster) |
| 7. Endeffektor | (Werkzeug) |

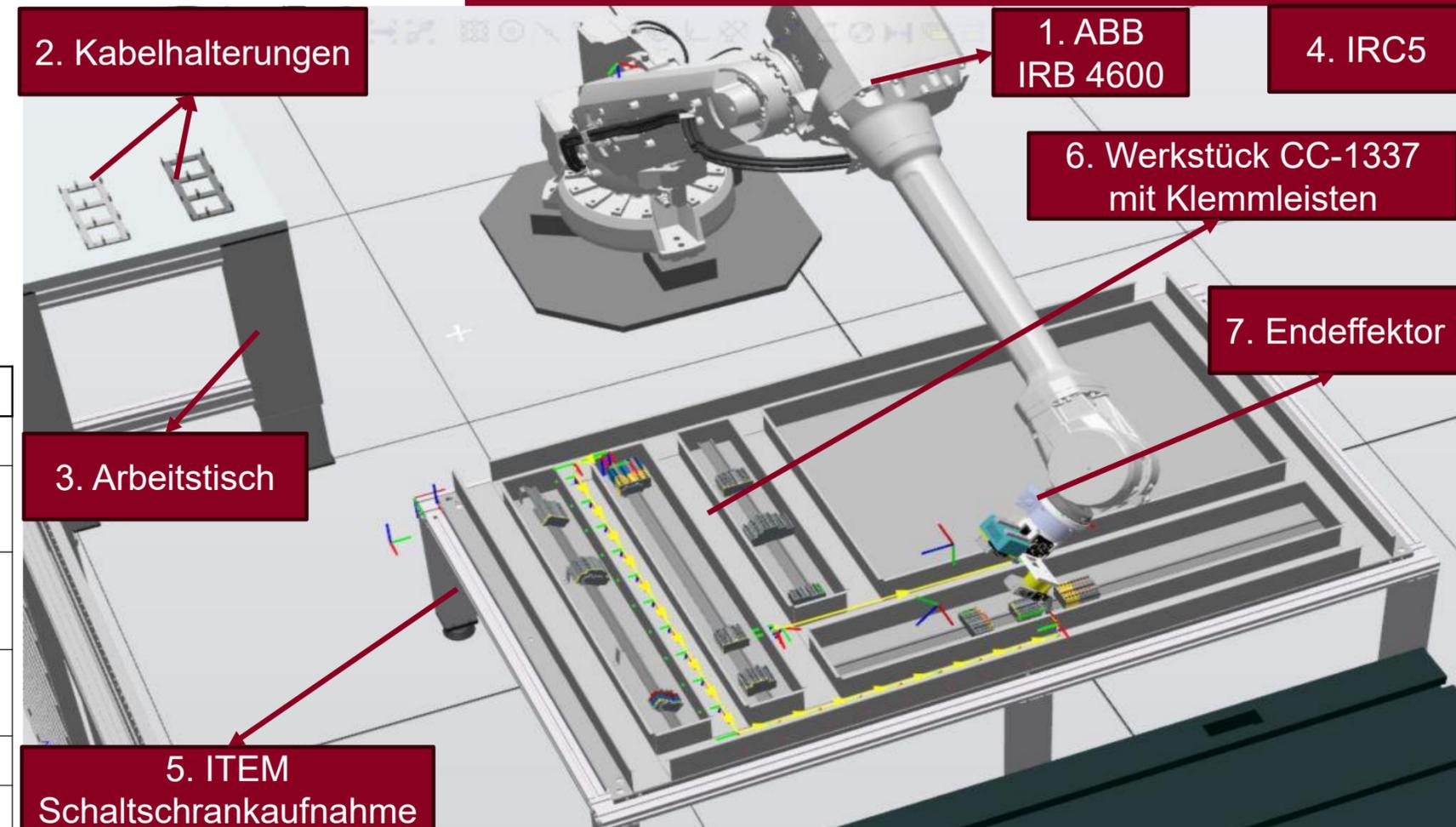


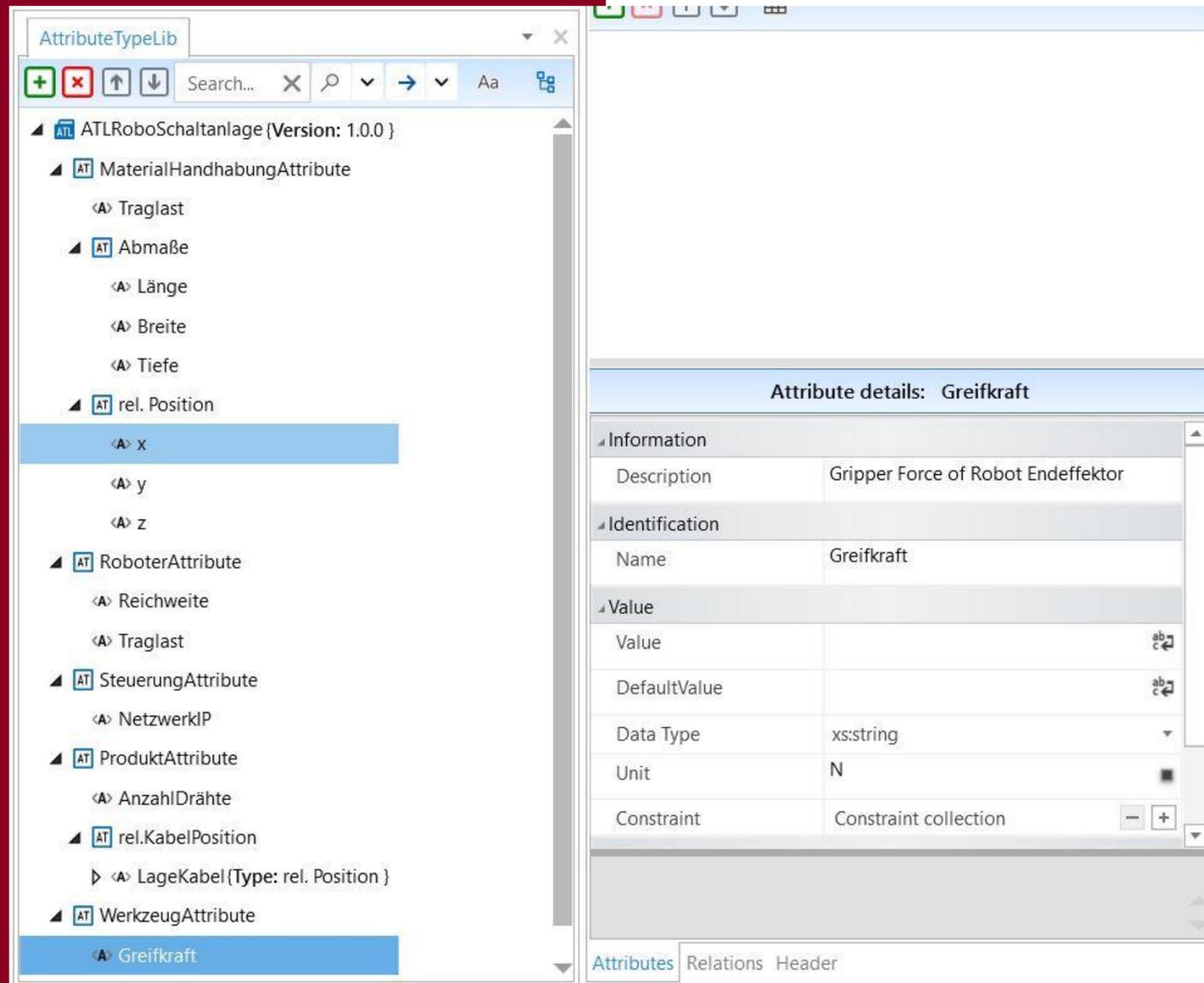
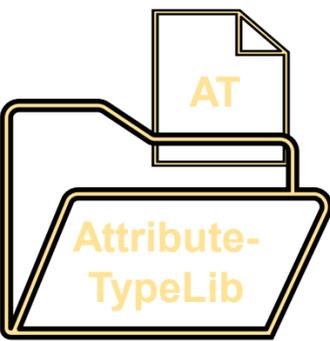
Lösung: RoboSchtalt Zelle



AttributeType und Attribute

Akteur:	AttributeType:	Attribute:
Arbeits Tisch	MaterialhandhabungAttribute	Abmaße LxBxT, Traglast
Kabelhalterung	MaterialhandhabungAttribute	Abmaße LxBxT, Koordinaten x, y, z
ABB IRB 4600	RoboterAttribute	Reichweite, Traglast
ITEM Schaltschrankaufnahme	MaterialhandhabungAttribute	Abmaße LxBxT, rel. Position x, y, z
Robotersteuerung IRC5	SteuerungAttribute	Netzwerk IP: Anzahl Verdrahtungen, Lage xx, yy, zz
Muster CC-1337	ProduktAttribute	
Roboschtalt Effektor	WerkzeugAttribute	Greifkraft





AttributeType/Attribute

AttributeTypeLib: Zelle

- AttributeTypeLib gemäß zugehörigem funktionalem Objekt
- Untergliederung in AT zusammenhängender Struktur (Positionen als Koordinaten, Abmaße als Triplet etc.)

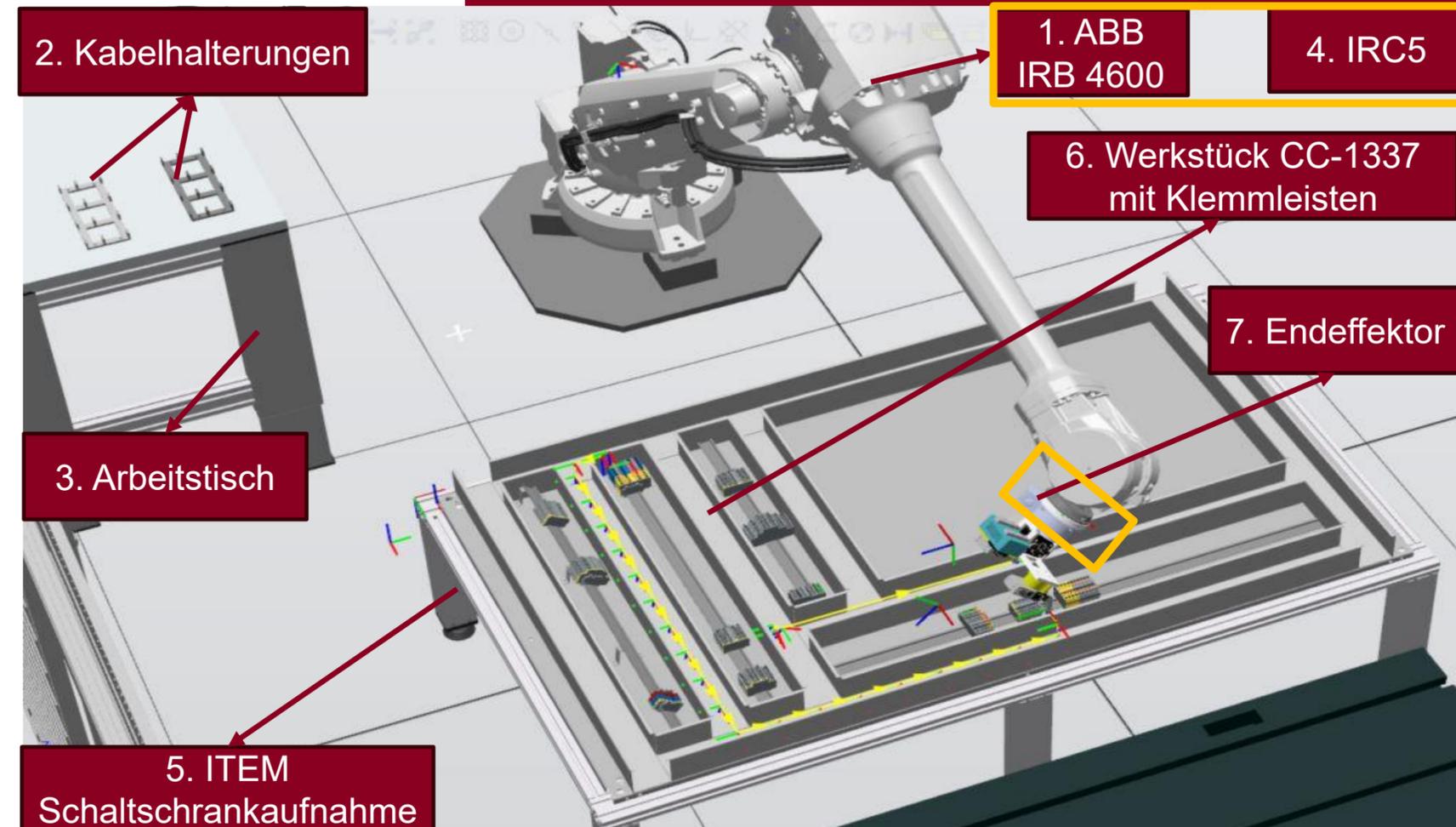
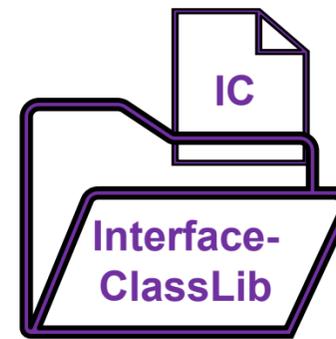
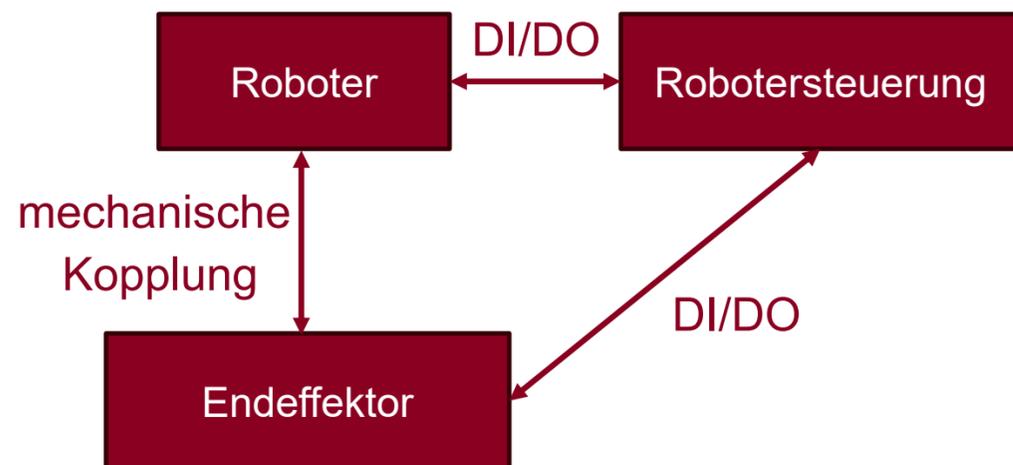
Constraints:

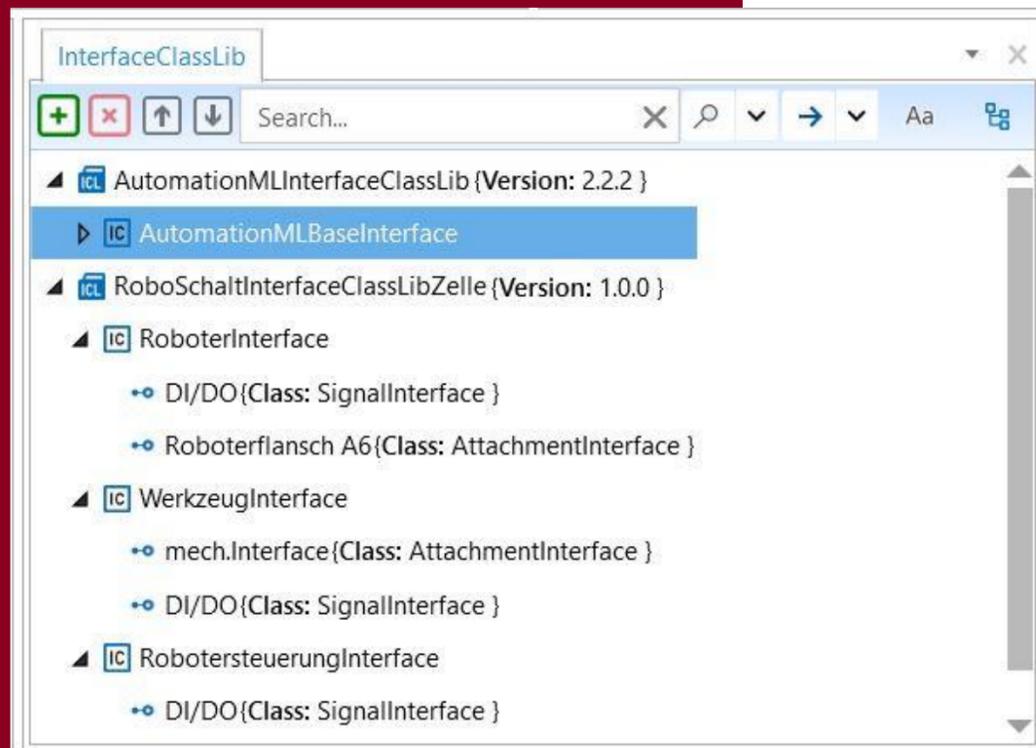
Attribute:	Constraint:	Einheit:
Reichweite	> 2	m
Positionsgenauigkeit	Abweichung < 10	mm
Standard IP Adressenbereich	zwischen XX.XXX.130.000 und XX.XXX.134.000	-

Schnittstellen: RoboSchtalt Zelle

Vereinfachte Schnittstellen Übersicht:

- DI / DO = Digitale In-/Output Schnittstelle
- mechanische Kopplung an Roboterflansch Achse 6





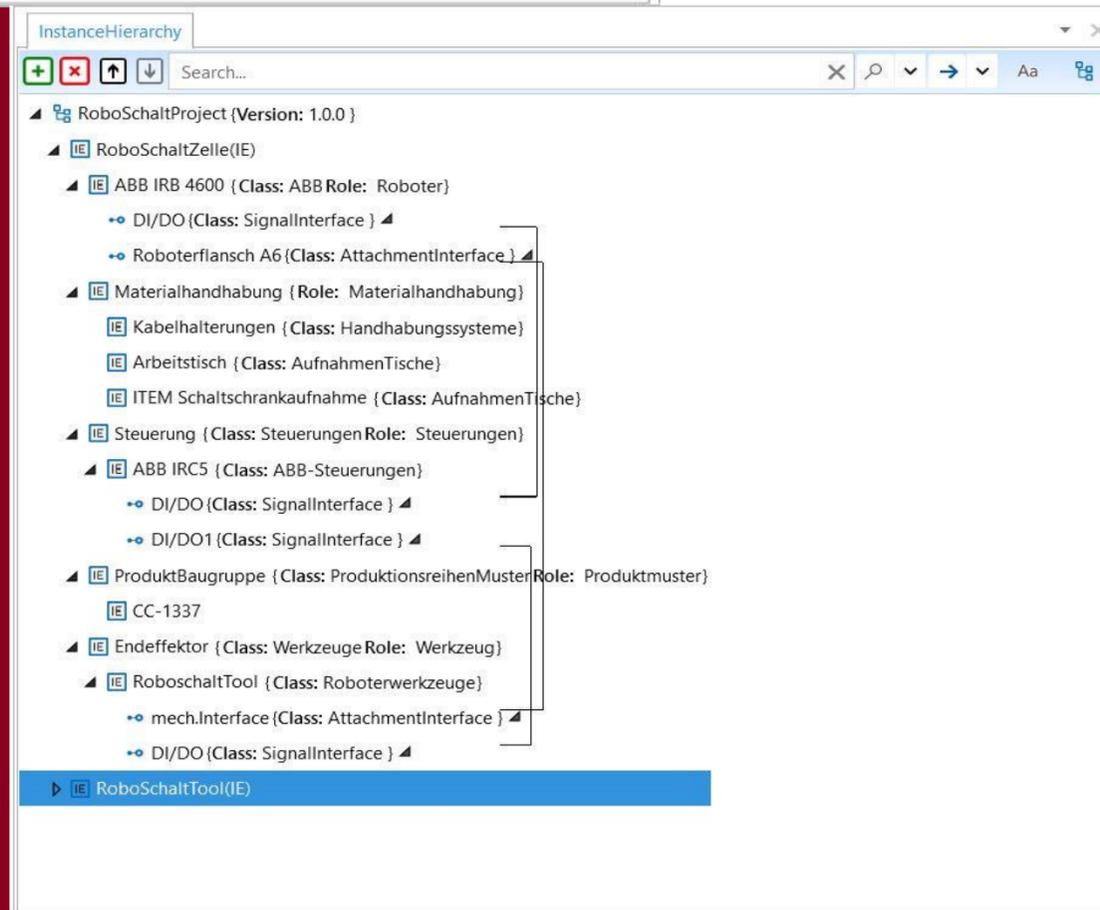
Lösung: InterfaceClass

InterfaceClassLib: Zelle

- Klassifikation gemäß funktionaler Zugehörigkeit
- allg. Schnittstellen die nicht herstellerspezifisch definiert sind

InternalLinks: Zelle

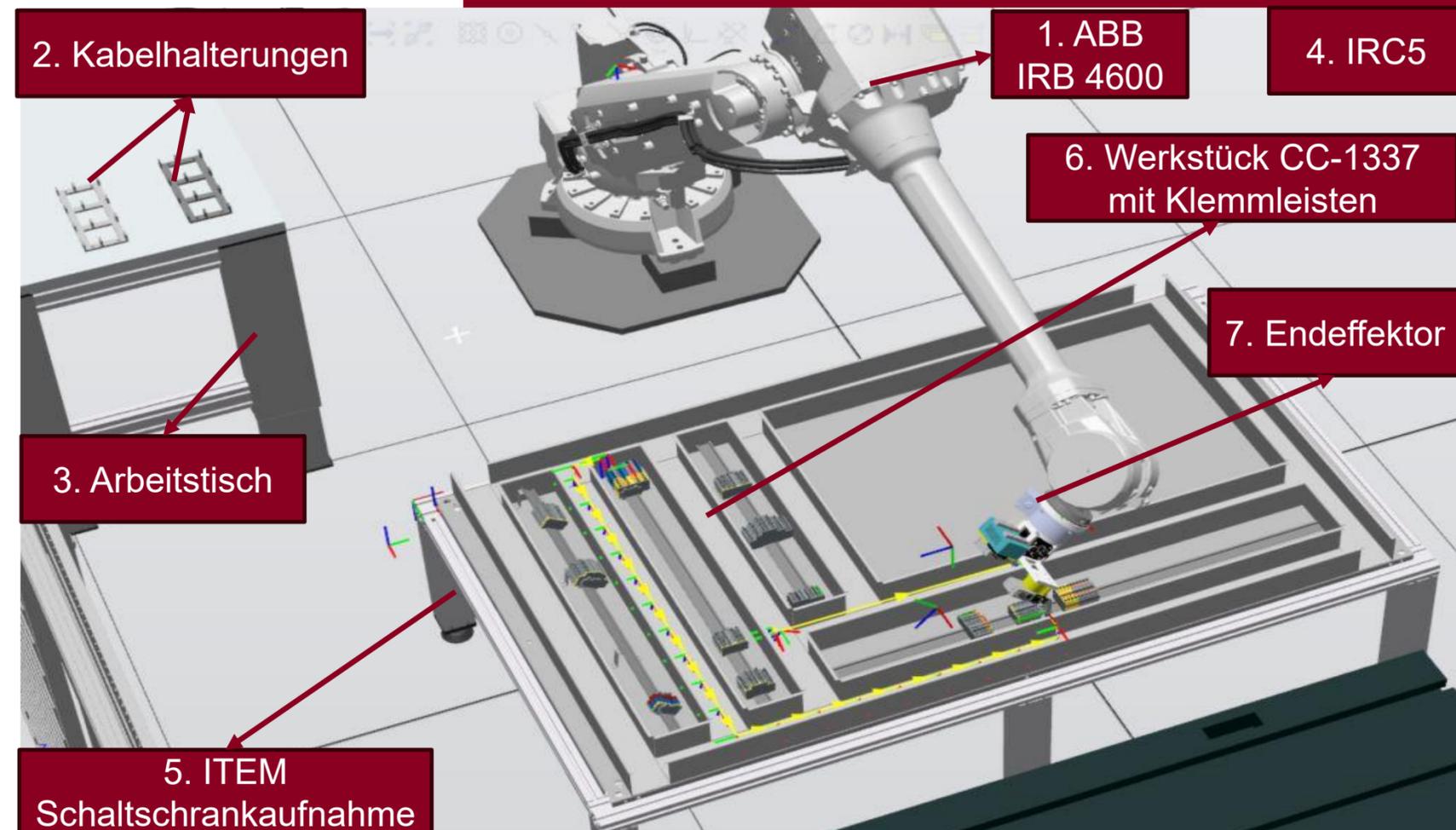
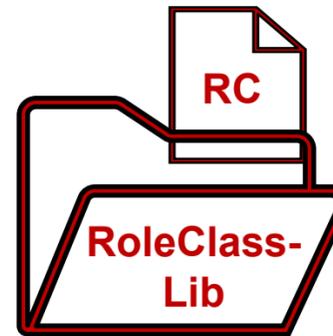
- Mechanische Schnittstellen Werkzeug und IRB 4600
- Digitale Schnittstellen Werkzeug und IRC 5
- Digitale Schnittstellen IRB 4600 zu IRC 5

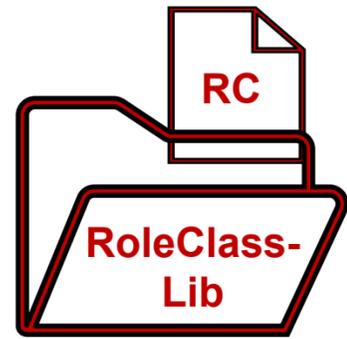


Rollen Roboschalt Zelle

Abstraktion der Systemakteure

- ABB IRB 4600 -> Roboter
- Kabelhalterungen -> Materialhandhabung
- Arbeitstisch -> Materialhandhabung
- Robotersteuerung IRC5 -> Steuerungen
- Werkstück CC-1337 -> Produktmuster
- Endeffektor -> Werkzeuge





Lösung: RoleClass

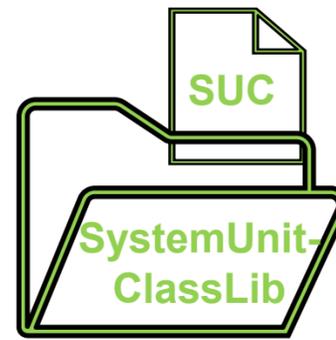
RoleClassLib: Zelle

- Abstraktion gemäß funktionaler Sicht
- Rollenklassen gemäß technischen Funktionen innerhalb der Fertigungszelle

Attribute & External Interfaces der RC:

- Keine Attribute aufgrund entsprechender Definition der ATL & SUC
- Keine External Interfaces aufgrund der Abbildung in den IC

Lösung: Roboschalt SystemUnitClass

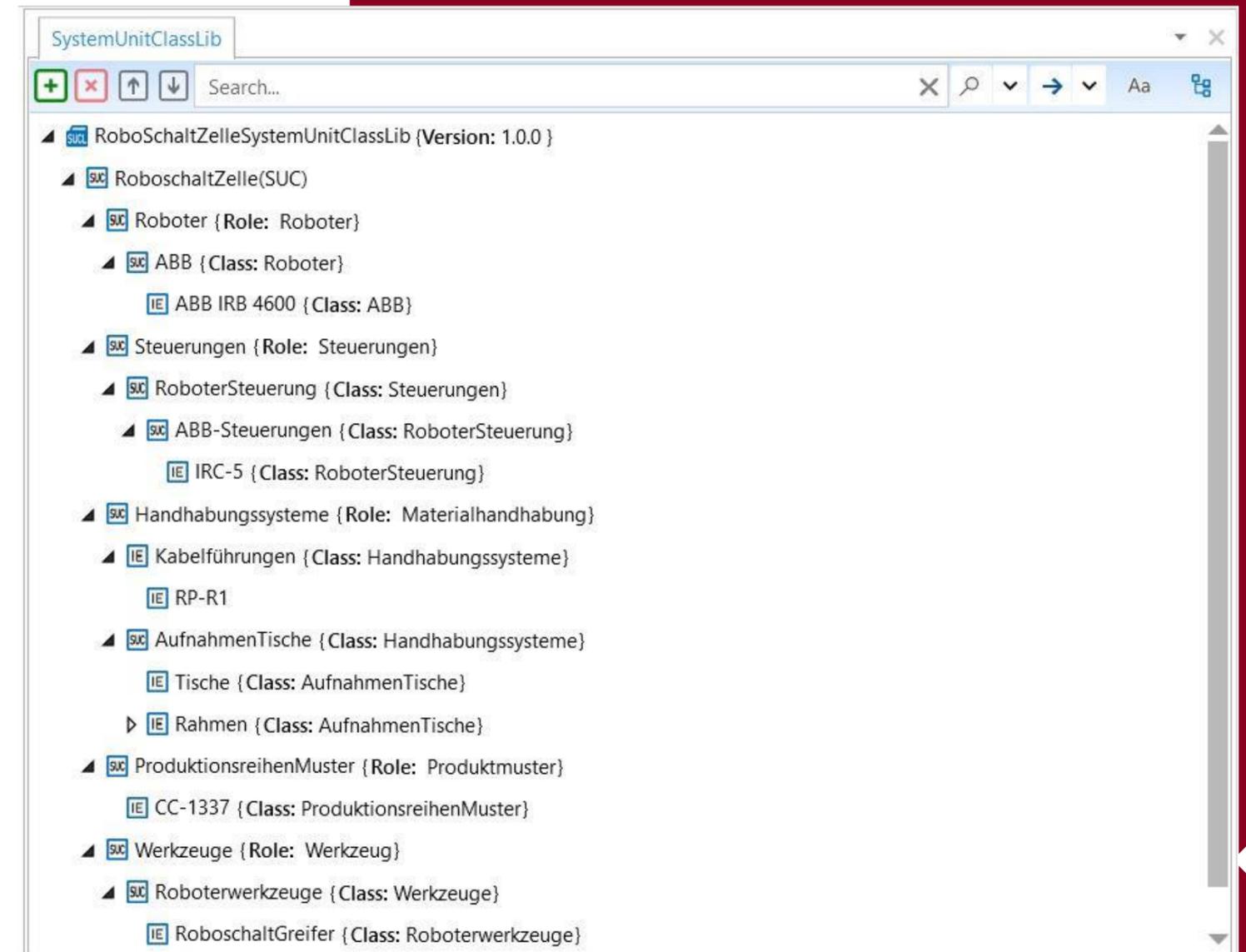


SystemUnitClassLib (SUCL): Zelle

- Gliederung nach funktionaler Baugruppe (Ebene1 und 2)
- Gliederung nach Hersteller Ebene 3
- Alle weiteren Ebenen stellen Detail Ebenen für Serien oder Modellbezeichnungen dar

Vererbung & Rollen (Ausblick):

- Rollen ergeben sich aus Semantik zur RoleClassLib
- Semantik und Klassenbezüge zur Vererbung ergeben sich nicht implizit und werden explizit hinzugefügt
- Vererbung erfolgt Vater Kind basiert





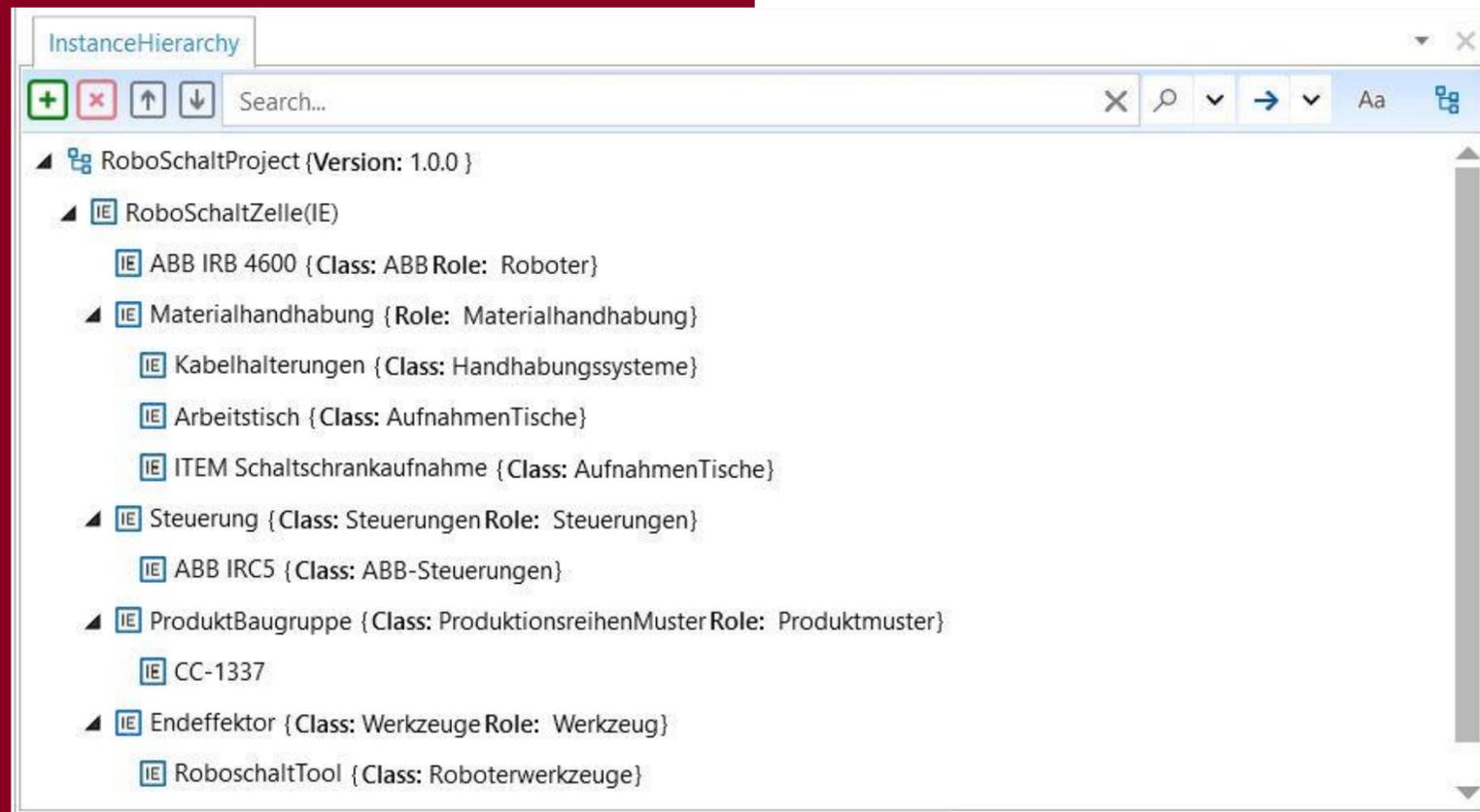
InstanceHierarchy

InstanceHierarchy: Zelle

- Ebene 1: Projekt
- Ebene 2: Funktionseinheiten Prozess
- Ebene 3: Anlagen & Baugruppen
- Ebene 4: IE mit Subfunktionen oder Untergeordneten Objekten
- Vererbung und Rollenverweise basierend auf InternalLinks zu IC, SUC RC und ATL

Modellierungsstrategie:

- gewählte Methode 1: Jede Anlage und Funktionseinheit erhält eigenes IE
- Entwurf in Methode 2 über SUC mit nur einem IE als Referenz ebenfalls möglich



AML-File mit dem Datenmodell der RoboSchtalt Zelle

The screenshot displays the AML development environment with several windows showing class hierarchies and attribute details.

InstanceHierarchy (RoboSchtaltProject (Version: 1.0.0))

- RoboSchtaltZelle(IE)
 - ABB IRB 4600 (Class: ABB Role: Roboter)
 - DI/DO (Class: SignallInterface)
 - Roboterflansch A6 (Class: AttachmentInterface)
 - Materialhandhabung (Role: Materialhandhabung)
 - Kabelhalterungen (Class: Handhabungssysteme)
 - Arbeitstisch (Class: AufnahmenTische)
 - ITEM Schaltschrankaufnahme (Class: AufnahmenTische)
 - Steuerung (Class: SteuerungenRole: Steuerungen)
 - ABB IRC5 (Class: ABB-Steuerungen)
 - DI/DO (Class: SignallInterface)
 - DI/DO1 (Class: SignallInterface)
 - ProduktBaugruppe (Class: ProduktionsreihenMuster Role: Produktmuster)
 - CC-1337
 - Endeffektor (Class: WerkzeugeRole: Werkzeug)
 - RoboSchtaltTool (Class: Roboterwerkzeuge)
 - mech.Interface (Class: AttachmentInterface)
 - DI/DO (Class: SignallInterface)
- RoboSchtaltTool(IE)

SystemUnitClassLib (RoboSchtaltZelleSystemUnitClassLib (Version: 1.0.0))

- RoboSchtaltZelle(SUC)
 - Roboter (Role: Roboter)
 - ABB (Class: Roboter)
 - ABB IRB 4600 (Class: ABB)
 - Steuerungen (Role: Steuerungen)
 - RoboterSteuerung (Class: Steuerungen)
 - ABB-Steuerungen (Class: RoboterSteuerung)
 - IRC-5 (Class: RoboterSteuerung)
 - Handhabungssysteme (Role: Materialhandhabung)
 - Kabelführungen (Class: Handhabungssysteme)
 - RP-R1
 - AufnahmenTische (Class: Handhabungssysteme)
 - Tische (Class: AufnahmenTische)
 - Rahmen (Class: AufnahmenTische)
 - ITEM 8x40x40
 - ProduktionsreihenMuster (Role: Produktmuster)
 - CC-1337 (Class: ProduktionsreihenMuster)
 - Werkzeuge (Role: Werkzeug)
 - Roboterwerkzeuge (Class: Werkzeuge)
 - RoboSchtaltGreifer (Class: Roboterwerkzeuge)
- RoboSchtaltTool(SUC)

RoleClassLib (RoboSchtaltRoleClassLib (Version: 1.0.0))

- AutomationMLBaseRoleClassLib (Version: 2.2.2)
- RoboSchtaltRoleClassLibZelle (Version: 1.0.0)
 - Roboter
 - Materialhandhabung
 - Steuerungen
 - Produktmuster
 - Werkzeug
- RoboSchtaltRoleClassLibTool (Version: 1.0.0)

InterfaceClassLib (RoboSchtaltInterfaceClassLib (Version: 1.0.0))

- AutomationMLInterfaceClassLib (Version: 2.2.2)
- RoboSchtaltInterfaceClassLibZelle (Version: 1.0.0)
 - RoboterInterface
 - DI/DO (Class: SignallInterface)
 - Roboterflansch A6 (Class: AttachmentInterface)
 - WerkzeugInterface
 - mech.Interface (Class: AttachmentInterface)
 - DI/DO (Class: SignallInterface)
 - RobotersteuerungInterface
 - DI/DO (Class: SignallInterface)
- RoboSchtaltInterfaceClassLibTool (Version: 1.0.0)

AttributeTypeLib (ATLRoboSchtaltanlage (Version: 1.0.0))

- MaterialHandhabungAttribute
 - Traglast
 - Abmaße
 - rel. Position
 - x
 - y
 - z
- RoboterAttribute
 - Reichweite
 - Traglast
- SteuerungAttribute
 - NetzwerkIP
- ProduktAttribute
 - AnzahlDrahte
 - rel.KabelPosition
 - LageKabel (Type: rel. Position)
- WerkzeugAttribute
 - Greifkraft

- ATLRoboSchtaltTool (Version: 1.0.0)

Attributes : rel. Position

- x
- y
- z

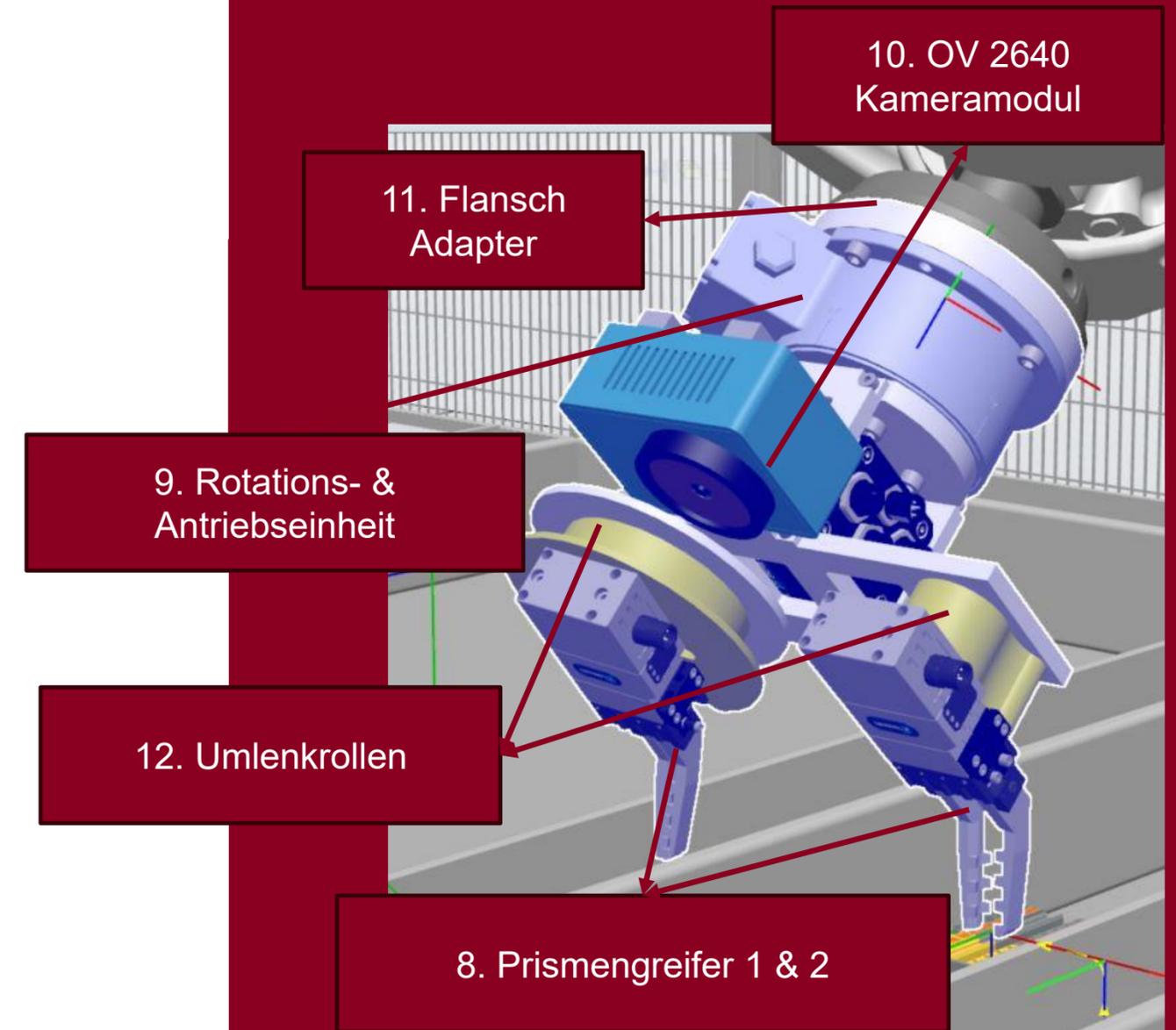
Attribute details: rel. Position

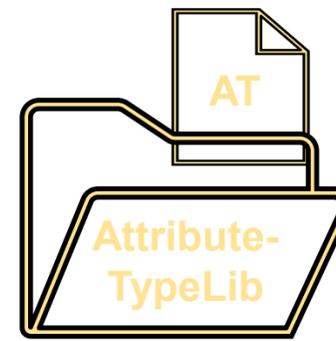
Attribute details: rel. Position	
Information	
Description	relative Position compared to Robot Base
Identification	
Name	rel. Position
Value	
Value	
DefaultValue	
Data Type	xs:string
Unit	
Constraint	Constraint collection
Relations	
Semantic	Semantic collection
Attribute Type	Drop ...

Komponenten RoboScht Werkzeug

Der Endeffektor besteht aus....

1. Prismengreifer 1 & 2 (Handhabungssystem)
2. Rotations- & Antriebseinheit (Handhabungssystem)
3. OV2640 Kameramodul (Sensorsystem)
4. Flansch Adapter (mechanische Verbindung)
5. Umlenkrollen/Kabelführung (Materialführung)
6. Plattform/Rahmen (mechanische Verbindung)

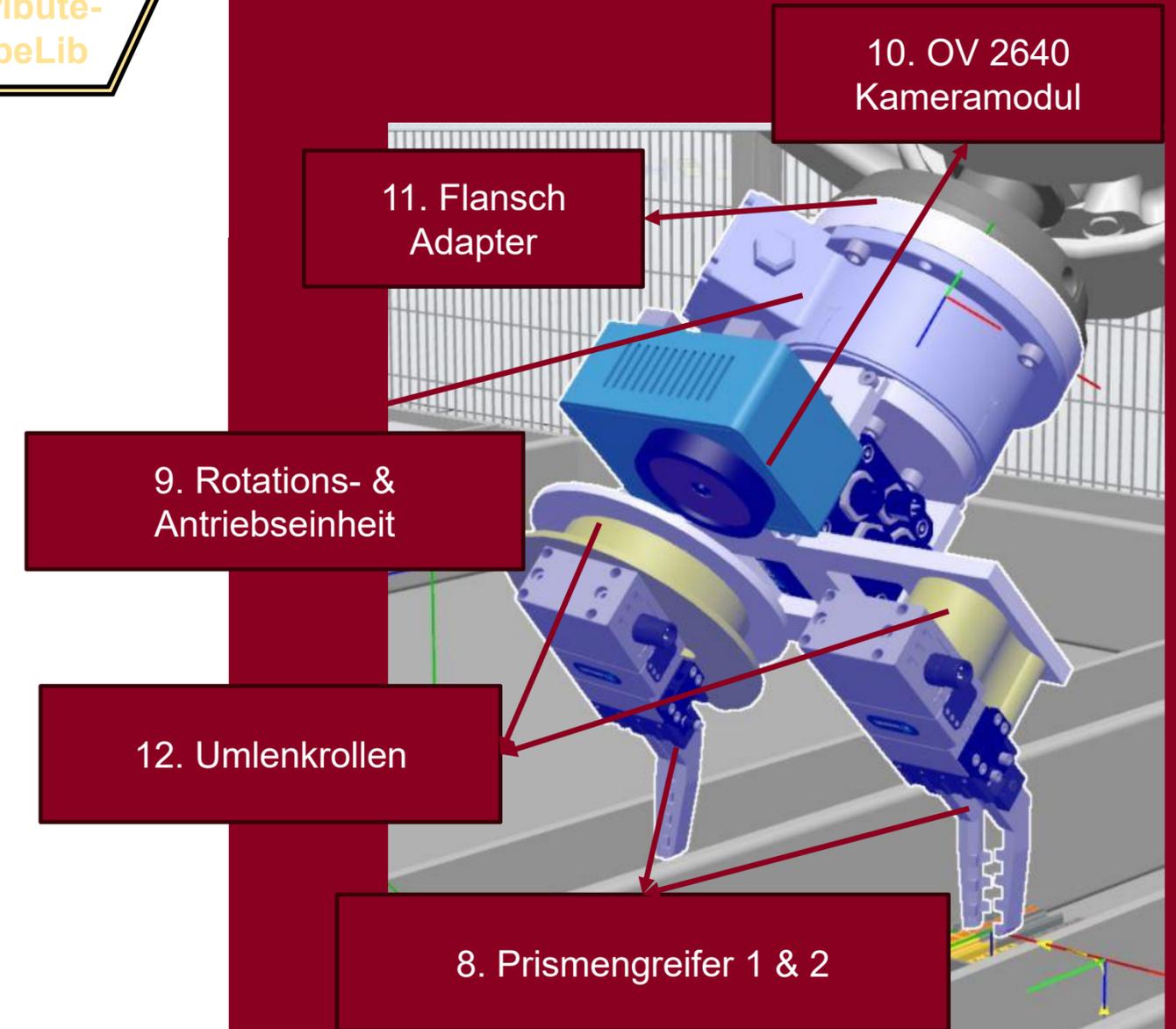


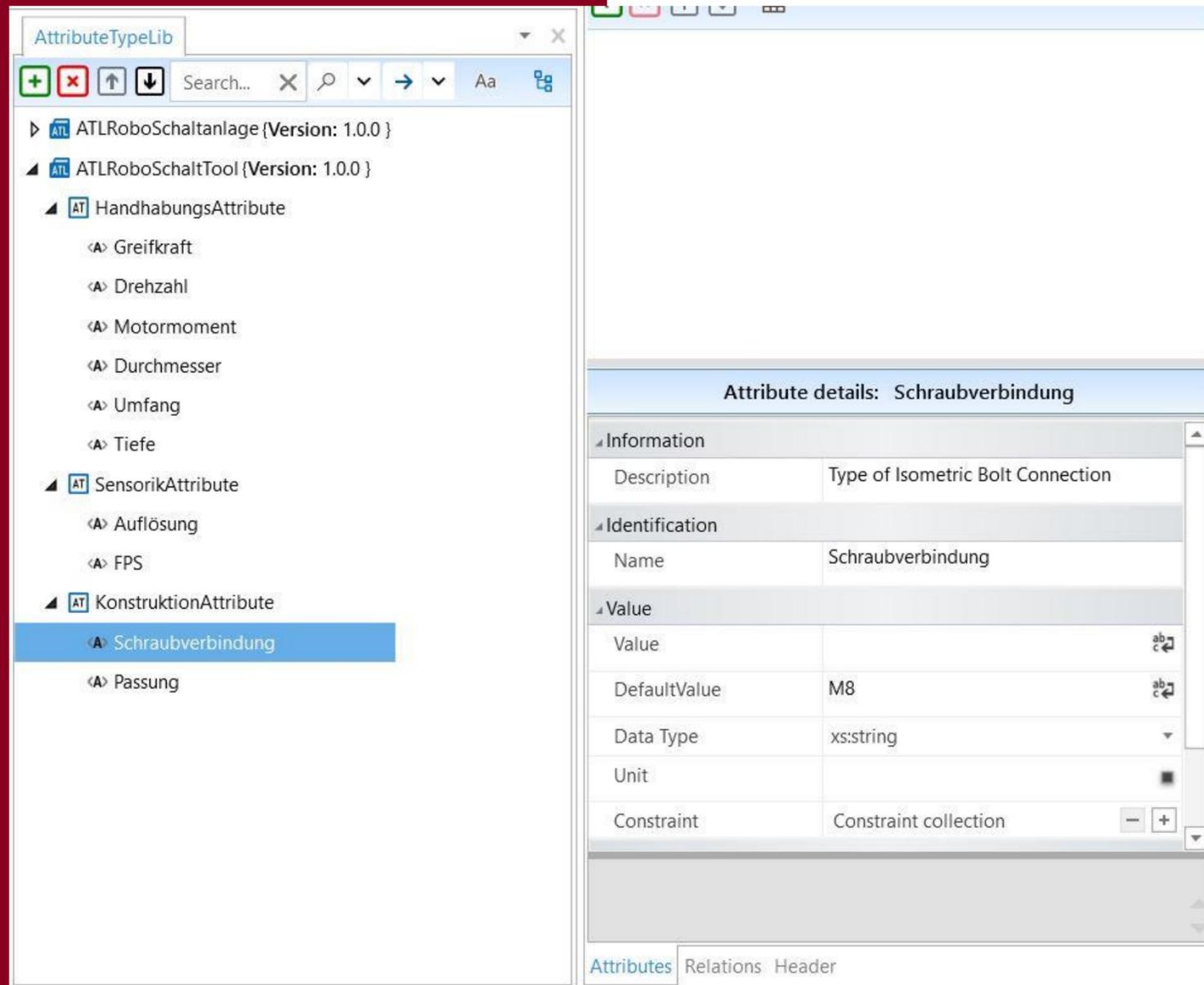
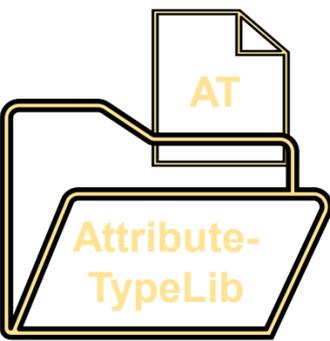


Lösung: RoboScht Werkzeug

AttributeType und Attribute

Akteur:	AttributeType:	Attribute:
Prismengreifer 1	HandhabungsAttribute	Greifkraft
Prismengreifer 2	HandhabungsAttribute	Greifkraft
Rotations- & Antriebseinheit	HandhabungsAttribute	Drehzahl, Motormoment
OV 2640 Kameramodul	SensorikAttribute	Auflösung, FPS
mech. Flanschadapter	KonstruktionAttribute	Schraubverbindung Isometrisch, Passung
Umlenkrollen	HandhabungsAttribute	Durchmesser, Umfang, Tiefe





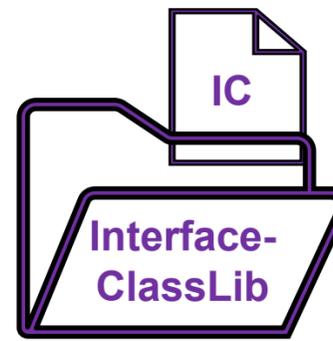
AttributeType/Attribute

AttributeTypeLib: Werkzeug

- AttributeTypeLib gemäß zugehörigem funktionalem Objekt
- Untergliederung in AT Zusammenhängender Struktur (Positionen als Koordinaten, Abmaße als Triplet etc.)

Constraints:

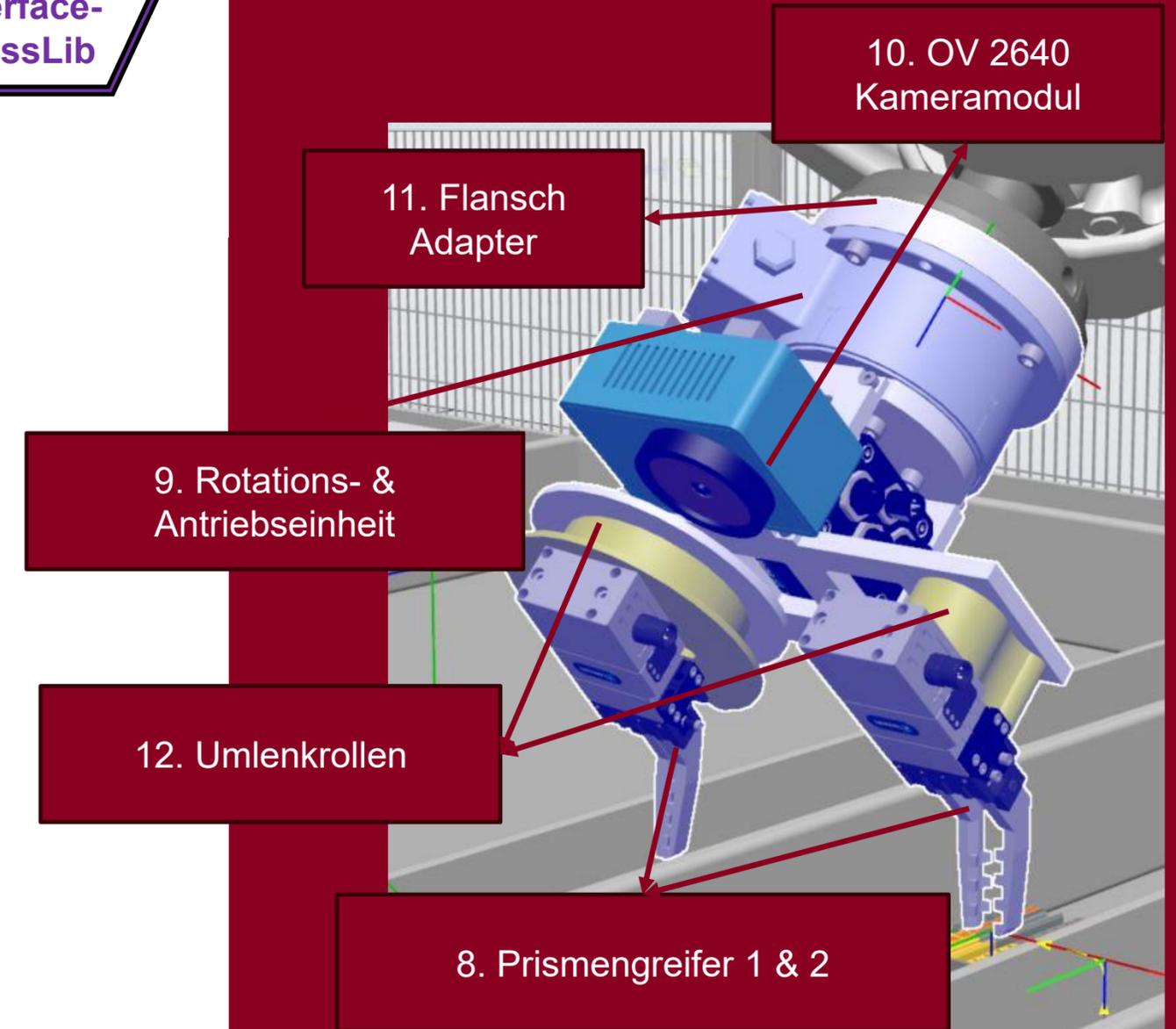
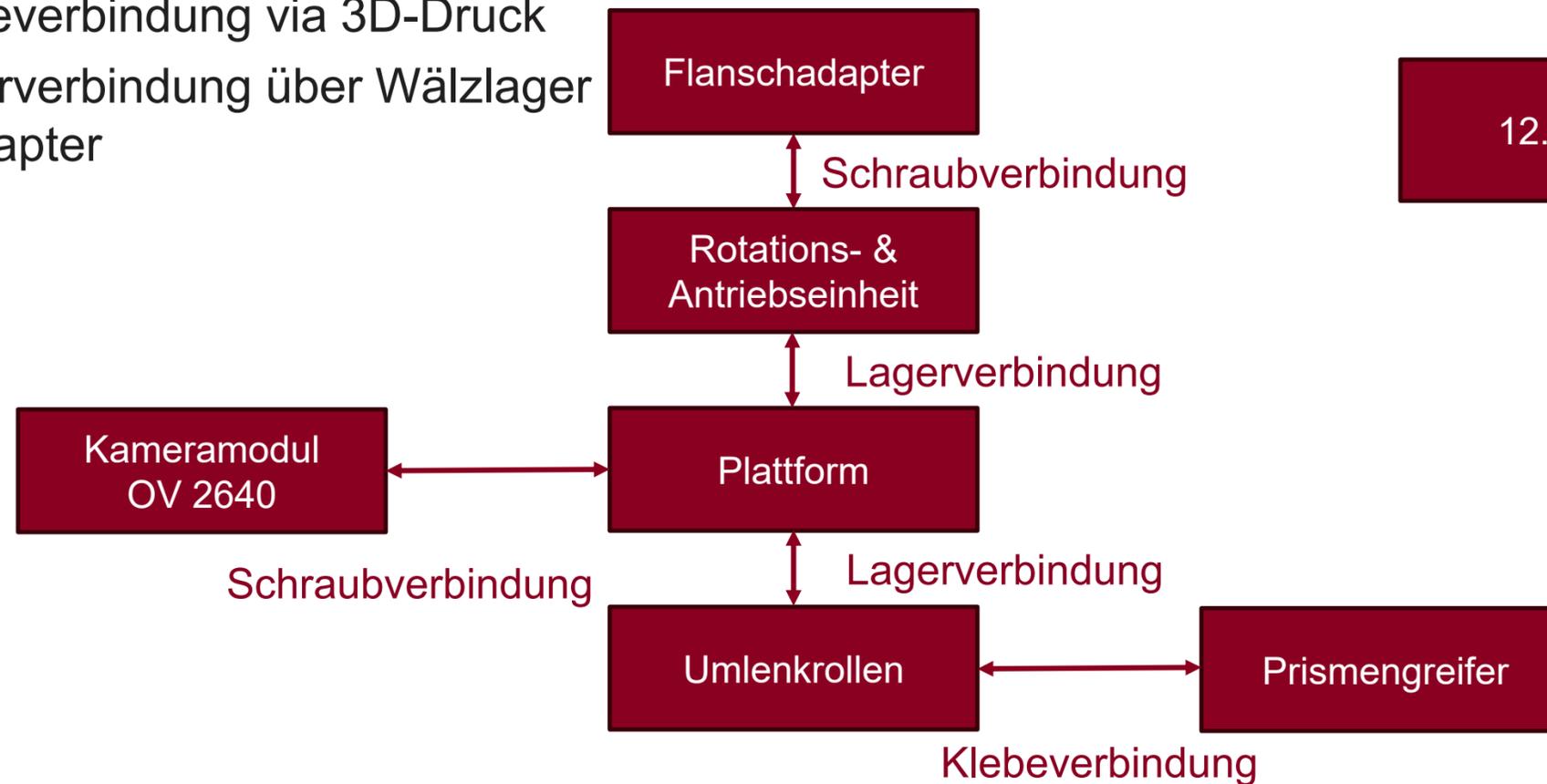
Attribute:	Constraint:	Einheit:
FPS	mind. 30	1/s
Schraubverbindungen	mind. ISO-M4 ; max. ISO-M8	-
Greifkraft	F < 100	N

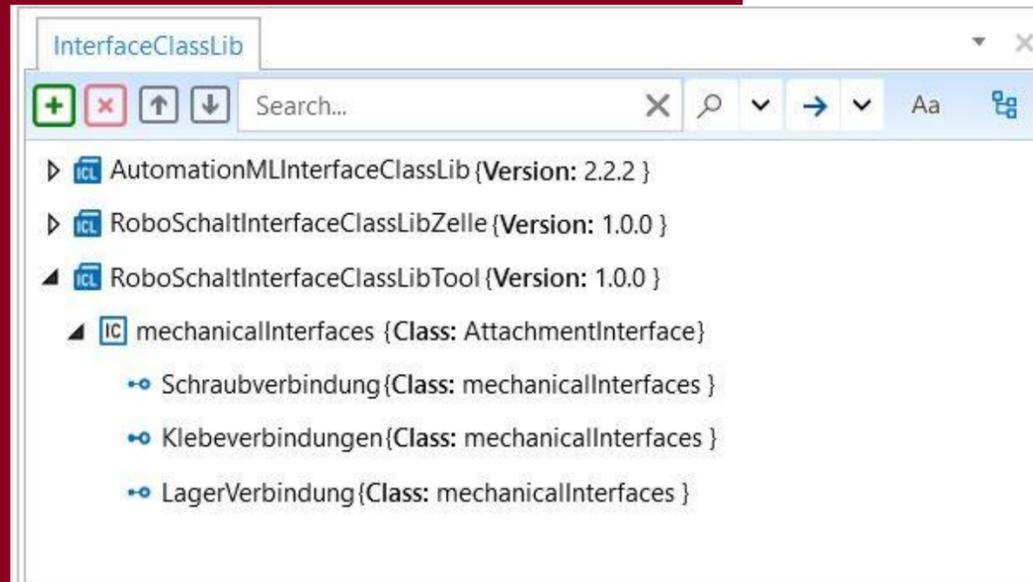
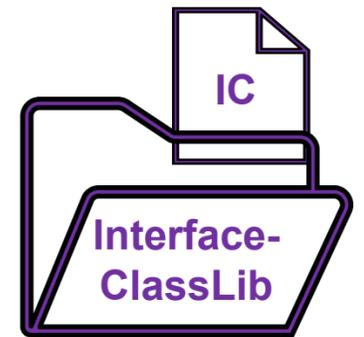


Schnittstellen: RoboSchaft Werkzeug

Vereinfachte Schnittstellen Übersicht:

- Schraubverbindung mittels isometrischer Innensechskantschrauben
- Klebeverbindung via 3D-Druck
- Lagerverbindung über Wälzlager & Adapter





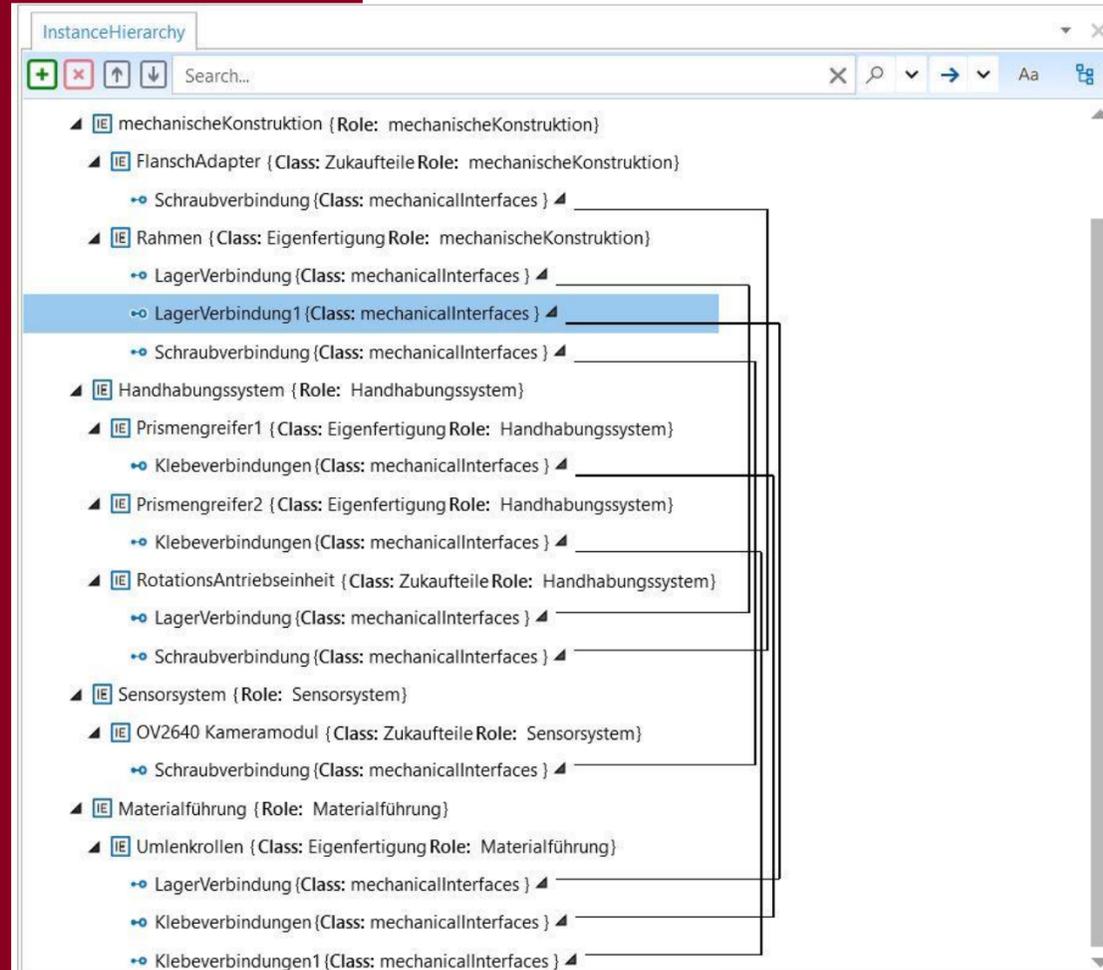
Lösung: InterfaceClass

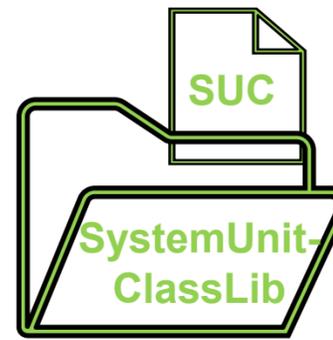
InterfaceClassLib: Werkzeug

- Klassifikation gemäß Art der mechanischen Kopplung
- Schnittstellen gemäß Schrauben und Wälzlager Normen
 - Hinterlegen von Normen via IE mit Dateiverweis in AML möglich (Ausblick)

InternalLinks:

- Schraubverbindungen Flanschadapter & Sensormodul, sowie Antriebseinheit
- gelagerte Verbindungen Rotations-& Antriebseinheit mit Plattform/Rahmen
- gelagerte Verbindung Umlenkrollen mit Plattform/Rahmen
- Klebeverbindung zwischen Umlenkrolle & Prismengreifer





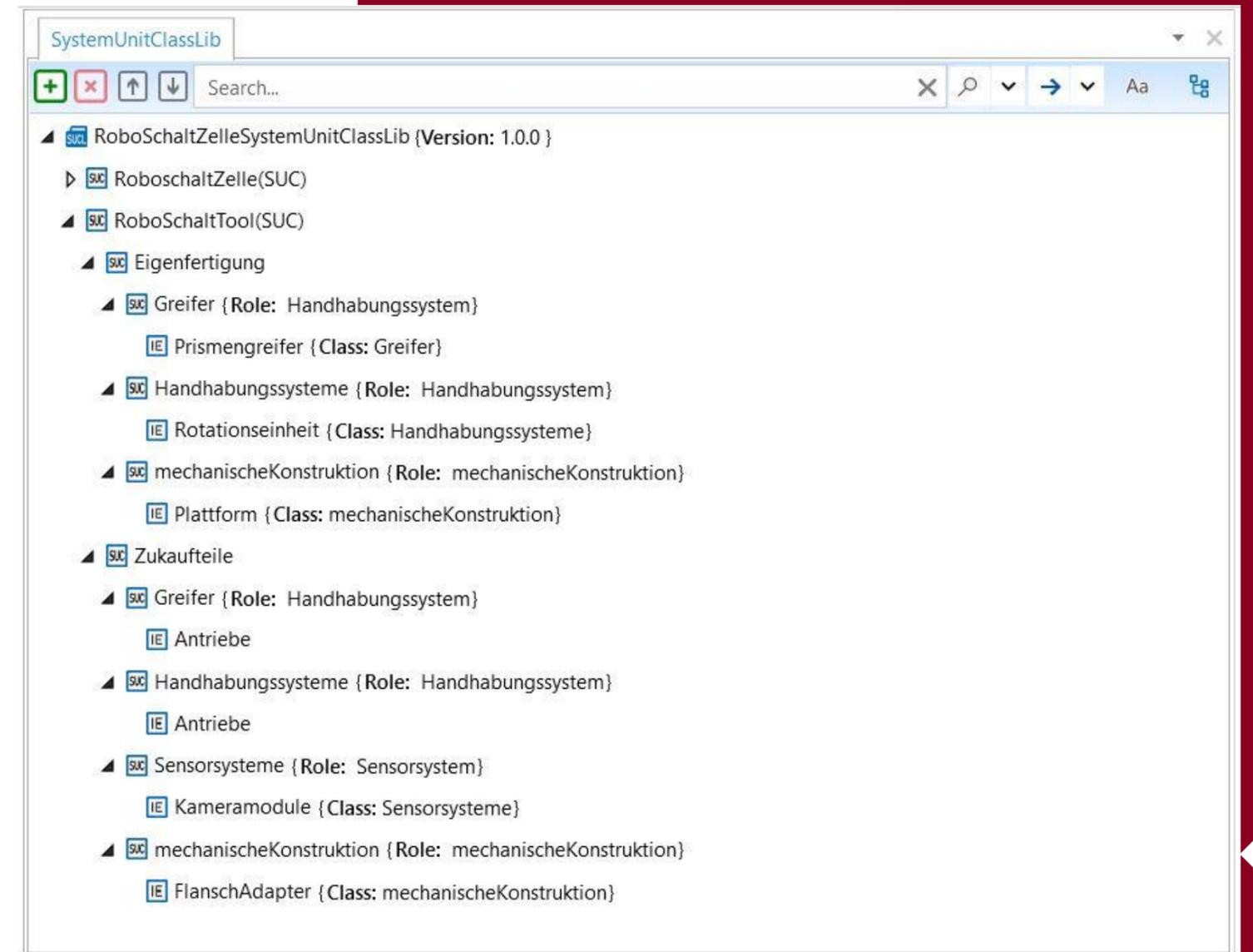
Lösung: Roboschalt SystemUnitClass

SystemUnitClassLib (SUC): Werkzeug

- Gliederung nach Eigenfertigung & Fremdfertigung (Ebene 1)
- Gliederung Baugruppe (Ebene 2)
- Alle weiteren Ebenen definieren Detail Ebenen der Komponenten

Vererbung & Rollen (Ausblick):

- Rollen ergeben sich aus Semantik zur RoleClassLib
- Semantik und Klassenbezüge zur Vererbung ergeben sich nicht implizit und werden explizit hinzugefügt
- Vererbung erfolgt Vater Kind basiert





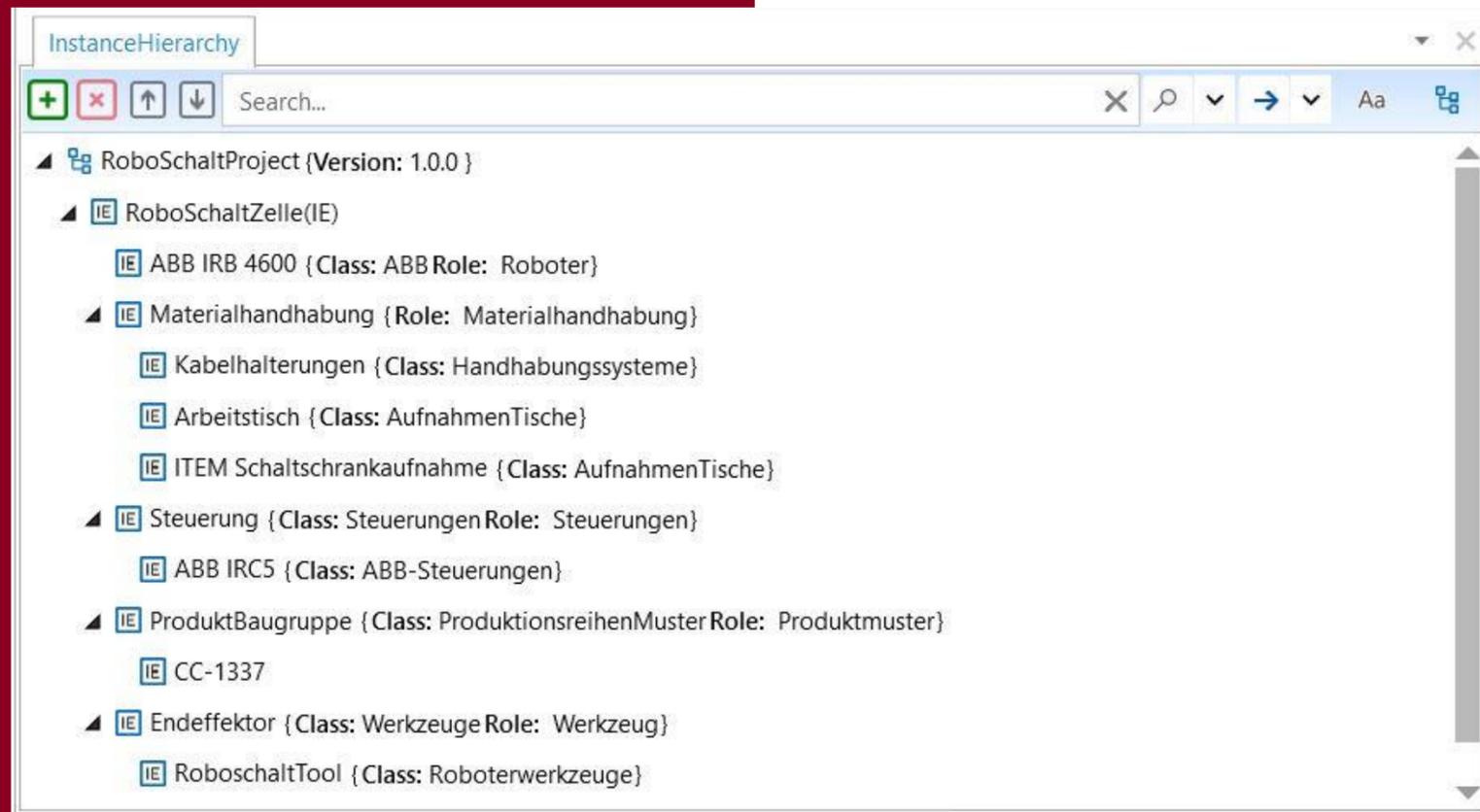
InstanceHierarchy

InstanceHierarchy: Zelle

- Ebene 1: Projekt
- Ebene 2: Funktionseinheiten Prozess
- Ebene 3: Anlagen & Baugruppen
- Ebene 4: IE mit Subfunktionen oder Untergeordneten Objekten
- Vererbung und Rollenverweise basierend auf InternalLinks zu IC, SUC RC und ATL

Modellierungsstrategie:

- gewählte Methode 1: Jede Anlage und Funktionseinheit erhält eigenes IE
- Entwurf in Methode 2 über SUC mit nur einem IE als Referenz ebenfalls möglich



AML-File mit dem Datenmodell des RoboSchaft Werkzeugs

The screenshot displays a software development environment with several windows showing class hierarchies and attribute details.

InstanceHierarchy: Shows a tree structure of classes and their instances. Key classes include:

- RoboSchaftTool(E)
- mechanischeKonstruktion (Role: mechanischeKonstruktion)
- FlanschAdapter (Class: Zukaufteile Role: mechanischeKonstruktion)
- Rahmen (Class: Eigenfertigung Role: mechanischeKonstruktion)
- Handhabungssystem (Role: Handhabungssystem)
- Sensorsystem (Role: Sensorsystem)
- Materialführung (Role: Materialführung)

SystemUnitClassLib: Shows a tree structure of classes and their instances. Key classes include:

- RoboSchaftZelleSystemUnitClassLib (Version: 1.0.0)
- RoboSchaftZelle(SUC)
- RoboSchaftTool(SUC)
- Eigenfertigung
- Zukaufteile
- Sensorsysteme (Class: Zukaufteile Role: Sensorsystem)
- mechanischeKonstruktion (Class: Zukaufteile Role: mechanischeKonstruktion)

RoleClassLib: Shows a tree structure of classes and their instances. Key classes include:

- AutomationMLBaseRoleClassLib (Version: 2.2.2)
- RoboSchaftRoleClassLibZelle (Version: 1.0.0)
- RoboSchaftRoleClassLibTool (Version: 1.0.0)
- Handhabungssystem
- Sensorsystem
- mechanischeKonstruktion
- Materialführung

InterfaceClassLib: Shows a tree structure of classes and their instances. Key classes include:

- AutomationMLInterfaceClassLib (Version: 2.2.2)
- RoboSchaftInterfaceClassLibZelle (Version: 1.0.0)
- RoboSchaftInterfaceClassLibTool (Version: 1.0.0)
- mechanicalInterfaces (Class: AttachmentInterface)

AttributeTypeLib: Shows a tree structure of classes and their instances. Key classes include:

- ATLRoboSchaftanlage (Version: 1.0.0)
- ATLRoboSchaftTool (Version: 1.0.0)
- HandhabungsAttribute
- Greifkraft
- SensorikAttribute
- KonstruktionAttribute

Attributes: Greifkraft: Shows the details of the 'Greifkraft' attribute. The attribute is of type 'xs:string' and has a unit of 'N'. The constraint collection includes an ordinal scaled constraint with a name of 'Gripper Force' and a required value of 10 to 100.

Attribute details: Greifkraft	
Information	
Description	Gripper Force
Identification	
Name	Greifkraft
Value	
Value	
DefaultValue	
Data Type	xs:string
Unit	N
Constraint	
Constraint collection	
[0]	Ordinal scaled constraint
Name	Gripper Force
Required value	
Required minima...	10
Required maxim...	100
Relations	
Semantic	Semantic collection
Attribute Type	Drop ...

Inhaltsübersicht

Einführung

Grundlagen und Definitionen

XML/CAEX/OOP

Der AML-Editor

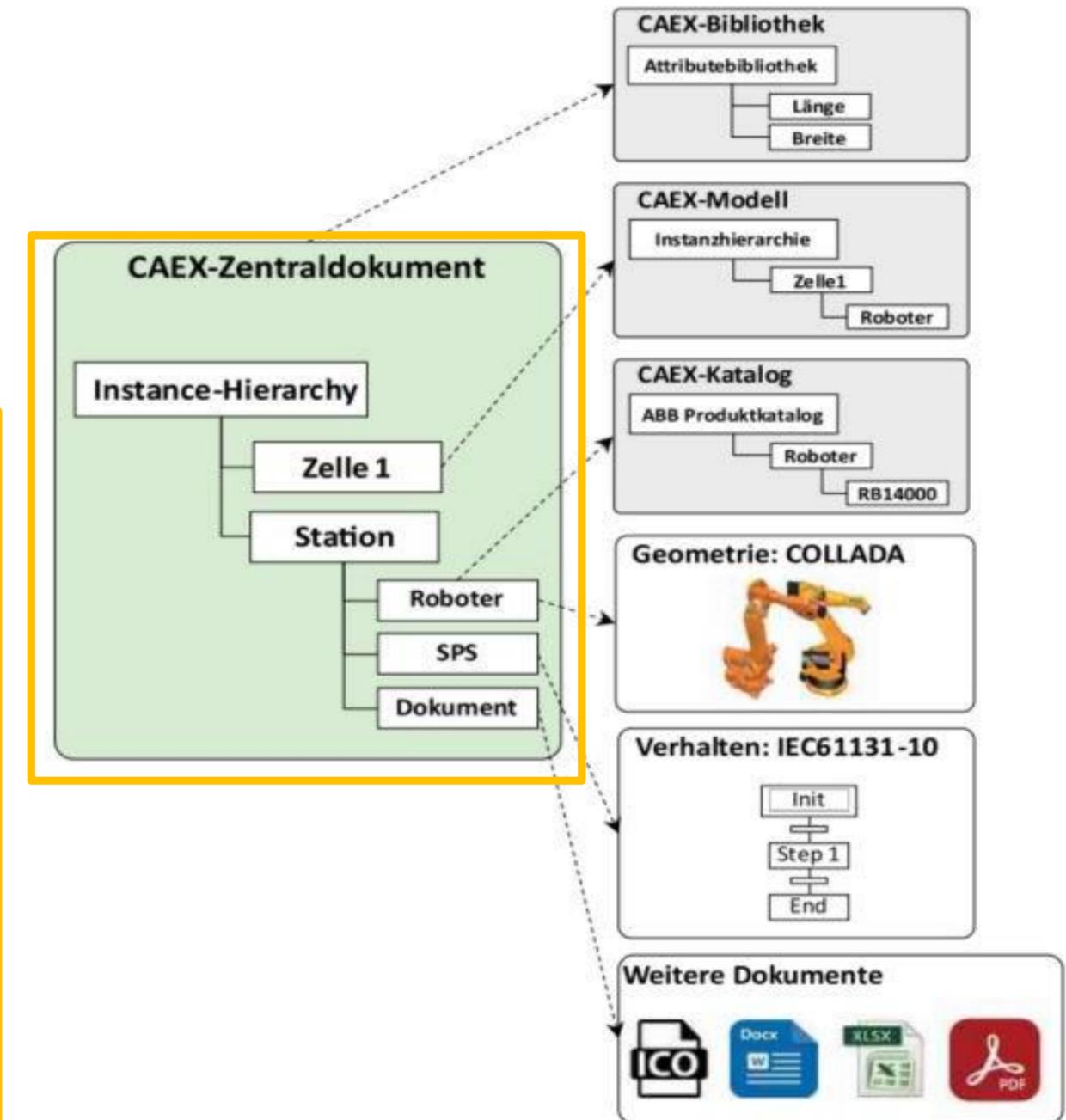
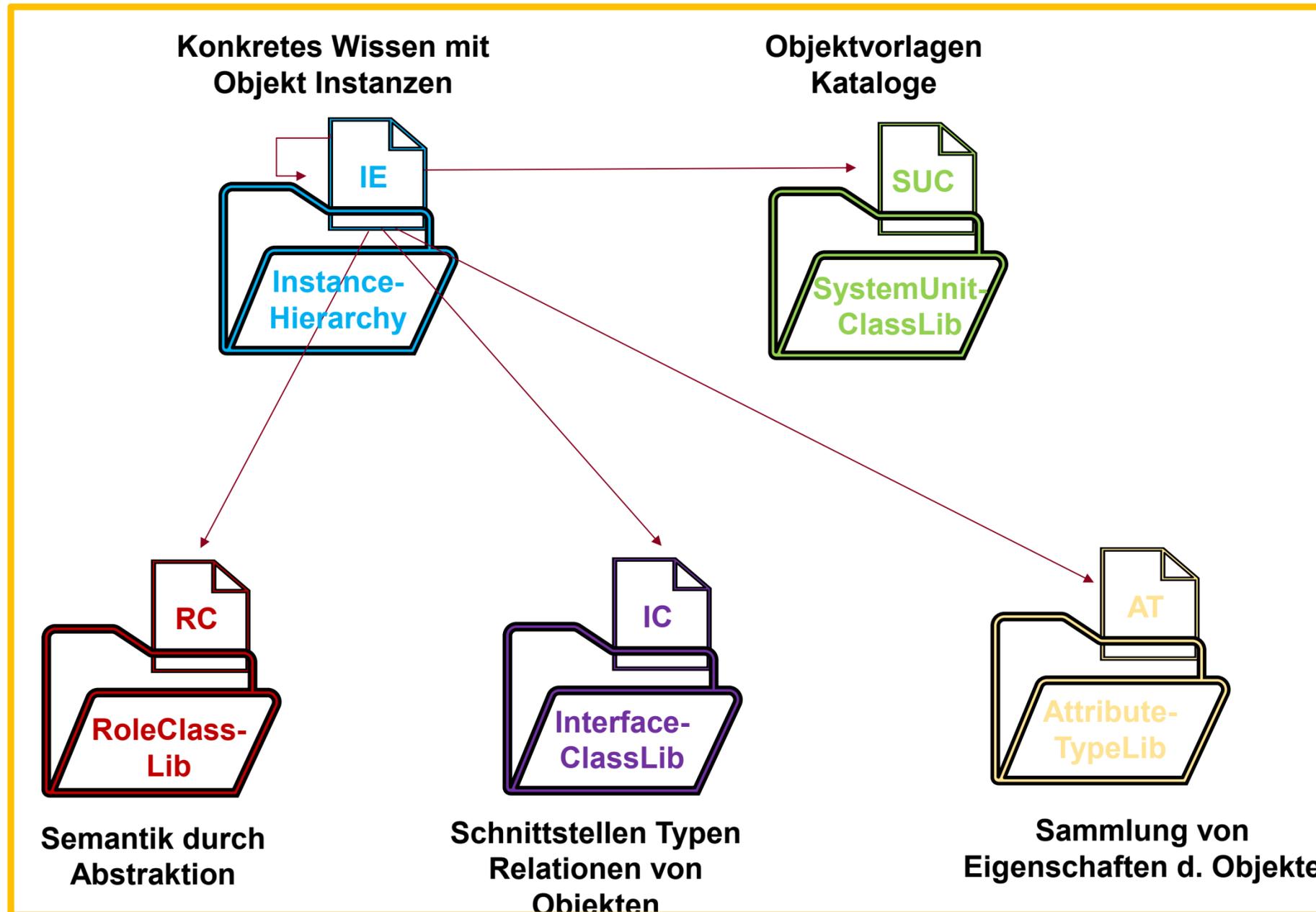
AML-Trainingsaufgabe

Beispiellösung

Lessons Learned & Ausblick

Zusammenfassung

Modellierung in CAEX/AML



Lessons Learned

Das haben Sie heute gelernt ...

- Grundlegende Begriffe und Definitionen im Zusammenhang mit AML/CAEX.
- Die Motivation für den Einsatz von AML und wofür man AML-Modellierung einsetzen kann.
- Wie das Vorgehen bei der objektorientierten Analyse von technischen Systemen ist und wie man technische Systeme gemäß dem CAEX-Schema modellieren kann.
- Sie können nun ...
 - hierarchische Modelle (InstanceHierarchy),
 - SystemUnitKlassen (SUC)
 - Abstraktion von Funktionen in Form von Rollen (RC)
 - Schnittstellen mittels InterfaceClass (IC)
 - sowie Eigenschaften von Objekten mittels Attribute im AML-Editor abbilden.
- Sie wissen wie man Anforderungen und Constraint's für Attribute referenziert und im Datenmodell berücksichtigt.
- Wie die Semantik von Objekten und Schnittstellen über Internal Links und Interface Referenzen abgebildet werden können.

