



 **DIAMOND**

EINFÜHRUNGSSCHULUNG

DIGITALER ZWILLING

Ruhr-Universität Bochum, Lehrstuhl für Produktionssysteme

ÜBERSICHT: ROTER FADEN

Zeitl. Umfang:	20-25 min	20-30 min	20-25 min	90 min	15-20 min	5-10 min
Kapitel:	Einführung	Motivation	Der DZ im AEP	Erstellung eines DZ	Standardisierungs-konzepte	Ausblick Zusammenfassung
Inhalt:	<ul style="list-style-type: none"> Historische Entwicklung und Entstehung des Konzepts vom DZ Abgrenzung der Konzepte VM / VS / DZ 	<ul style="list-style-type: none"> Einsatzgebiete Potenziale für den Einsatz Voraussetzungen für den Einsatz Herausforderungen beim Einsatz 	<ul style="list-style-type: none"> Definition DZ im AEP Anwendungen des DZ über den AEP in VIBN, Konstruktion, Management 	<ul style="list-style-type: none"> Vorgehensweise bei der Erstellung Exkurs: Robotik & Simulation Transformationsprozess VM/DS/DZ 	<ul style="list-style-type: none"> Datenstandardisierung Metadatenstandards Modellierungssprachen Lebenszyklus Management Kommunikationsstandards 	<ul style="list-style-type: none"> Big Picture des Diamond Konzepts Zusammenhang des DZ mit dem CDM Zusammenfassung
Didaktische Methodik:	Direkte Instruktion	Direkte Instruktion + interaktive Umfragen	Brainstorming Session Direkte Instruktion	Direkte Instruktion Aktives lernen via LFF-Demonstrator	Direkte Instruktion Aktives lernen	Evaluation



Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

Standardisierungskonzepte

Zusammenfassung

Ausblick

Einführung

Historische Entwicklung des Konzepts vom digitalen Zwilling

Abgrenzung der Konzepte Digitaler Schatten, Virtuelle Modelle,
Digitaler Zwilling

Motivation

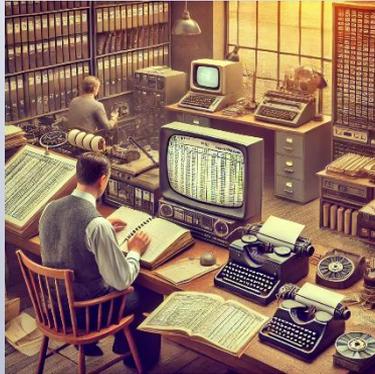
Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

...

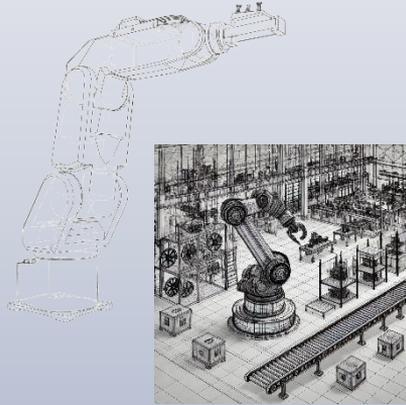
HISTORISCHE ENTWICKLUNG DES KONZEPTS VOM DIGITALEN ZWILLING

Automatisierung und Digitalisierung



1990er

Vernetzung und ganzheitliche Betrachtung von Produktionssystemen



2000er

Simulation Cyberphysische Systeme Product Lifecycle Management



2010er

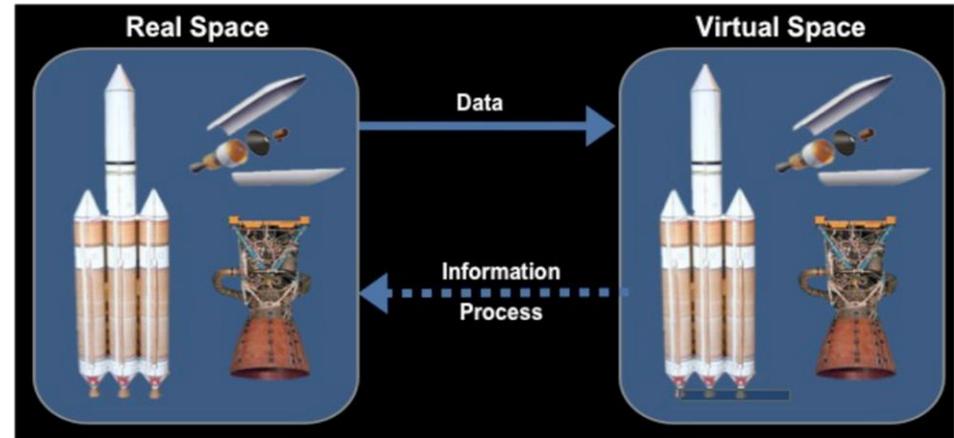
Virtuelle Prozessmodelle Digitaler Schatten Digitaler Zwilling



2020er

HISTORISCHE ENTWICKLUNG DES KONZEPTS VOM DIGITALEN ZWILLING

- **1960er:** Nachbildung von im Weltraum eingesetzter Raumfahrttechnologie durch die NASA
- **1990er:** Die Idee des digitalen Zwillings wird durch Publikationen wie “Mirror Worlds” von David Gelernter der Öffentlichkeit zugänglich gemacht
- **2000er:** Erstmalige Anwendung des Konzepts auf Fertigungssysteme mitsamt Proklamationen erster Softwarekonzepte digitaler Zwillinge
- **2010er:** Einführung eines umfassenden Begriffs zum „digitalen Zwilling“ seitens NASA (J.V)



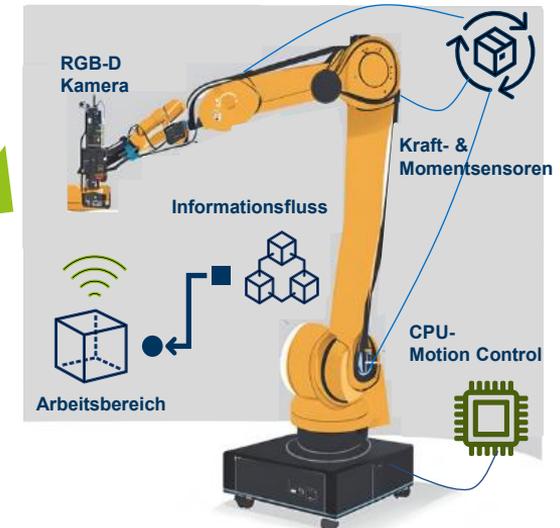
Konzeptskizze des digitalen Zwillings von Grieves und Vickers

HISTORISCHE ENTWICKLUNG DES KONZEPTS VOM DIGITALEN ZWILLING

- **2005:**
 - Erstellung digitaler Modelle für einzelne Produktionsanlagen (Geometrie, Struktur,...)
 - Überwachung der Maschinen- und Anlagenzustände und Optimierung der Wartungsarbeiten
- **2010:**
 - Erfassung von Echtzeitdaten durch Sensoren und Vernetzung im „Internet of Things“ (IoT)
 - Optimierung der Produktionsprozesse und Effizienzsteigerung durch den Einsatz von digitalen Schatten



3D-CAD Modell

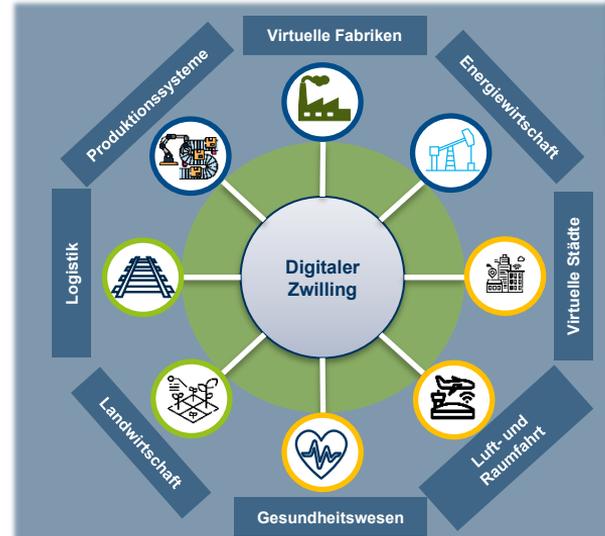


3D-CAD Modell mit integriertem Sensor Informations- und Datenfluss

HISTORISCHE ENTWICKLUNG DES KONZEPTS VOM DIGITALEN ZWILLING

- **2015:**
 - Globalere Dimensionierung von DZ
 - Einsatz von DZ für ganze Fabriken und Produktionsketten.
- **2020:**
 - Erschließung neuer Einsatzgebiete DZ (Gesundheitswesen, Logistik, Energiewirtschaft,...)
 - Leistungssteigerung von DZ durch Fortschritte in den Bereichen KI, maschinellem Lernen und Big Data Analyse.

Cloud basierte Echtzeitsimulation eines Roboterbearbeitungsprozess



Übersicht weiterer Einsatzgebiete von DZ

-
1. In welchem Kontext wurde das Konzept des digitalen Zwillings erstmalig verwendet?
 2. Skizzieren Sie einen Zeitstrahl von 1990 – 2020 und benennen Sie die Schwerpunkte technischer Entwicklungen im Bereich der Digitalisierung.
 3. Nennen Sie 4 mögliche Einsatzgebiete für den Einsatz von digitalen Zwillingen.

Einführung

Historische Entwicklung des Konzepts vom digitalen Zwilling

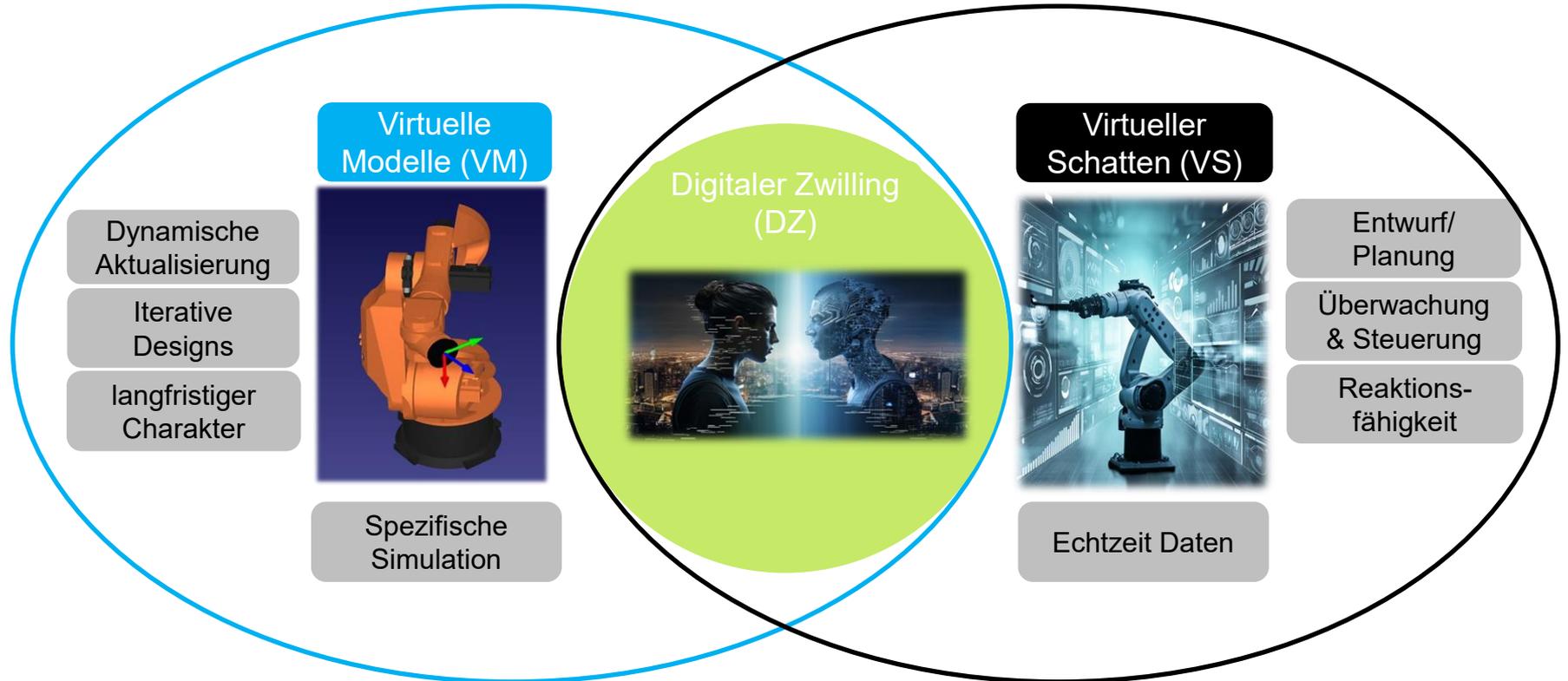
Abgrenzung der Konzepte Digitaler Schatten, Virtuelle Modelle, Digitaler Zwilling

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

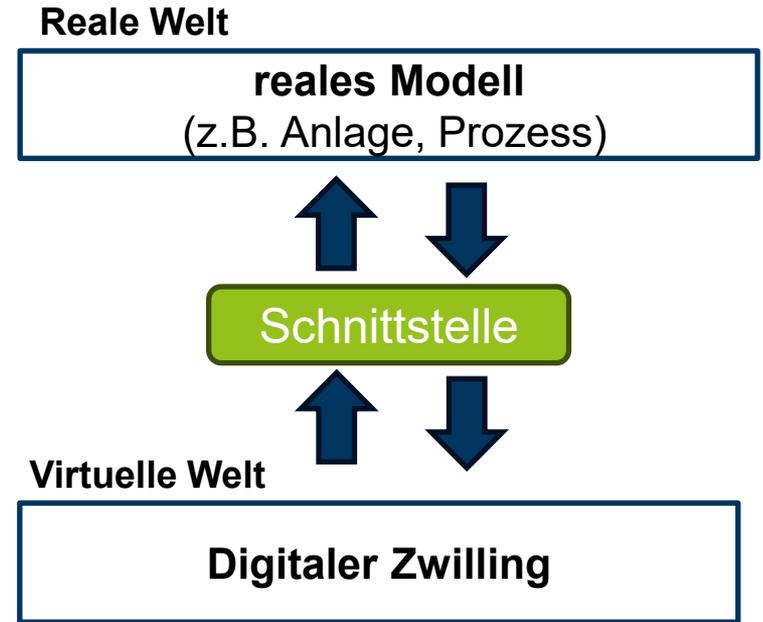
...



KONZEPTABGRENZUNG VIRTUELLES MODELL UND DIGITALER ZWILLING

Virtuelles Modell: digitale Repräsentanzen von Dingen aus der realen Welt.

Digitaler Zwilling: Umfangreiche, Echtzeit-orientierte virtuelle Abbildung mit Schnittstellen zu Simulation und Analysewerkzeugen.



KONZEPTABGRENZUNG DIGITALER SCHATTEN UND DIGITALER ZWILLING



Realer Roboter



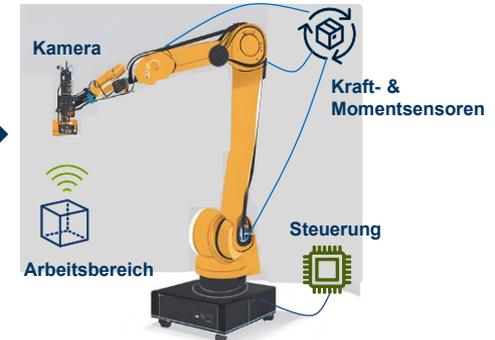
Unidirektional Datenaustausch von
Zustandsinformationen



Digitaler Schatten



Bidirektionaler Datenaustausch von
Steuerungs- und
Optimierungsinformationen
(„Digitaler Impuls“)



Digitaler Zwilling

1. Skizzieren Sie den Zusammenhang zwischen VM, DS und DZ.
2. Was ist der Unterschied zwischen einem virtuellen Modell und einem digitalen Zwilling?
3. Was unterscheidet den virtuellen Schatten (VS) und den digitalen Zwilling (DZ) voneinander?
4. Wie ist der Informationsaustausch zwischen dem virtuellen Modell, der realen Anlage, dem digitalen Schatten und dem digitalen Zwilling definiert?

Einführung

Motivation

Potenziale beim Einsatz vom digitalen Zwilling

Herausforderungen beim Einsatz eines digitalen Zwillings

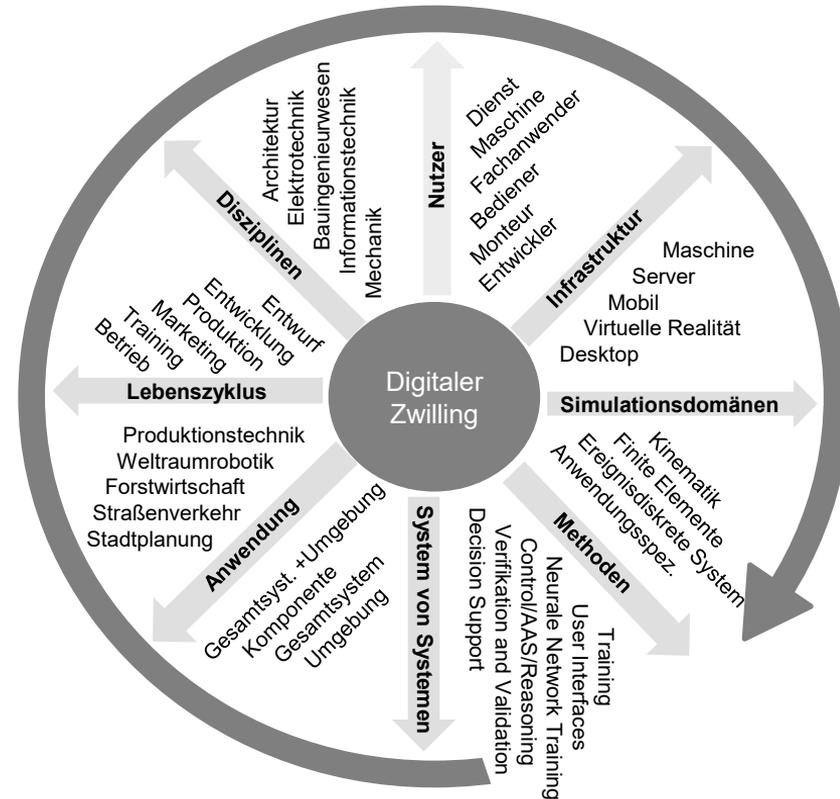
Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

...

EINSATZGEBIETE DES DIGITALEN ZWILLINGS

- **Forschung und Entwicklung:**
→ frühzeitige Ermittlung von Daten und KPI
- **Produktionsplanung:**
→ Simulation basierend auf DS
- **(Prädiktive) Instandhaltung:**
→ Verbesserung der Beurteilungsmöglichkeiten über Zustand und Komponenten eines Assets



Einführung

Motivation

Potenziale beim Einsatz vom digitalen Zwilling

Herausforderungen beim Einsatz eines digitalen Zwillings

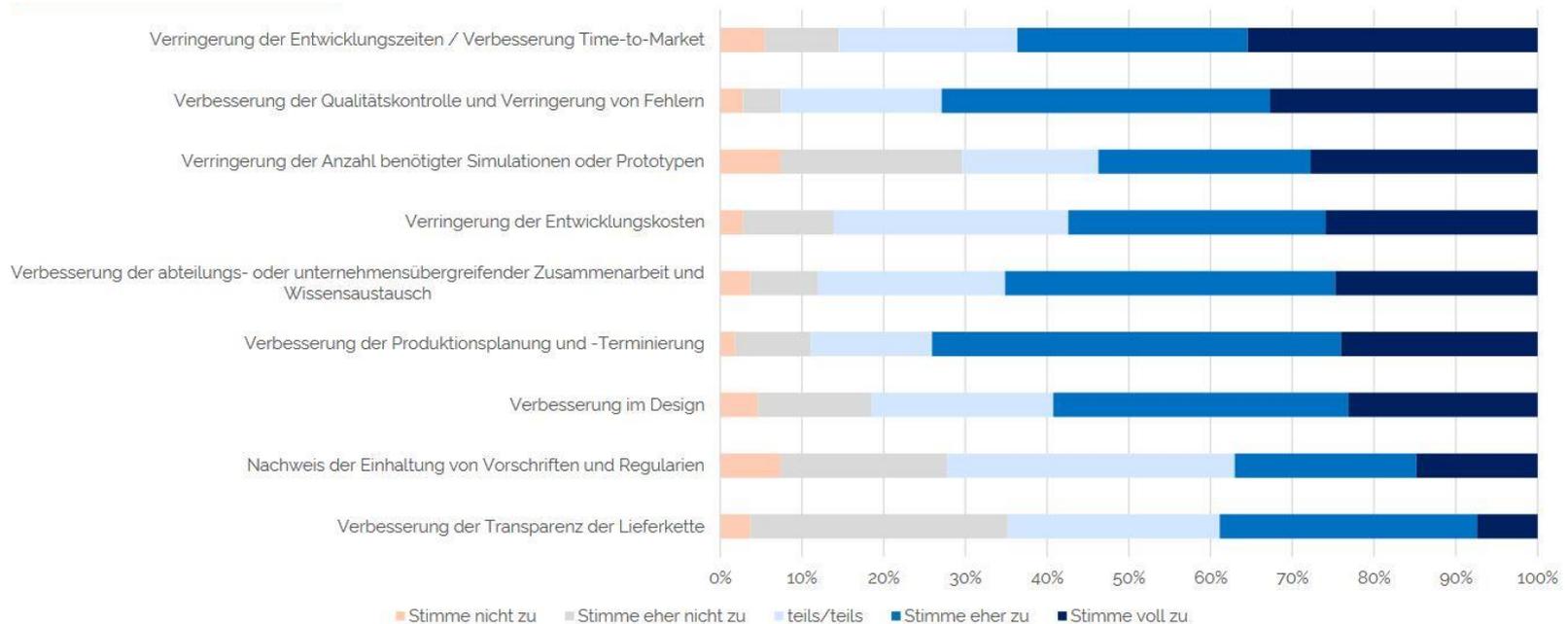
Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

...

(BISHER) UNGENUTZTE POTENZIALE DES DIGITALEN ZWILLINGS

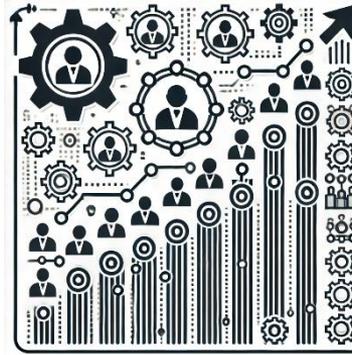
Erhoffter Nutzen durch Einführung von DT



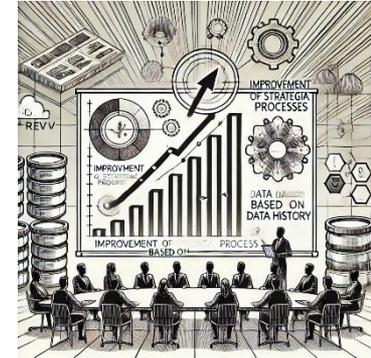
POTENZIALE DES DIGITALEN ZWILLINGS



Unterstützung & Optimierung
von Entscheidungsprozessen



Steigerung der Vernetzung
und Prozessautomation



Verbesserung von strategischen
Prozentscheidungen



Integration von IoT Technologien



Produktivitätssteigerung

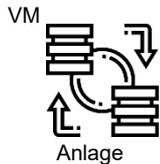


Bereitstellung umfassender
Datenhistorie/ Datenbasis

POTENZIALE DES DIGITALEN ZWILLINGS



Synchronisation virtueller Modelle
mit realem Anlagenzustand



Unterstützung frühzeitiger
Fehlerdetektion



Unterstützung iterativer
Entwicklungsprozesse



1. Nennen Sie 3 mögliche Einsatzbereiche vom digitalen Zwilling.
2. Erläutern Sie welchen Mehrwert der Einsatz eines DZ in der Instandhaltung von Anlagen bringen kann.
3. Beschreiben Sie 3 ungenutzte Potenziale des DZ, im Vergleich zur konventionellen Modellierung und Datenhandhabung.
4. Beschreiben Sie inwiefern der Einsatz von digitalen Zwillingen, Entscheidungsprozesse unterstützen und eine Optimierung der Informationsgrundlage darstellen können.

Einführung

Motivation

Potenziale beim Einsatz vom digitalen Zwilling

Herausforderungen beim Einsatz eines digitalen Zwillings

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

...

VORAUSSETZUNGEN ZUM EINSATZ DIGITALER ZWILLINGE

- **Know-how:** Es erfordert spezifisches Wissen, um digitale Zwillinge effektiv zu nutzen.
- **Rechenleistung:** Die Verarbeitung großer Datenmengen erfordert leistungsstarke Computer.
- **Datensilos:** Informationen sind oft in verschiedenen Systemen verteilt und müssen integriert werden.
- **IT-Architektur:** Die Integration von digitalen Zwillingen in bestehende IT-Systeme kann komplex sein.



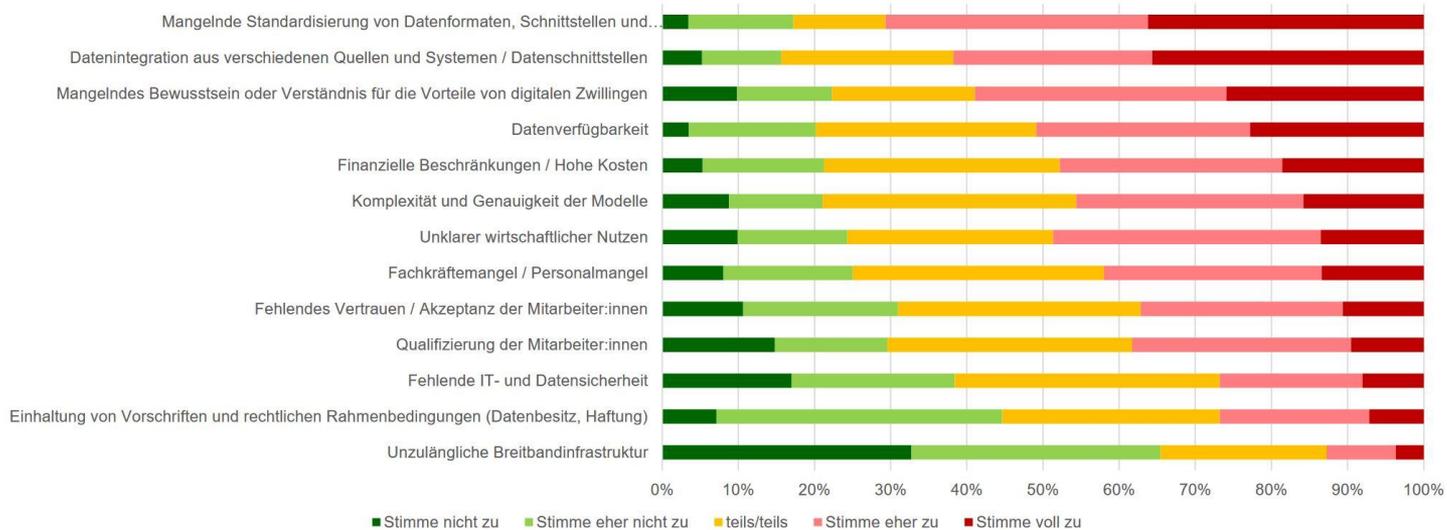
VORAUSSETZUNGEN ZUM EINSATZ DIGITALER ZWILLINGE

Bewertung zur Datenqualität (geordnet)

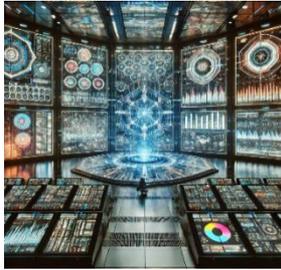


HERAUSFORDERUNGEN DES DIGITALEN ZWILLINGS

Barrieren bei der Einführung von DT



HERAUSFORDERUNGEN DES DIGITALEN ZWILLINGS



Mangelnde **Standardisierung**
(Prozesse, Datenformat)



Mangelndes gemeinsames
Verständnis vom DZ



Interoperabilität stark von
Digitalisierungsstand abhängig



variierende Skalierbarkeit



Stark variierende Komplexität



Komplexe Sicherheitsfragestellungen
(juristisch, etisch, IT)

1. Beschreiben Sie 4 Voraussetzungen zum Einsatz eines DZ in der Anlagenplanung.
2. Worin bestehen die größten Herausforderungen gegenwärtiger Datenstrukturen und Informationskonzepte? Nennen und Erläutern Sie 3 signifikante Probleme mit bestehenden Datenstrukturen.
3. Was sind die geringsten Probleme in bereits etablierter Datenhaltung und wie kann diese Eigenschaft die Einführung eines DZ begünstigen?
4. Welche besonderen Herausforderungen entstehen bei der Einführung von DZ, die unternehmensübergreifenden Daten und Modelle einbinden?

Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Definition digitaler Zwilling im AEP

Anwendungsgebiete von DZ in den AEP

Erstellung eines digitalen Zwillings

...

Gesellschaft für Informatik

„Digitale Zwillinge sind **digitale Repräsentanzen** von Dingen aus der realen Welt. Sie beschreiben sowohl **physische** Objekte als auch **nicht-physische** Dinge wie zum Beispiel Dienste, indem sie alle relevanten Informationen und Dienste mittels einer einheitlichen Schnittstelle zur Verfügung stellen. Für den digitalen Zwilling ist es dabei unerheblich, ob das Gegenstück in der realen Welt schon existiert oder erst existieren wird.“

Roßmann, Jürgen; Schluse, Michael (2020)

Ein Digitaler Zwilling ist die virtuelle digitale Repräsentanz eines materiellen oder immateriellen Assets mit (zumindest teilweise) seinen **Daten und Metadaten**, seinen **Funktionen**, seinen **Kommunikationsmöglichkeiten** sowie der Beschreibung seines **Verhaltens**.

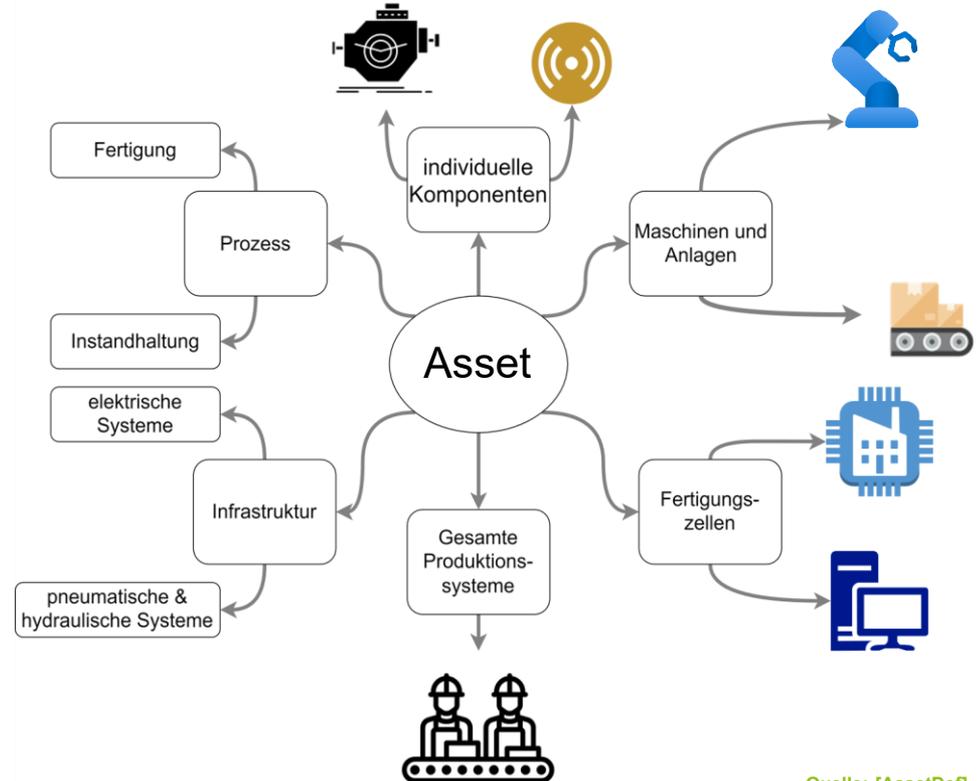
Siegfried Ripperger, VDI Buch Springer 2020

"Ein digitaler Zwilling ist das virtuelle Abbild einer Anlage mit allen ihren Komponenten und Funktionsebenen. Alle relevanten Daten, wie z. B. Abmessungen, Betriebsparameter, Informationen zur Lieferung sind in Form von Dateien darin enthalten und abrufbar. Auch Rechenprogramme zur Auslegung von Verfahren und Anlagenkomponenten sind Teil eines digitalen Zwillings."

ASSET BEGRIFF IM KONTEXT DES DIGITALEN ZWILLING

Asset:

Ein Asset ist eine "Entität", die einen wahrgenommenen oder tatsächlichen Wert für eine Organisation hat und sich im Besitz der Organisation befindet oder von dieser individuell verwaltet wird."



Zusammenfassung:

- Ein digitaler Zwilling fungiert als Schnittstelle zwischen der physischen und virtuellen Welt.
- Er bildet alle beteiligten Vorgänge und Elemente eines realen betrachteten Systems, in der virtuellen Welt ab.
- Im Kontext des Anlagenentstehungsprozesses (AEP) zeichnet sich dieser durch die Integration aller Perspektiven beteiligter Domänen aus und kann als übergeordnetes, vereinigt, virtuelles Modell alle Informationen erfassen.
- Der digitale Zwilling ermöglicht eine kontinuierliche Synchronisation von Daten und Informationen, über alle betrachteten Phasen des Lebenszyklus einer Anlage.

Reale Welt

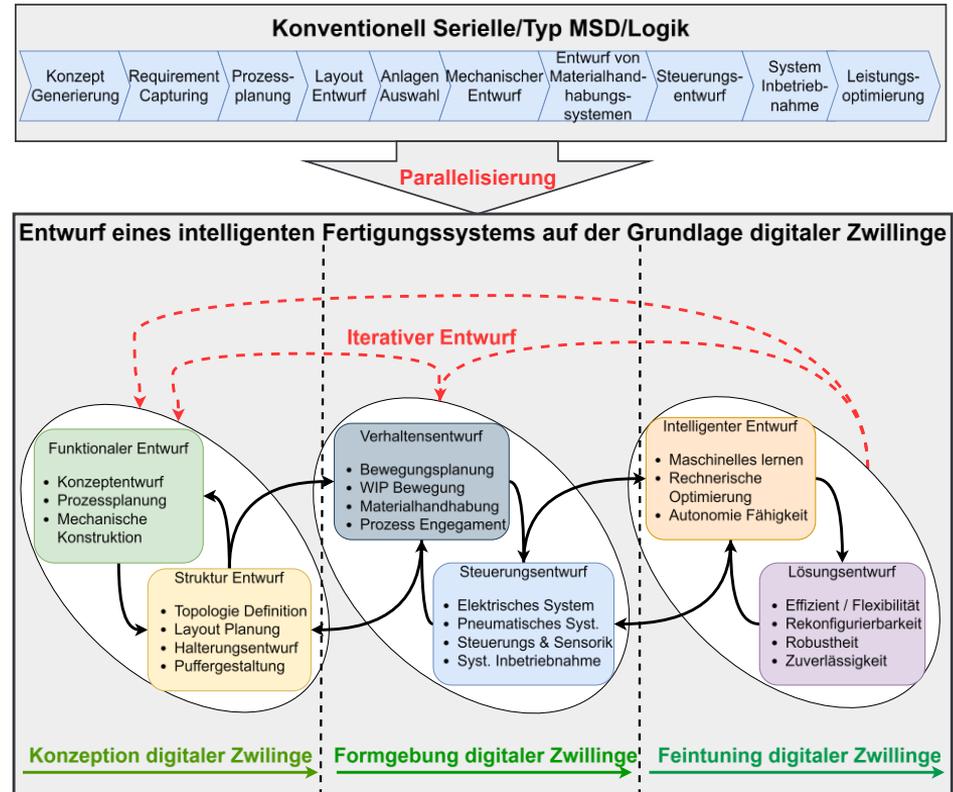


Virtuelle Welt



DIGITALE ZWILLINGE ZUR PARALLELISIERUNG VON ENGINEERING PROZESSEN

- Schlüsseltechnologie der Zukunft
- Werkzeug bei der Modellierung von Anlagen und Prozessen
- Werkzeug zur Leistungsüberwachung einzelner Maschinen und ganzer Produktionssysteme
- Ermöglicht frühzeitige Fehlerdetektion
- Unterstützt Systemoptimierung durch die Minimierung von Ausfallzeit und Leistungsmaximierung



1. Definieren Sie den Begriff eines „Assets“ im Kontext digitaler Zwillinge.
2. Geben Sie 5 Beispiele für Ausprägungen eines Assets an.
3. Skizzieren Sie die Dimensionen eines realen Objekts und des digitalen Zwillings und Erläutern sie den Zusammenhang dieser.
4. Was versteht man unter einem „digitalen Zwilling“ im Anlagenentstehungsprozess?
5. Welche besondere Relevanz kann der Einsatz von DZ für Engineering Prozesse haben?
6. Welche Arten von Entwürfen werden durch digitale Zwillinge unterstützt und welche Ausprägung nehmen diese dabei an?

Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

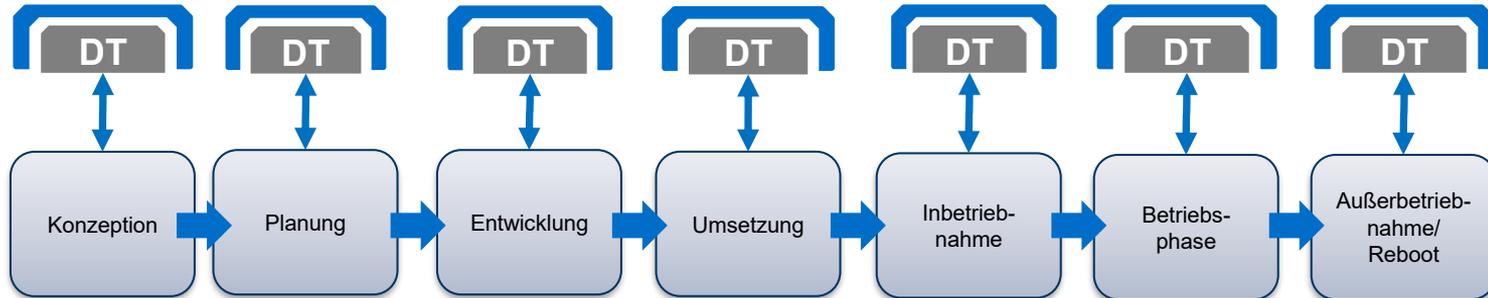
Definition digitaler Zwilling im AEP

Anwendungsgebiete von DZ in den AEP

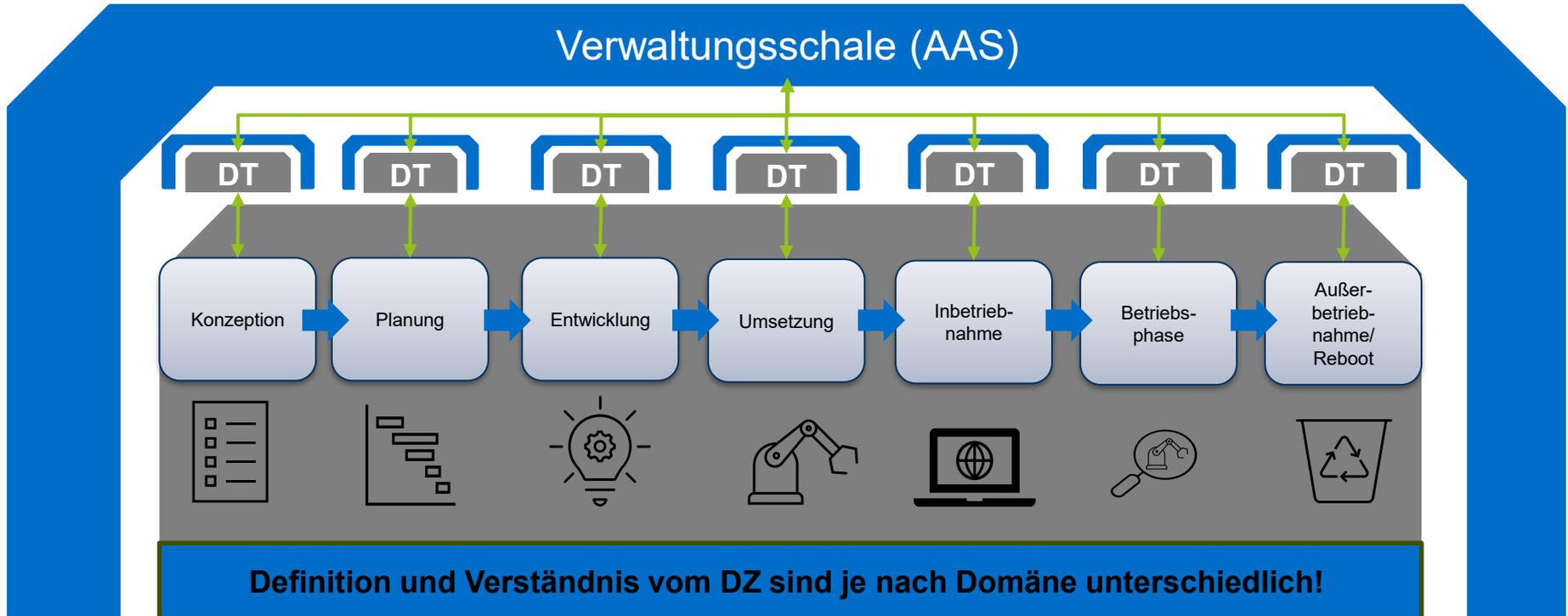
Erstellung eines digitalen Zwillings

...

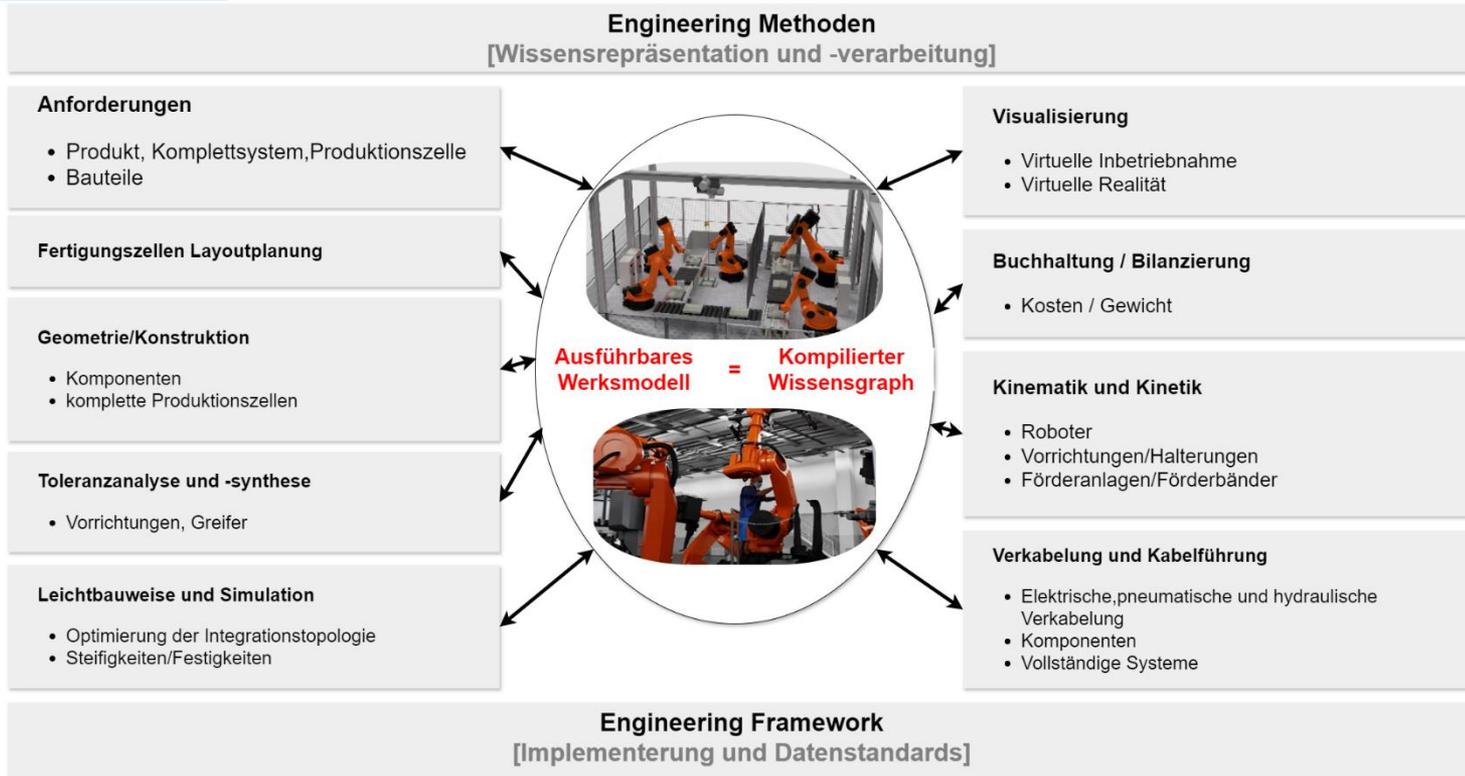
DIGITALE ZWILLINGE IM ANLAGENTESTEHUNGSPROZESS



DIGITALE ZWILLINGE IM ANLAGENTESTEHUNGSPROZESS



DIGITALER ZWILLING IN DER VIRTUELLEN INBETRIEBNAHME (VIBN)



Quelle: [DTBodyNWhite photography extracted fro: <https://la.blogs.nvidia.com/blog/que-es-un-digital-twin/>]

DIGITALER ZWILLING IN DER KONSTRUKTION

- Erweiterung VM um Informationen der DS
- Unterschiedliche Umsetzung eines DZ in verschiedenen CAD-Umgebungen
- Autodesk – ermöglicht AR/VR Integration
- PTC - Creo integriert Sensor/Physiksimulation auf dem virtuellen CAD-Modell
- Siemens – Integration von Werkzeugen über alle Phasen eines Produktzyklus
- Ansys – ganzheitliche Simulationsfunktionen für Prozesse und Lebenszyklus

1. Skizzieren Sie die 7 Phasen des Anlagenentstehungsprozesses und nennen Sie Beispiele für DZ innerhalb der jeweiligen Phase.
2. Erläutern Sie die Bedeutung des DZ für die virtuelle Inbetriebnahme und Beschreiben sie Bestandteile, die ein DZ in der VIBN beinhalten kann.
3. Wie kann der digitale Zwilling in der Konstruktion eingesetzt werden?
4. Erläutern Sie die Unterschiede der Möglichkeiten zur DZ Schnittstellen Integration zwischen PTC Creo, Autodesk und Ansys.
5. Welche Rolle spielt der DZ im Bereich des Managements. Erläutern Sie diese beispielhaft für den Bereich der Fabrikplanung.

Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

Exkurs: industrielle Robotik & Simulation

Der Transformationsprozess zum DZ

...

Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

Exkurs: industrielle Robotik & Simulation

Der Transformationsprozess zum DZ

...

Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

Exkurs: industrielle Robotik & Simulation

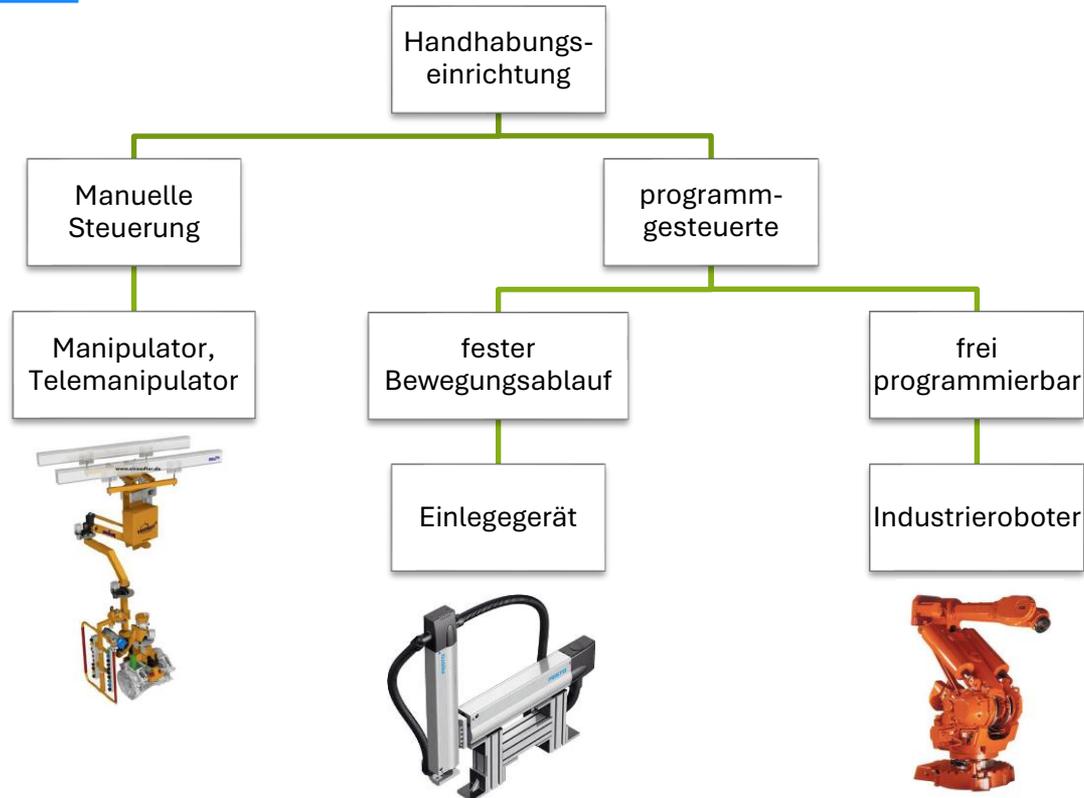
Der Transformationsprozess zum DZ

...

- **Was ist ein Industrieroboter?**
 - Ursprung im tschechischen Wort *robota* (arbeiten)
 - Definition nach der VDI-Richtlinie 2860:

Industrieroboter sind universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabe- oder andere Fertigungsaufgaben ausführen.

EINTEILUNG VON HANDHABUNGSGERÄTEN



- **Hauptklassen der Industrieroboter**

- **Serielle Kinematik** besteht aus einer Aneinanderreihung von Armteilen

- **Parallelkinematik** besteht aus mehreren Schub- oder Drehgelenken, die direkt auf den Endeffektor wirken

- **Hauptteile eines Industrieroboters**

- **Hauptachsen** beeinflussen die Position des Effektors

- **Nebenachsen oder Handachsen** beeinflussen die Orientierung des Effektors

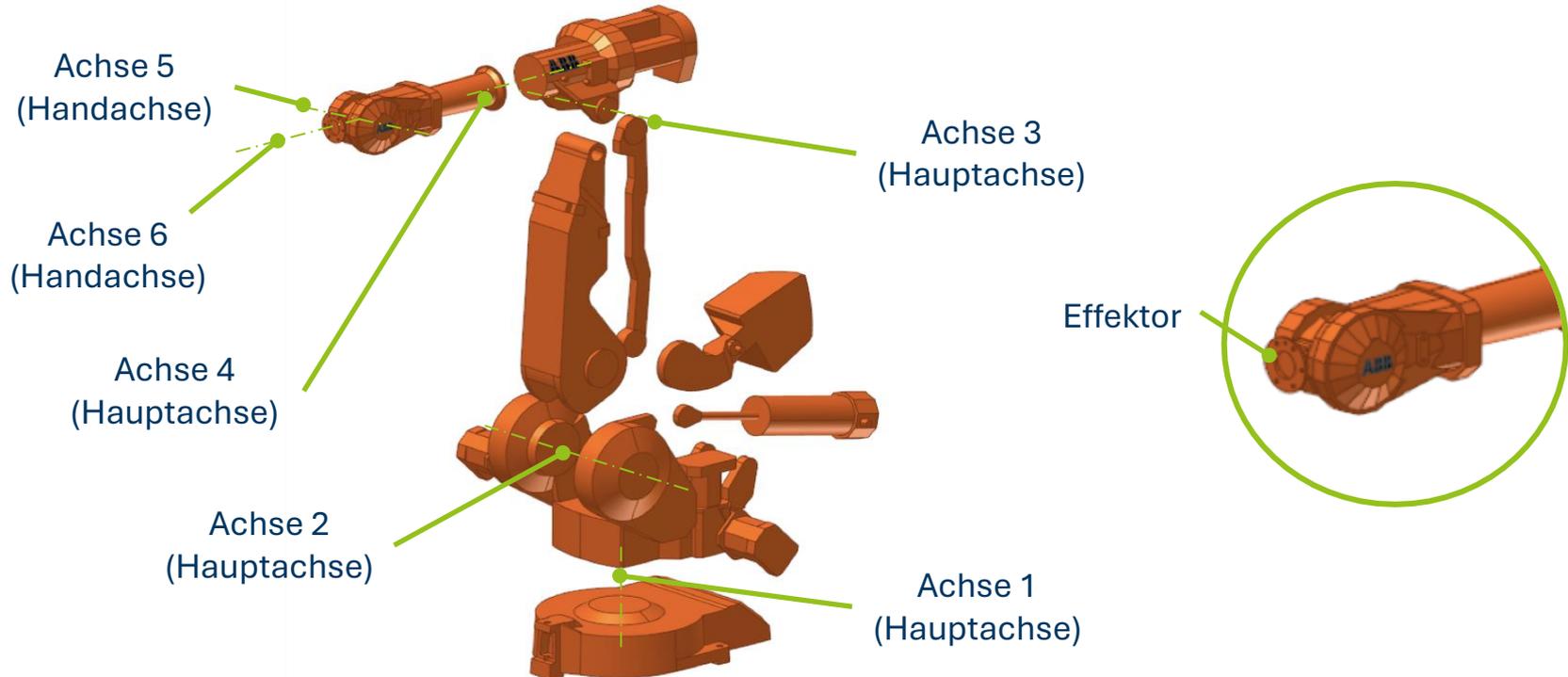
- **Effektor** Teil des Roboterarms, welches mit der Umgebung in Kontakt tritt

- **Hauptaufgabe eines Industrieroboters**

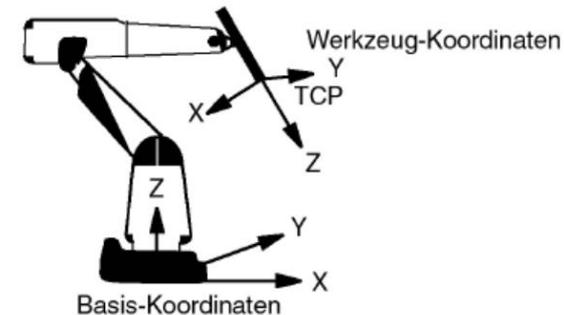
- Den Effektor im Raum zu führen



Quelle: Autodesk, ABB

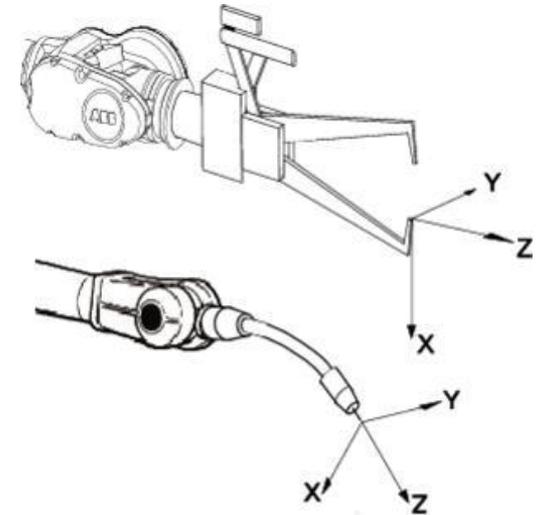
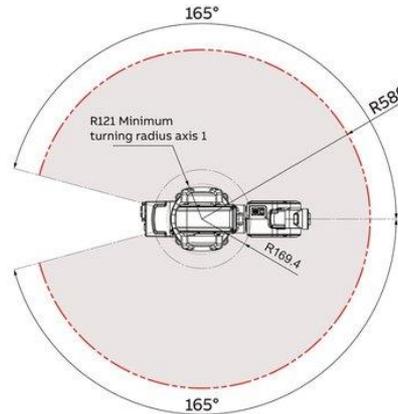
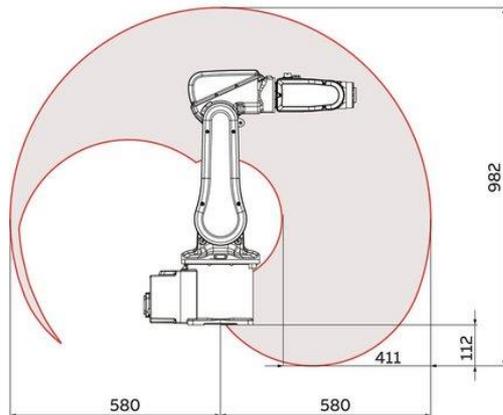


- **Die Anzahl des Getriebefreiheitsgrades F**
 - $F < 6$, für Arbeiten, die sich in einer Ebenen stattfinden (Schwenkarmroboter)
 - $F = 6$, häufigster Robotertyp, Industriestandard (Knickarmroboter)
 - $F > 6$, sogenannte redundante Kinematiken (Leichtbauroboter)
- **Typen von Robotergelenken**
 - **Translatorische Gelenke:** nächstes Armglied bewegt sich auf gerader Bahn vorwärts
 - **Rotatorische Gelenke:** nächstes Armglied dreht sich um eine Rotationsachse
- **Pose des Industrieroboters**
 - Position und Orientierung des Endeffektors
 - Gibt den Bezug des Koordinatensystems im Roboterflansch zum Basiskoordinatensystem im Fuß des Roboters an

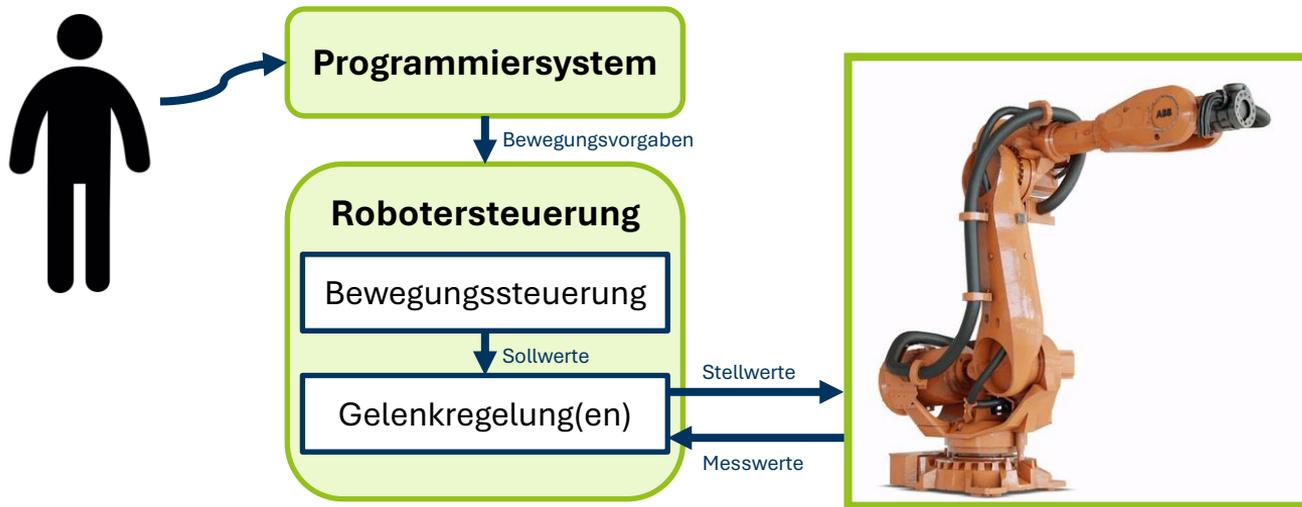


Quelle: IRPA und developercenter.robotstudio.com

- **TCP = Tool-Center-Point (Werkzeugarbeitspunkt)**
 - für eine vorgegebene Anwendung festgelegter Punkt mit Bezug auf das Koordinatensystem der mechanischen Schnittstelle
- **Arbeitsraum**
 - Raumbereich, der mit dem TCP erreicht werden kann



- **Programmiersystem** stellt dem Anwender Funktionen und Befehle bereit, um Bewegungsprogramme aufzustellen
- **Robotersteuerung** umfasst die notwendige Hardware und Systemsoftware, um die Antriebsmotoren anzusteuern



BETRIEBSARTEN EINER ROBOTERSTEUERUNG

Einricht-Betrieb (Teach-In)

- Reduzierte Achsgeschwindigkeiten und -beschleunigungen
- Programmierung und Auswahl von Koordinatensystemen
- Verfahren des Roboters und Speicherung der Positionen
- Programme ergänzen, ändern und verwalten

Test-Betrieb

- Reduzierte Achsgeschwindigkeiten und –beschleunigungen
- Einzelschrittmodus
- Optional: Ausführung des Programms ohne Bewegung des Roboters

Automatik-Betrieb

- Volle Antriebsleistung verfügbar
- Einfaches oder zyklisches Abarbeiten von Programmen

Stand-By

- Antriebe ausgeschaltet
- Je nach Ausführung Kommunikation mit der Steuerung und Konfiguration möglich

BEDEUTUNG UND CHANCEN VON ROBOTERN



■ Vorbehalt gegen Robotik

- hohe Investitionskosten
- Schulungen & Fortbildungen notwendig
- spezielle Anforderungen an Arbeitssicherheit
- verdrängen Mitarbeiter aus Tätigkeitsfeldern
- fördert das Freisetzen von Arbeitsplätzen
- ...

■ Chancen von Robotik

- Handhaben hoher Lasten
- dynamisch einsetzbar
- weitgehend unabhängig von Raumluft, Lärm und Lichtverhältnissen
- unterstützen Menschen z.B. durch kooperierendes Arbeiten
- hohe Wiederholgenauigkeit und Präzision
- schaffen neuer Arbeitsplätze und neuer Tätigkeitsfelder
- ...

AUFGABENFELDER VON INDUSTRIEROBOTERN

Punktschweißen



Handhaben



Lackieren



Biegen



Fräsen



Schmieden



Schneiden



...



Entwicklungen in der industriellen Automatisierung

- Komplexität der Arbeitszellen steigt und Platzbedarf muss reduziert werden
- Anzahl der Produkte und der Produktversionen steigt und Setup-Zeit für neue Produkte muss verringert werden
- Frühzeitiger Beginn der Produktion (SOP) ist von wirtschaftlichem Interesse und die gewünschte Produktionszeit ist 24/7

State-of-the-Art in industriellen Arbeitszellen

- Integration vieler Geräte wie Roboter, Förderbänder und Spannvorrichtungen innerhalb einer Arbeitszelle
- Sehr komplexe geometrische Strukturen für Werkzeuge und Werkstücke
- Viele elektrische und pneumatische Verbindungen sind notwendig

Probleme

- Erreichbarkeiten, Kollisionen,

Aktuelle Produktdesign-Trends

- Die Produkte werden für bestimmte Kunden entwickelt
- Die Batch-Größe geht in Richtung 1

Problem

- Neue Produkte müssen mit Minimaländerungen der Arbeitszelle gefertigt werden
- Das Verhältnis zwischen Rüst- und Produktionszeiten ist zu minimieren

Voraussetzungen

- Programmierung ohne reale Arbeitszelle muss möglich sein
- Optimierte Layout-Planung ist erforderlich
- Fehlersimulation

Bewegungen

- Beispiele: Roboterbewegungen, Objekttransport, Kollisionen
- **Elektro- und Pneumatiksimulation**
 - Beispiele: Ausgänge, Eingänge, Verbindungen
- **Sensorsimulation**
 - Beispiele: optische, kapazitive, induktive und Ultraschall-Sensoren, Bildverarbeitung

Eingeschränkt

- Physikalische Effekte
 - Beispiele: Gravitation und Stöße
- Werkstückänderung
 - Beispiele: Montage, Schneiden, Bohren
- Beleuchtung
 - Beispiele: direkte Beleuchtung, Umgebungslicht

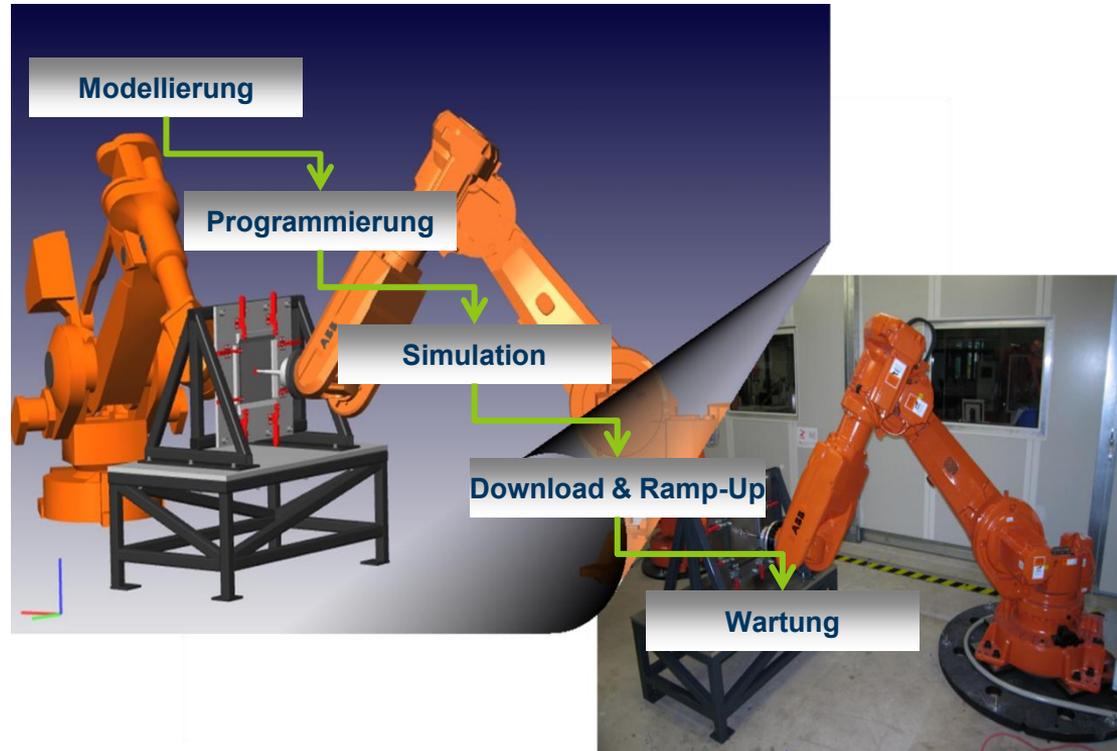
Layout-Planung und Erreichbarkeitsüberprüfung

- Änderung der Gerätpositionen ohne mechanische Eingriffe
 - Keine arbeits-, zeit- und kostenintensiven Änderungen in der Produktionsumgebung
 - Behebung von Layout- und Prozessproblemen vor der Installation der Komponenten und der Inbetriebnahme
- Einfacher Austausch von Robotern und Peripheriegeräten
 - Hersteller und Simulationssysteme bieten Bibliotheken mit ihren Komponenten
 - Schnelle Überprüfung der verschiedenen Robotertypen, bspw. Vergleich der Kosten, Geschwindigkeit, Nutzlast usw. ist möglich
- Die Tests erfolgen ohne die reale Arbeitszelle
 - Risiken von Schäden an Komponenten sind reduziert
 - Risiko für den Menschen ist reduziert!

Früherer Beginn der Produktion

- Roboterprogrammierung ohne realen Roboter
 - Erste Schritte der Programmierung in frühen Phasen des Projekts
 - Programme werden getestet, die Anzahl der Fehler wird verringert
- Concurrent Engineering
 - Mehrere Roboter-Programmierer schreiben und testen Programme
 - Kein einzelner Programmierer blockiert die Arbeitszelle oder die Produktionslinie
- Wiederverwendung von „geleisteter“ Arbeit
 - Änderung der alten Programme
 - Wiederverwendung/Änderung der alten CAD-Daten

VOM PRODUKTDESIGN ZUR REALEN ARBEITSZELLE



Einführung

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

Exkurs: industrielle Robotik & Simulation

Der Transformationsprozess zum DZ

...

- **Eigenschaften:**
 - Reichweite: 559 mm
 - Nutzlast: 0,5 kg
 - Wiederholgenauigkeit: 0,02 mm
 - Gewicht: 38 kg
 - Max. lineare TCP-Geschwindigkeit: 1,5 m/s



Einführung in die Programmierung in Rapid

RAPID – Grundlagen

- Programmiersprache der ABB-Roboter
- Prozedurale Programmierung (vgl. C, Pascal)

Struktur eines RAPID-Programms

- Module
 - Programmmodule
 - Systemmodule
- Globale Deklaration
 - Targets, Werkzeuge, Variablen, etc.
- Prozeduren
 - Menge an Instruktionen
 - Eine Hauptprozedur (main)

Datentypen

- **num** – Numerische Werte

```
VAR num reg1;
```

```
...
```

```
reg1 := 3;
```

- **jointtarget** – Achsenpositionsdaten

```
VAR jointtarget pHome:=[[0,0,0,0,0,0],[0,0,9E9,9E9,9E9,9E9]];  
                        [  robax  ] [      extax      ]
```

- **robtarger** – Positionsdaten

```
VAR robtarget p15:=[[60,50,25.3],[1,0,0,0],[1,1,0,0],[11,12.3,0,0,9E9,9E9]];  
                  [  trans  ] [  rot  ] [robconf] [      extax      ]
```

Datentypen

- **speeddata** – Geschwindigkeitsdaten

```
VAR speeddata vmedium := [ 1000, 30, 200, 15 ];  
                        [ v_tcp, v_ori, v_leax, v_reax ];
```

Vordefinierte Typen:

v100	100 mm/s	500°/s	5000 mm/s	1000°/s
v1000	1000 mm/s	500°/s	5000 mm/s	1000°/s
vmax	*)	500°/s	5000 mm/s	1000°/s

- **zonedata** – Zonendaten

```
VAR zonedata path := [ FALSE, 25, 40, 40, 10, 35, 5 ];  
                    [ finep, pzone_tcp, pzone_ori, Externe Achsen ]
```

Vordefinierte Typen:

z0	0,3 mm	0,3 mm	0,3 mm	0.03°	0,3 mm	0.03°
z10	10,0 mm	15,0 mm	15,0 mm	1.5°	15,0 mm	1.5°
z100	100,0 mm	150,0 mm	150,0 mm	15.0°	150,0 mm	15.0°

Datentypen

- **tooldata** – Werkzeugdaten

```
PERS tooldata tool0 := [ TRUE, [ [0, 0, 0], [1, 0, 0, 0] ],  
                        [0.001, [0, 0, 0.001], [1, 0, 0, 0], 0, 0, 0] ];
```

- **wobjdata** – Werkobjektdaten

```
PERS wobjdata wobj0 := [ FALSE, TRUE, "", [ [0, 0, 0], [1, 0, 0, 0] ],  
                        [ [0, 0, 0], [1, 0, 0, 0] ] ];
```

Mögliche Bewegungsinstruktionen

- **MoveAbsJ** – Bewegen des Roboters an eine absolute Achsenposition
- **MoveJ** – Achsenbewegung des Roboters
- **MoveL** – Lineare Bewegung des Roboters
- **MoveC** – Kreisförmige Bewegung des Roboters

Notwendige Parameter

- Zielpose, Geschwindigkeit, Überschleifradius, Werkzeug

Bewegungsart



MoveL

Geschwindigkeit



v100,

Werkzeug



AW_Gun

Target_10, z100, WObj := w0



Zielpose



Überschleifradius



Koord.-System

Bewegungsinstruktion MoveJ

- Point-to-Point Bewegung (vereinfachte Syntax)

```
MoveJ [\Conc] ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool  
[\Wobj]
```

- Parameter:

- ToPoint ← rotarget
- Speed ← speeddata
- Zone ← Zonedata
- Tool ← Tooldata
- \Wobj (optional) ← Workobject
- Optional: \Conc, \ID, \V, \T, \Inpos

- Beispiel: Deklaration

```
MoveJ p1, vmax, z30, tool2;  
MoveJ *, vmax \T:=5, fine, grip3;
```

Bewegungsinstruktion MoveL

- Kontinuierlicher Pfad (Linear)

```
MoveL [\Conc] ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool  
[\WObj] [\Corr]
```

- Parameter:

- Analog zur MoveJ-Instruktion
- \Corr – Korrekturdaten

- Beispiel: Deklaration

```
MoveL p1, v1000, z30, tool2;  
MoveL start, v2000, z40, grip3\WObj:=fixture;
```

- **wObj:** Koordinatensystem auf das sich die Roboterposition bezieht (Datentyp: wobjdata)

Bewegungsinstruktion MoveC

- Kontinuierlicher Pfad (Zirkular)

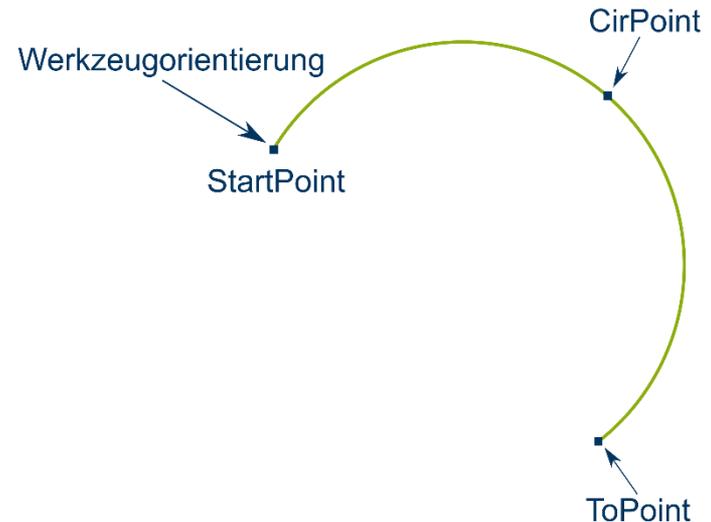
```
MoveJ [\Conc] CirPoint ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos]  
Tool [\WObj] [\Corr]
```

- Parameter:

- Analog zu MoveL-Instruktion
- CirPoint – Bogenpunkt des Roboters

- Beispiel: Deklaration

```
MoveC p1, p2, v500, z30, tool2;
```



Sonstige Instruktionen (1)

- Zuweisungsoperator

`:=`

- For-Schleife

```
FOR i FROM 1 TO 10 STEP 1 DO
```

```
    x := x + i;
```

```
ENDFOR
```

- **OrientZYX** – Erzeugen einer Orientierung aus Eulerschen Winkeln

```
object.rot := OrientZYX(anglez, angley, anglx)
```

- **offs** – Verschieben einer Roboterposition

```
Offs (Point, Xoffset, Yoffset, ZOffset)
```

Sonstige Instruktionen (2)

- **SetDO** – Ändern des Wertes eines digitalen Ausgangssignals
`SetDO R_Attach_Ring_1, 1;` (rechten Greifer schließen und Ring verbinden)
`SetDO R_Attach_Ring_1, 0;` (rechten Greifer öffnen und Ring lösen)

- **WaitTime** – Warten bis eine vorgegebene Zeit verstrichen ist
`WaitTime 0.5;`

- **Pose erstellen**
 - Programm Daten → Ansicht → Alle Datentypen → robtarget
 - Erstellen Sie eine Pose wo der Roboter das Logo des LPS abfährt
 - Übernehmen Sie das Programm in eine Simulation und spielen Sie die Simulation zur Kontrolle Ihrer Programmierung ab
 - Spielen Sie die Programmierung auf den YuMi und führen Sie die Pose aus
- **Routine erstellen**
 - Module → user → Routine → Datei → Neue Routine
 - *moveHome*
 - *MoveJ pHome, v100, fine, tool0*
 - Fahren Sie den Roboter in eine beliebige Position und führen dann die Routine *moveHome* aus

- Erstellen Sie für Ihre Pose im MainModule die Koordinatenpunkte, die der Roboter einzelnen anfahren soll
- Benennen Sie Ihre Pose um
- Programmieren Sie die einzelnen Bewegungen der Pose, indem Sie die Reihenfolge der anzufahrenden Punkte festlegen

```
MODULE MainModule
  CONST robtarget p10:=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]];
  CONST robtarget p20:=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]];
  CONST robtarget p30:=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]];
  CONST robtarget p40:=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]];
  CONST robtarget p50:=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]];

  PROC main()
    Name_Pose;
  ENDPROC

  PROC Name_Pose()
    MoveJ p10, v100, fine, tool0;
    MoveL p20, v100, fine, tool0;
    MoveL p30, v100, fine, tool0;
    MoveL p40, v100, fine, tool0;
    MoveL p20, v100, fine, tool0;
    MoveL p50, v100, fine, tool0;
    MoveL p30, v100, fine, tool0;
    MoveL p10, v100, fine, tool0;
    MoveL p40, v100, fine, tool0;
  ENDPROC
endmodule
```

```
1  MODULE MainModule
2  ▢  CONST robtarget p10:=[[357.67,-213.06,106.79],[0.436169,-0.544164,0.56837,-0.436575],[1,-3,0,4],[-129.039,9E+09,9E+09,9E+09,9E+09]];
3  ▢  CONST robtarget p20:=[[357.67,-213.06,242.25],[0.436164,-0.544165,0.568371,-0.436577],[1,-2,0,4],[-129.038,9E+09,9E+09,9E+09,9E+09]];
4  ▢  CONST robtarget p30:=[[357.67,-29.13,242.25],[0.43617,-0.544162,0.568369,-0.436578],[1,-3,0,4],[-129.038,9E+09,9E+09,9E+09,9E+09]];
5  ▢  CONST robtarget p40:=[[357.68,-29.13,106.86],[0.436162,-0.544166,0.568369,-0.436581],[1,-3,0,4],[-129.039,9E+09,9E+09,9E+09,9E+09]];
6  ▢  CONST robtarget p50:=[[357.67,-113.00,334.45],[0.436166,-0.544161,0.568365,-0.436588],[1,-2,0,4],[-129.039,9E+09,9E+09,9E+09,9E+09]];
7
8
9  ▢  PROC main()
10     HausVomNikolaus;
11  ▢  ENDPROC
12
13  ▢  PROC HausVomNikolaus()
14     MoveJ p10, v100, fine, tool0;
15     MoveL p20, v100, fine, tool0;
16     MoveL p30, v100, fine, tool0;
17     MoveL p40, v100, fine, tool0;
18     MoveL p20, v100, fine, tool0;
19     MoveL p50, v100, fine, tool0;
20     MoveL p30, v100, fine, tool0;
21     MoveL p10, v100, fine, tool0;
22     MoveL p40, v100, fine, tool0;
23  ▢  ENDPROC
24  ▢  endmodule
```

Koordinatenpunkte



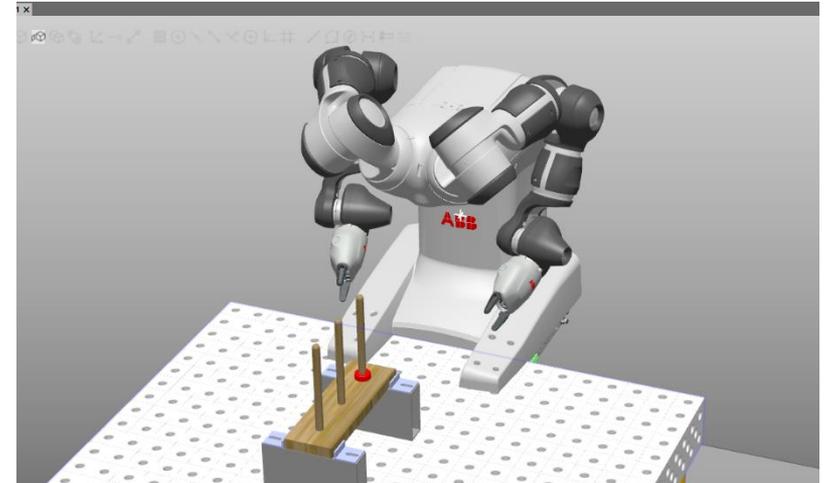
Bewegungsreihenfolge



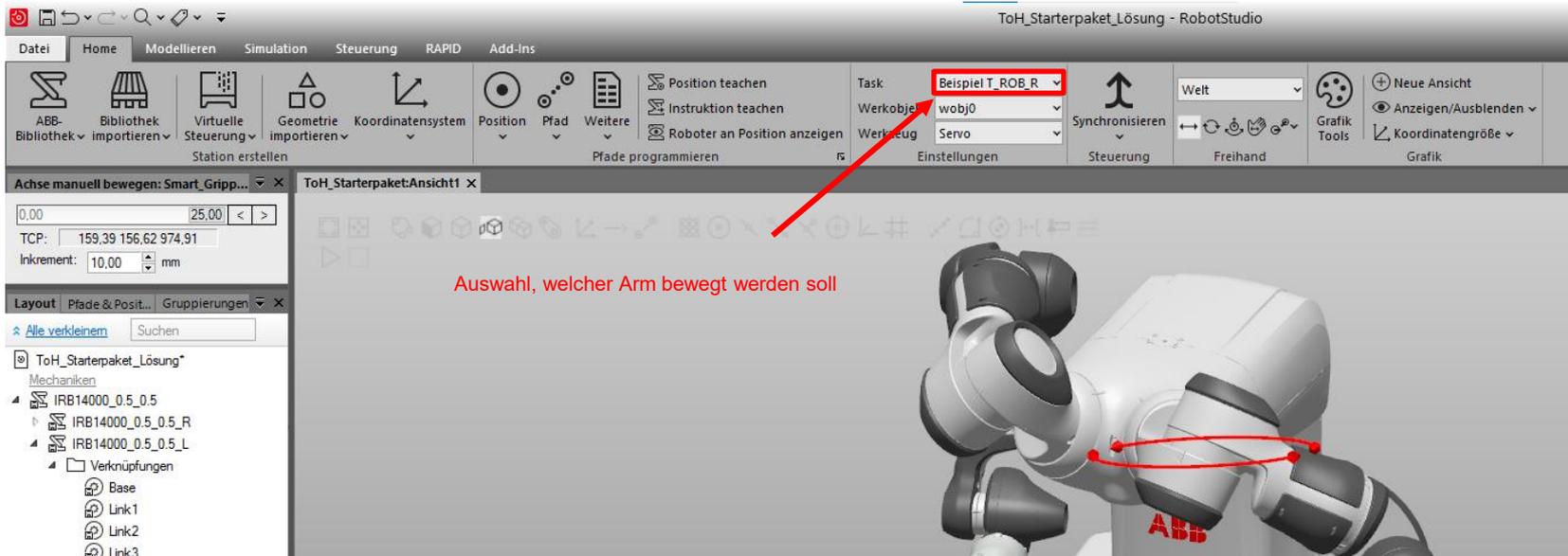
Einführung in RobotStudio

Navigation in der 3D-Ansicht

- Zoom:
 - Mausrad *oder*
 - [Strg] + rechte Maustaste
- Rotieren:
 - Mausrad + rechte Maustaste *oder*
 - [Strg] + [Umschalt] + linke Maustaste
- Verschieben:
 - [Strg] + linke Maustaste



Bewegung der Mechanik



The screenshot shows the RobotStudio interface with the following elements:

- Task dropdown:** A red box highlights "Beispiel T_ROB_R" in the Task dropdown menu.
- Red arrow:** A red arrow points from the "Beispiel T_ROB_R" dropdown to the text "Auswahl, welcher Arm bewegt werden soll".
- 3D Model:** A 3D model of a robot arm is shown with red arrows indicating movement directions.
- Left Panel:** Contains a coordinate system, TCP values (159.39, 156.62, 974.91), and a tree view showing the robot's structure (ToH_Starterpaket_Lösung, IRB14000_0.5_0.5, etc.).
- Top Menu:** Includes options like "Datei", "Home", "Modellieren", "Simulation", "Steuerung", "RAPID", and "Add-Ins".

Auswahl, welcher Arm bewegt werden soll

Achswise bewegen:
Achse auswählen und
mit dem Cursor bewegen

Linear bewegen:
Flansch auswählen und
über Pfeile linear bewegen

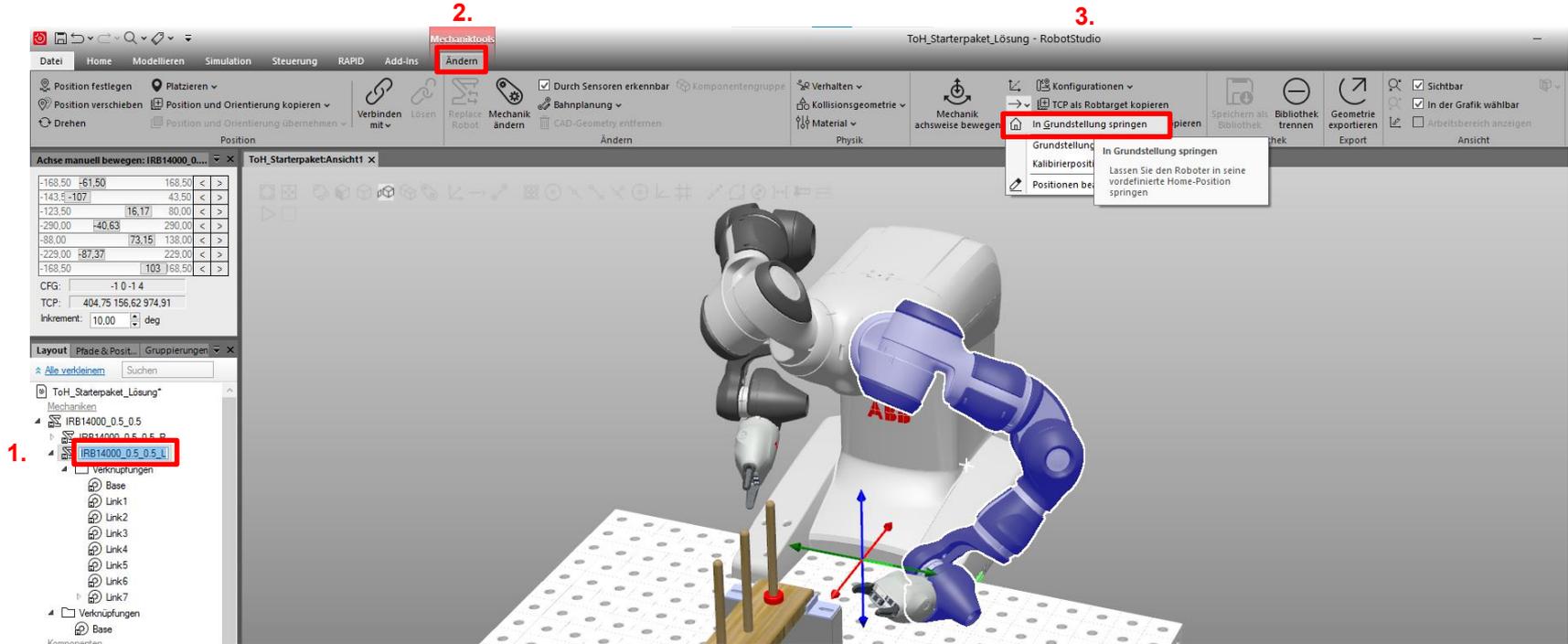
Umorientieren:
Flansch auswählen und
über Pfeile umorientieren

Bewegung der Mechanik

1. IRB14000_0.5_0.5_1

2. Mechaniktools > **Ändern**

3. In Grundstellung springen



Achse manuell bewegen: IRB14000_0...

-168,50	-61,50	168,50	<	>	
-143,5	-107	43,50	<	>	
-123,50		16,17	80,00	<	>
-290,00	-40,63	290,00	<	>	
-88,00		73,15	138,00	<	>
-229,00	-87,37	229,00	<	>	
-168,50		103	68,50	<	>

CPG: -1 0 -1 4
TCP: 404,75 156,62 974,91
Inkrement: 10,00 deg

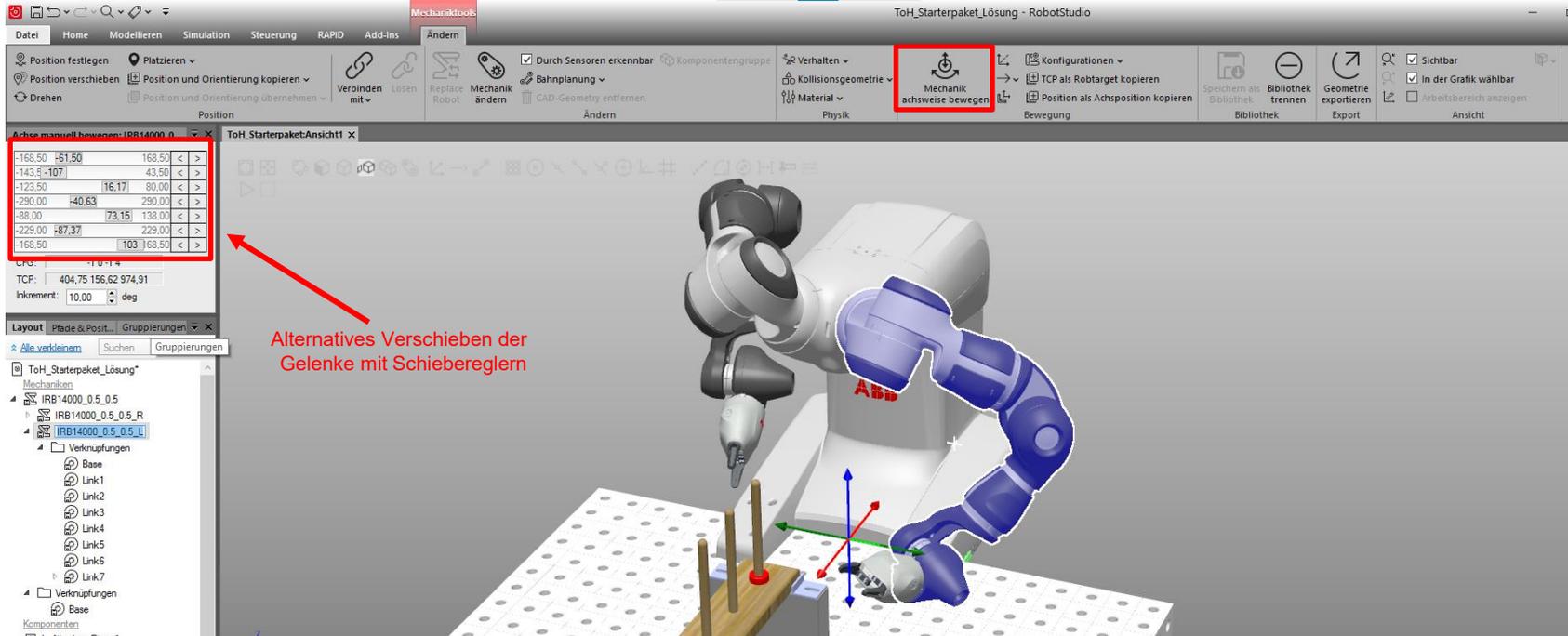
Layout | Platte & Posit... | Gruppierungen

- Alle verkleinern
- Suchen
- ToH_Starterpaket_Lösung*
- Mechaniken
 - IRB14000_0.5_0.5
 - IRB14000_0.5_0.5_P
 - IRB14000_0.5_0.5_1
 - Verknüpfungen
 - Base
 - Link1
 - Link2
 - Link3
 - Link4
 - Link5
 - Link6
 - Link7
 - Verknüpfungen
 - Base

Komponentenbaum

In Grundstellung springen
Lassen Sie den Roboter in seine vordefinierte Home-Position springen

Bewegung der Mechanik

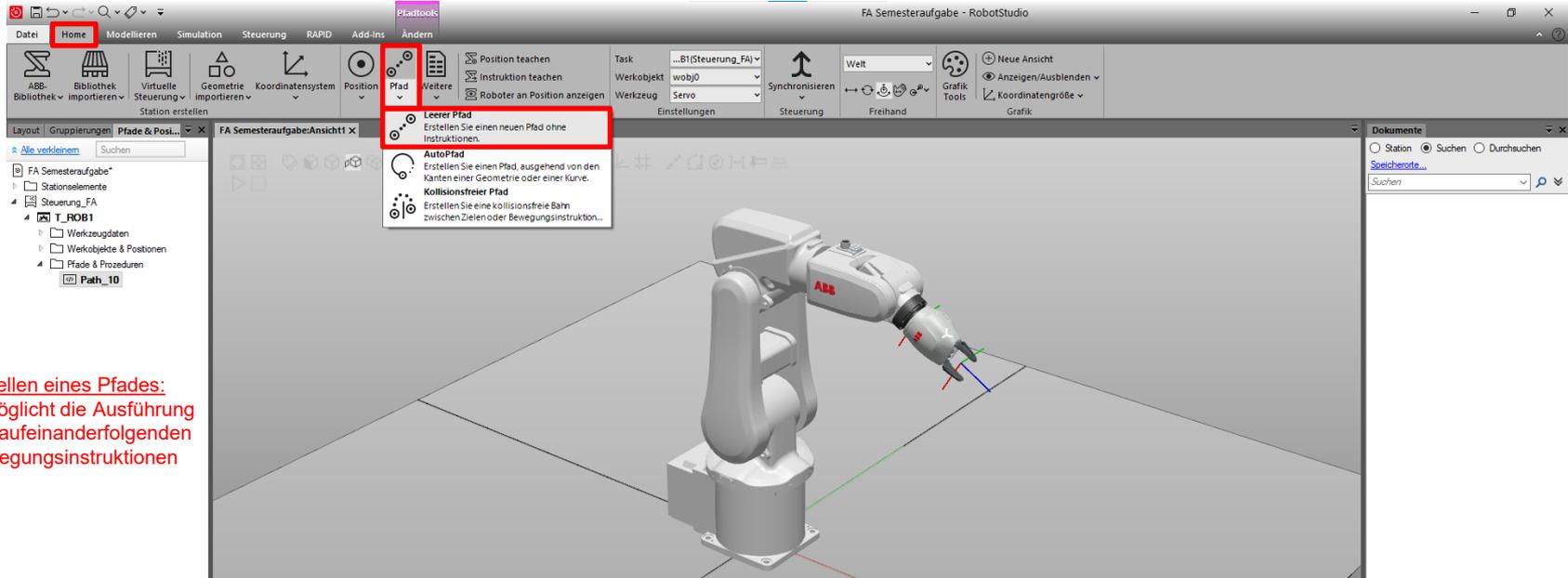


The screenshot shows the RobotStudio interface with the 'MechanikTools' tab active. A red box highlights the 'Mechanik achsweise bewegen' button in the toolbar. Another red box highlights a table of joint coordinates in the 'Achse manuell bewegen: IRB14000_0' window. A red arrow points from the text 'Alternatives Verschieben der Gelenke mit Schiebereglern' to this table.

X	Y	Z	W	U	V
-168.50	-61.50	168.50	<	>	
-143.4	-107	43.50	<	>	
-123.50	16.17	80.00	<	>	
-290.00	-40.63	290.00	<	>	
-88.00	73.15	138.00	<	>	
-229.00	-87.37	229.00	<	>	
-168.50	103	168.50	<	>	

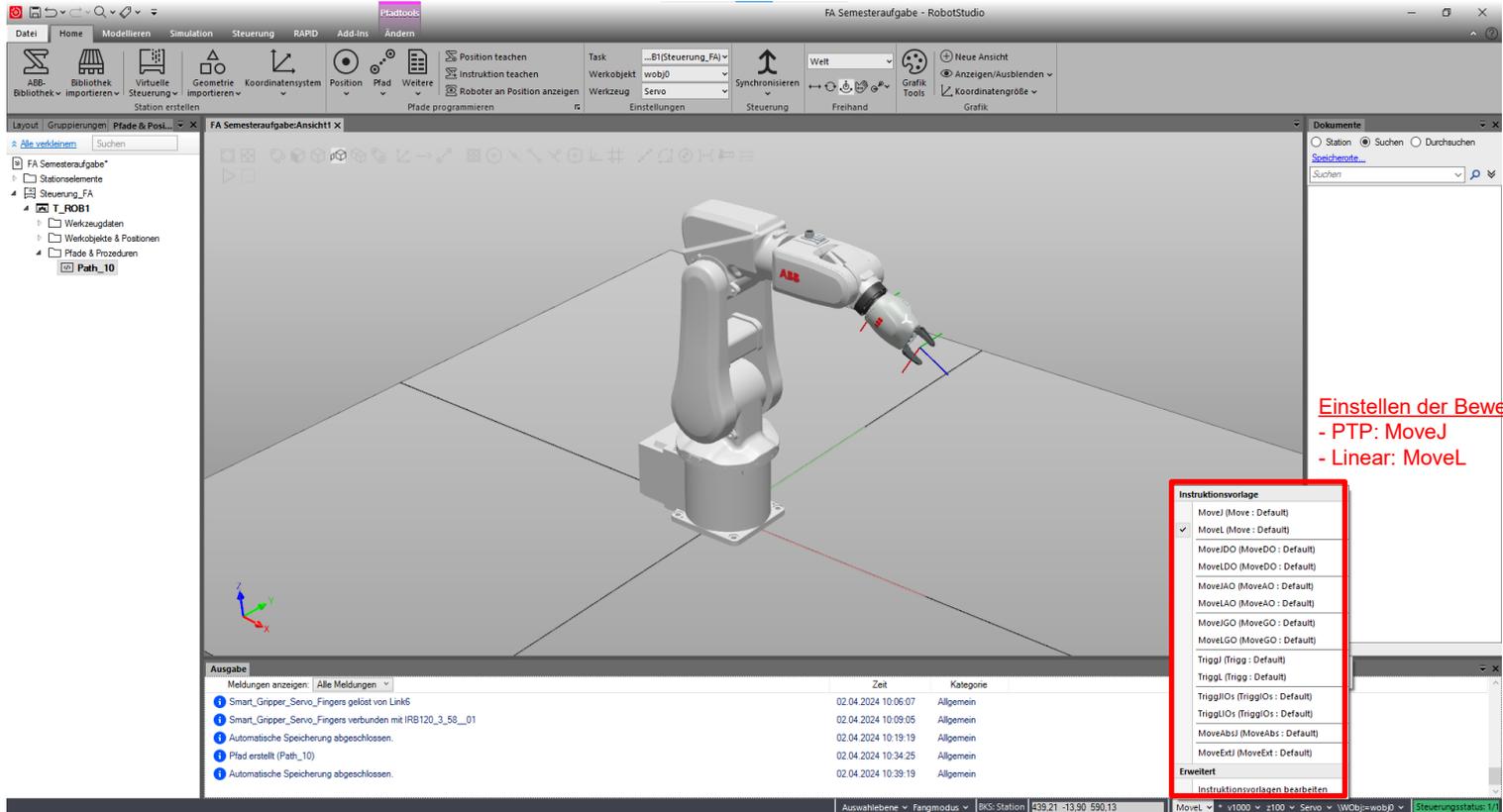
Alternatives Verschieben der Gelenke mit Schiebereglern

Erstellen eines Pfades



Erstellen eines Pfades:
Ermöglicht die Ausführung
von aufeinanderfolgenden
Bewegungsinstruktionen

Definition der Bewegungsart



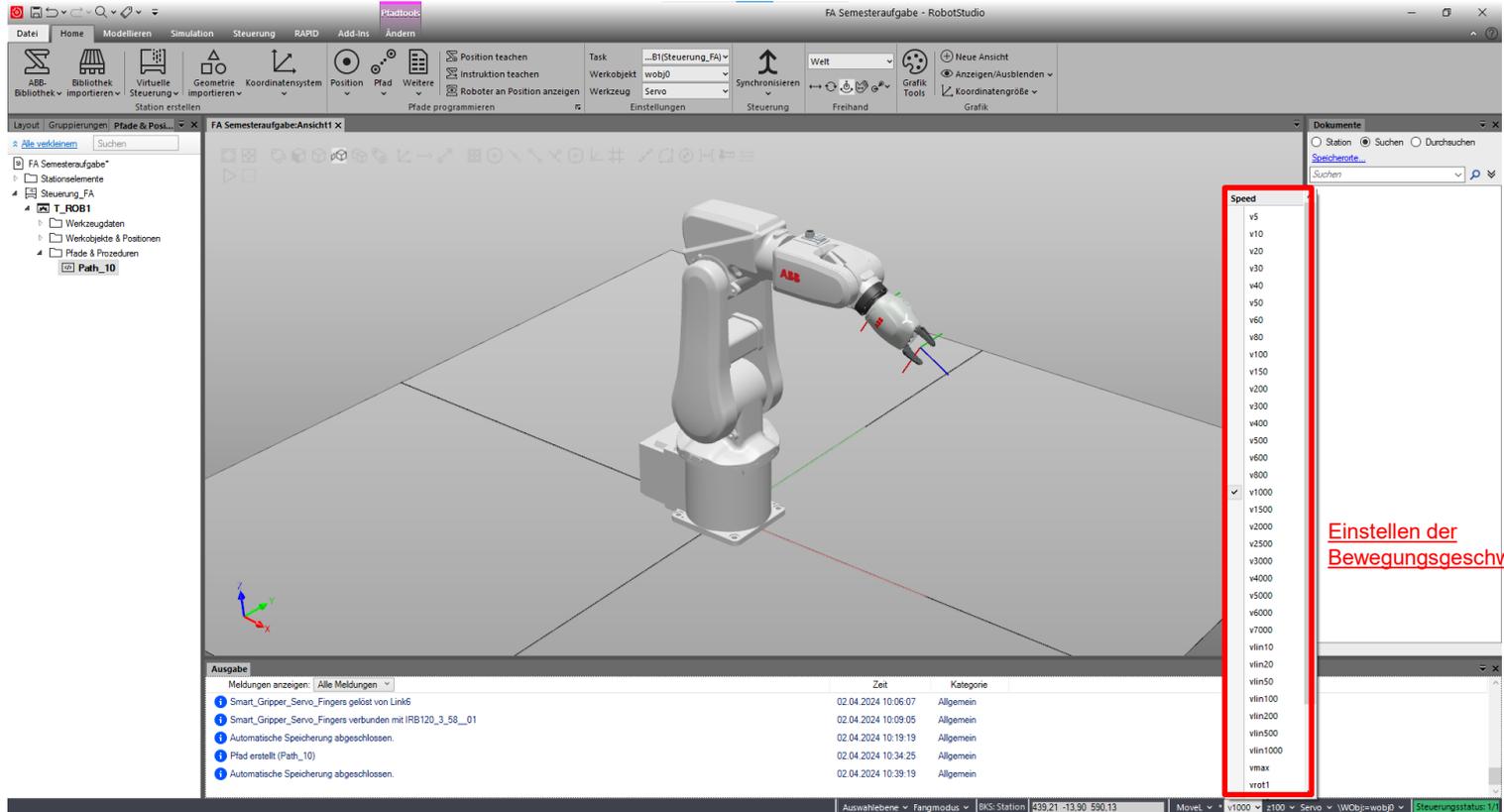
The screenshot displays the RobotStudio software interface. The main window shows a 3D model of a robotic arm in a virtual environment. The interface includes a top menu bar with options like 'Datei', 'Home', 'Modifizieren', 'Simulation', 'Steuerung', 'RAPID', 'Add-Ins', and 'Ändern'. Below the menu is a toolbar with various icons for tasks, work objects, and tools. The left sidebar shows a project tree with folders like 'FA Semesteraufgabe' and 'Steuerung_FA'. The bottom status bar displays system information and the current motion type, which is 'MoveL'. A red box highlights a list of motion types in the 'Instruktionsvorlage' panel.

Einstellen der Bewegungsart:

- PTP: MoveJ
- Linear: MoveL

Instruktionsvorlage		
<input type="checkbox"/>	MoveJ (Move : Default)	
<input checked="" type="checkbox"/>	MoveL (Move : Default)	
<input type="checkbox"/>	MoveJDO (MoveDO : Default)	
<input type="checkbox"/>	MoveLDO (MoveDO : Default)	
<input type="checkbox"/>	MoveJAO (MoveAO : Default)	
<input type="checkbox"/>	MoveLAO (MoveAO : Default)	
<input type="checkbox"/>	MoveJGO (MoveGO : Default)	
<input type="checkbox"/>	MoveLGO (MoveGO : Default)	
<input type="checkbox"/>	TriggJ (Trigg : Default)	
<input type="checkbox"/>	TriggL (Trigg : Default)	
<input type="checkbox"/>	TriggJOs (TriggOs : Default)	
<input type="checkbox"/>	TriggLOs (TriggOs : Default)	
<input type="checkbox"/>	MoveAbsJ (MoveAbs : Default)	
<input type="checkbox"/>	MoveExtJ (MoveExt : Default)	
Erweitert		
Instruktionsvorlagen bearbeiten		

Definition der Bewegungsgeschwindigkeit



The screenshot shows the RobotStudio interface with a 3D model of a robot arm. A red box highlights a 'Speed' list on the right side of the interface. The list contains various speed values from v5 to v1000, with v1000 selected. A red arrow points to the 'v1000' entry. Below the speed list, there is a table with columns for 'Zeit' and 'Kategorie'.

Zeit	Kategorie
02.04.2024 10:06:07	Allgemein
02.04.2024 10:09:05	Allgemein
02.04.2024 10:19:19	Allgemein
02.04.2024 10:34:25	Allgemein
02.04.2024 10:39:19	Allgemein

Einstellen der
Bewegungsgeschwindigkeit

Definition des Überschleifradius

The screenshot shows the RobotStudio interface with a robotic arm model. A diagram in the bottom left illustrates the spherical overshoot factor with points P0, P1, and P2. A red box highlights the 'Zone' list in the bottom right, with 't100' selected.

Einstellen des Überschleifradius:
Je kleiner, umso genauer wird die Zielpose angefahren

Zeit	Kategorie
02.04.2024 10:06:07	Allgemein
02.04.2024 10:09:05	Allgemein
02.04.2024 10:19:19	Allgemein
02.04.2024 10:34:25	Allgemein
02.04.2024 10:39:19	Allgemein

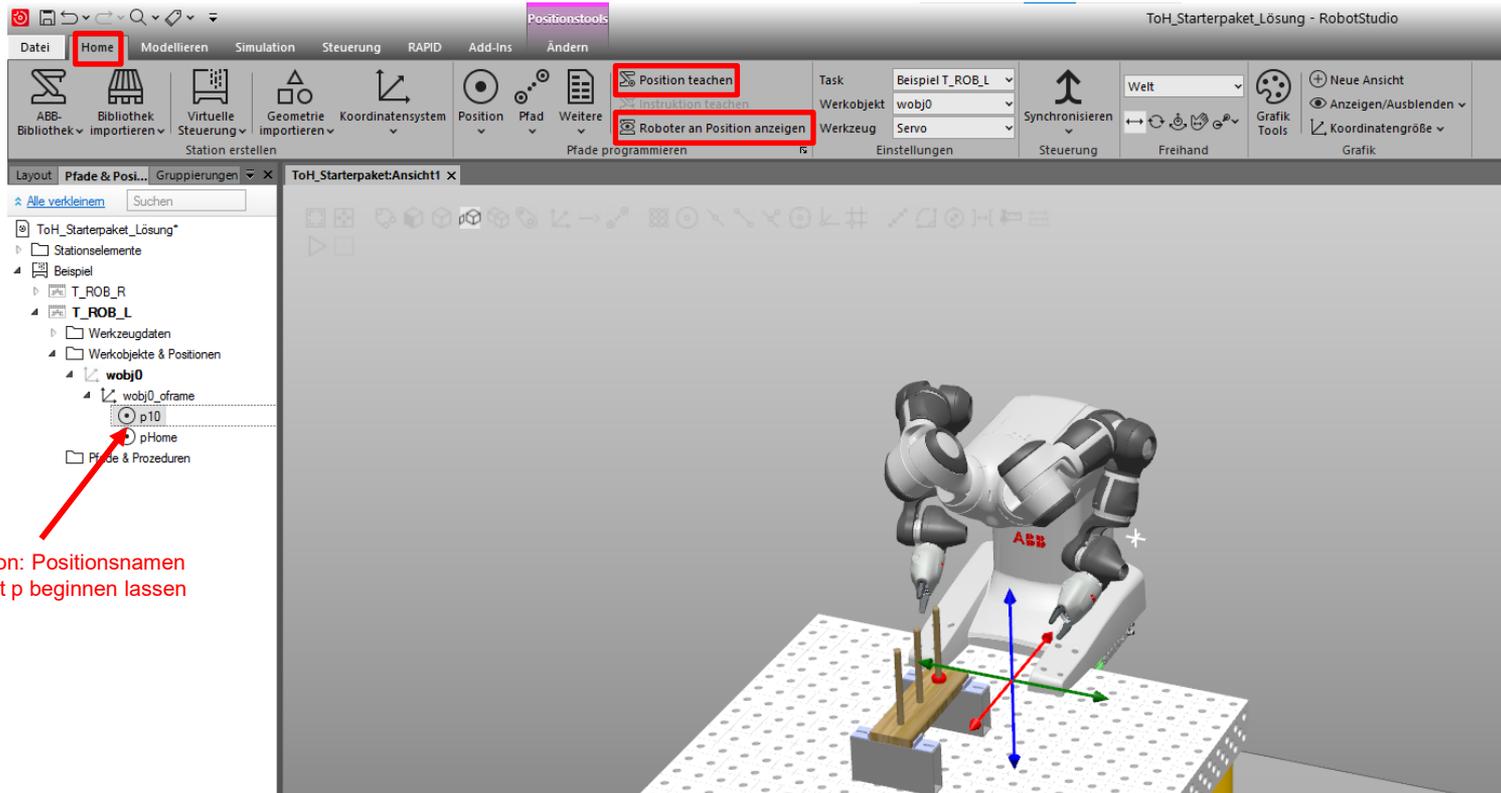
Zone
t100

P0 Startpunkt

P1 überschleifender Punkt

P2 Endpunkt

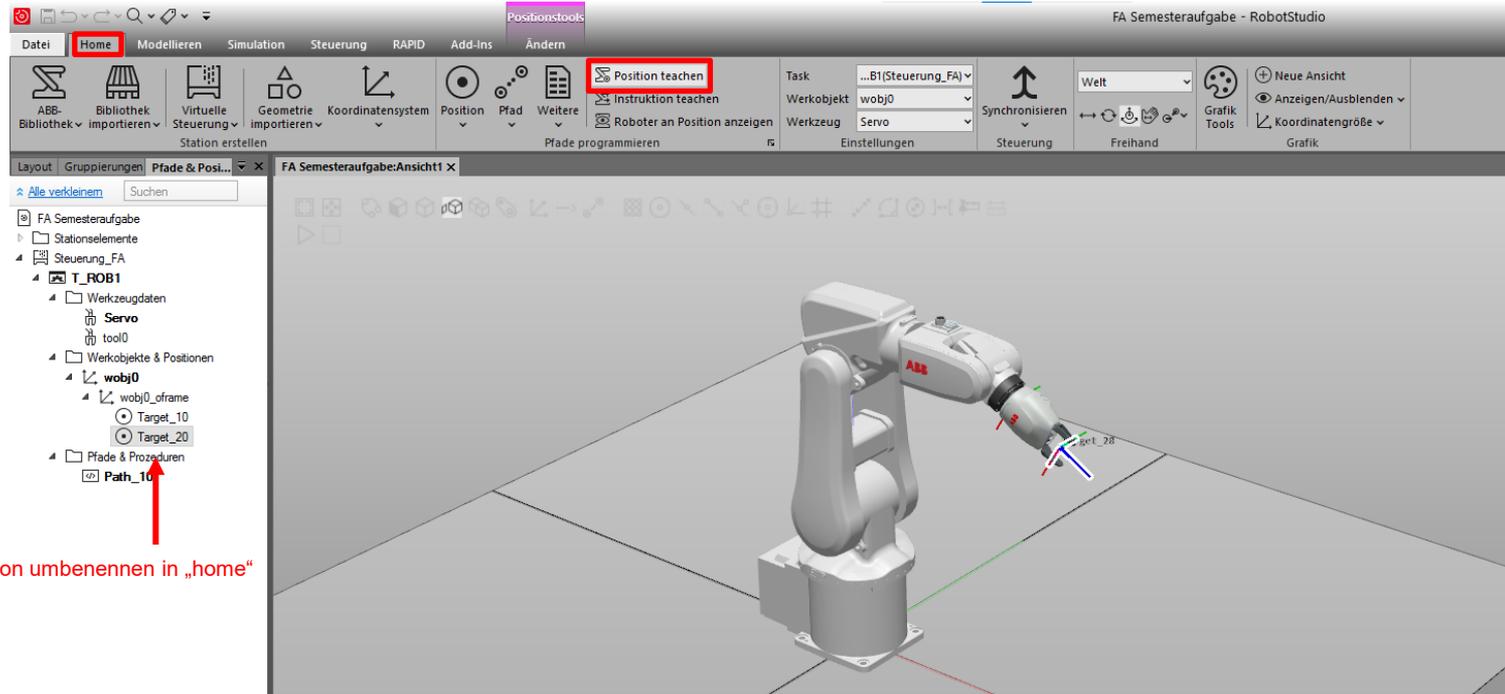
Position teachen



The screenshot displays the RobotStudio software interface. The 'Position tools' menu is open, showing options like 'Position teachen' and 'Roboter an Position anzeigen', both highlighted with red boxes. The 'Home' tab is selected in the top navigation bar. The left sidebar shows a tree view of the project structure, with 'p10' selected under 'wobj0'. The main 3D view shows a robot arm positioned over a workpiece on a table, with a coordinate system overlaid.

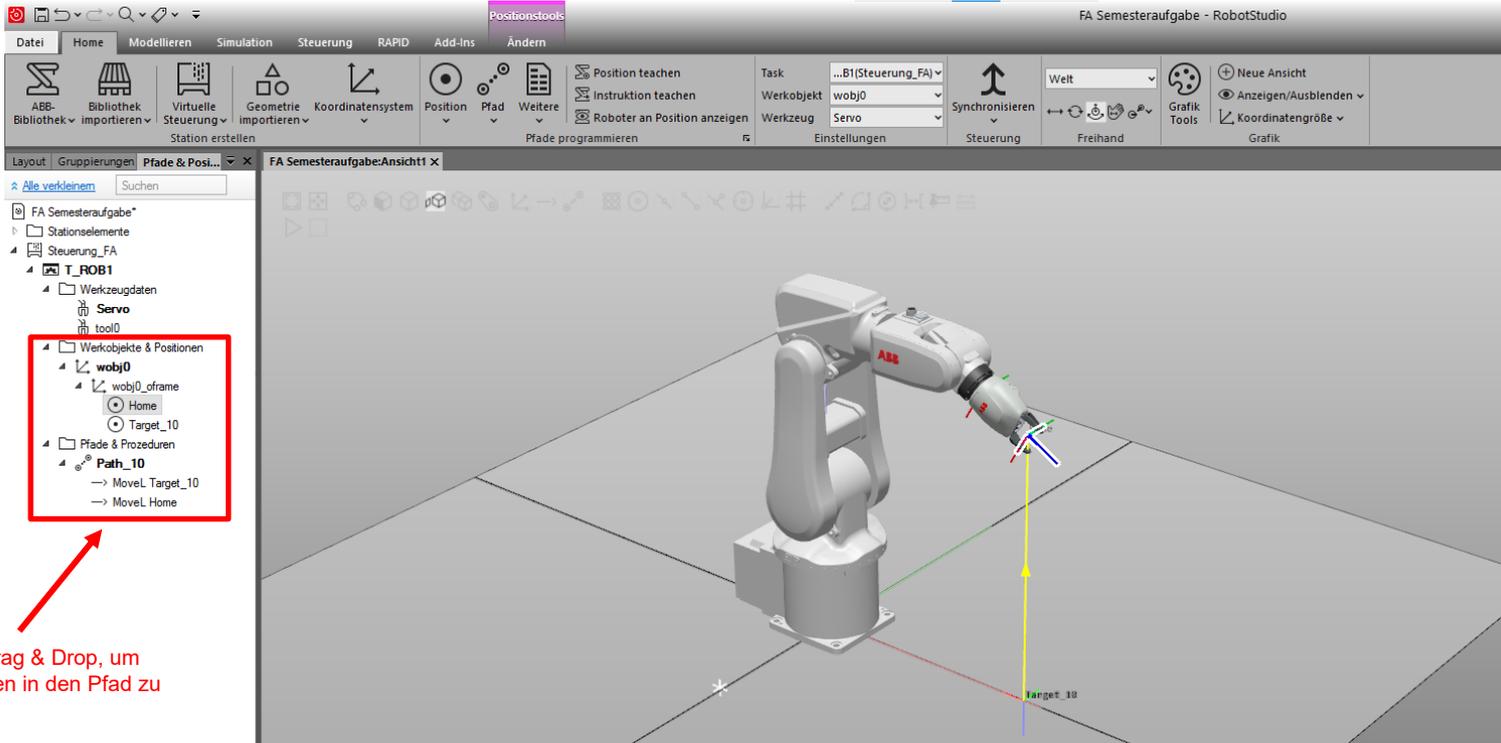
Konvention: Positionsnamen immer mit p beginnen lassen

Aktuelle Position teachen



Position umbenennen in „home“

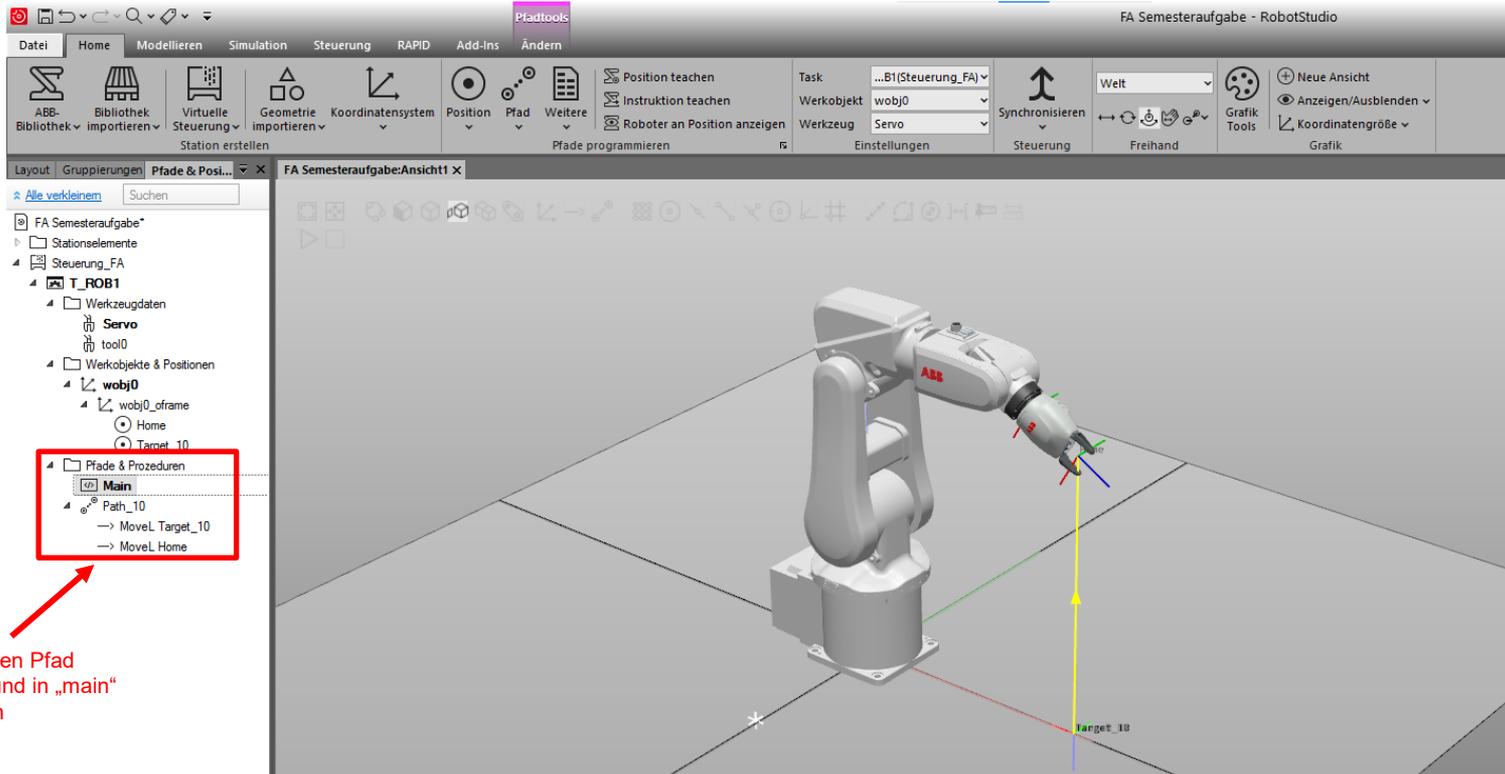
Bewegungsplan erstellen



The screenshot shows the RobotStudio interface for a task named 'FA Semesteraufgabe'. The left-hand tree view shows the project structure, with a red box highlighting the 'wobj0' position under 'Werkobjekte & Positionen'. The main 3D view shows a robotic arm model with a yellow arrow pointing to a 'Target_10' position on the floor. The task editor on the right shows the task name '...B1(Steuerung_FA)', the work object 'wobj0', and the tool 'Servo'. The task editor also shows a path 'Path_10' with two steps: 'MoveL Target_10' and 'MoveL Home'.

Nutze Drag & Drop, um
Positionen in den Pfad zu
ziehen

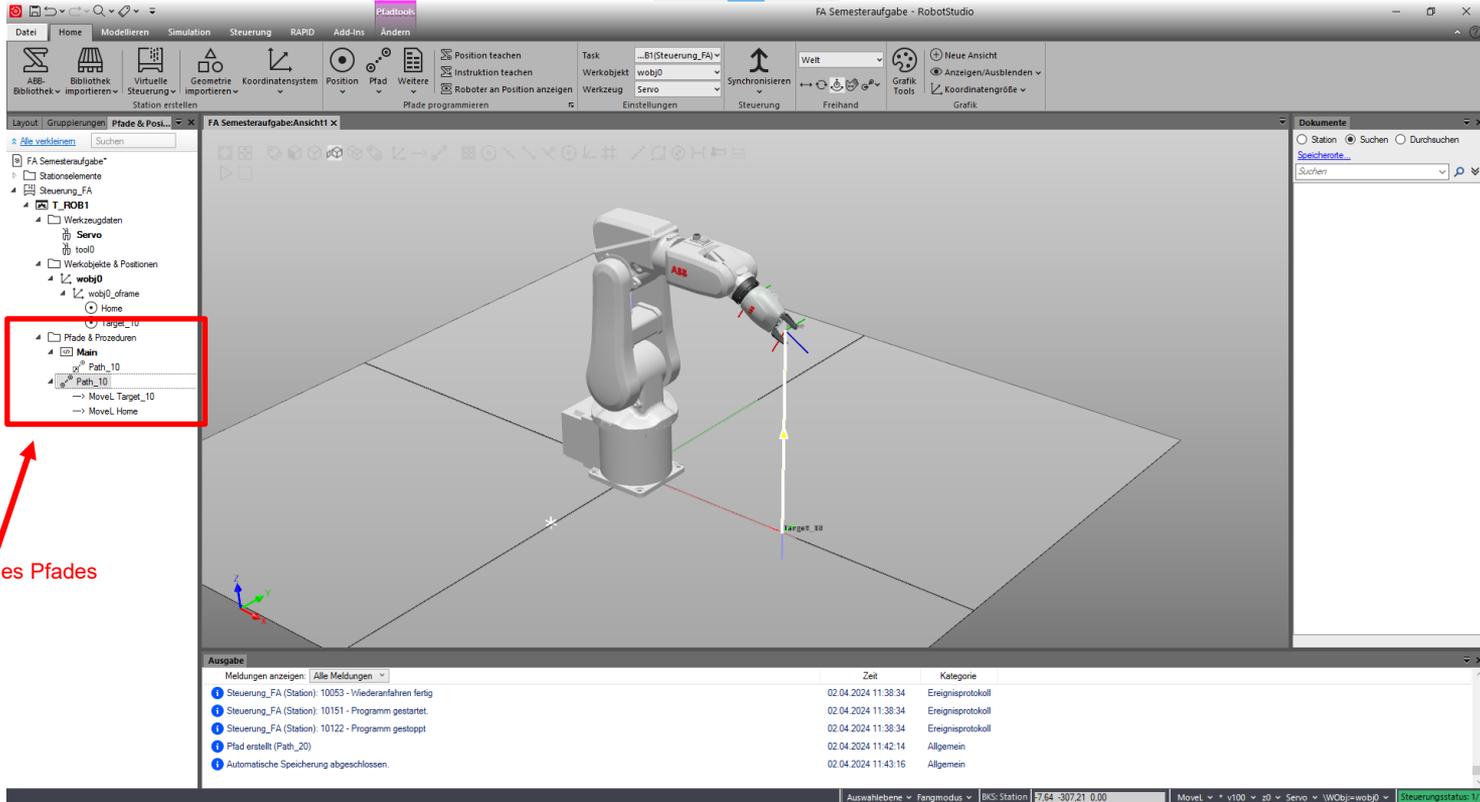
Erstellen einer Main



The screenshot displays the RobotStudio interface. The left sidebar shows a tree view of the project structure. The 'Main' path is highlighted in a red box. A red arrow points from the text below to this box. The main 3D view shows a robotic arm in a simulation environment.

Einigen leeren Pfad erstellen und in „main“ umbenennen

Pfad verknüpfen



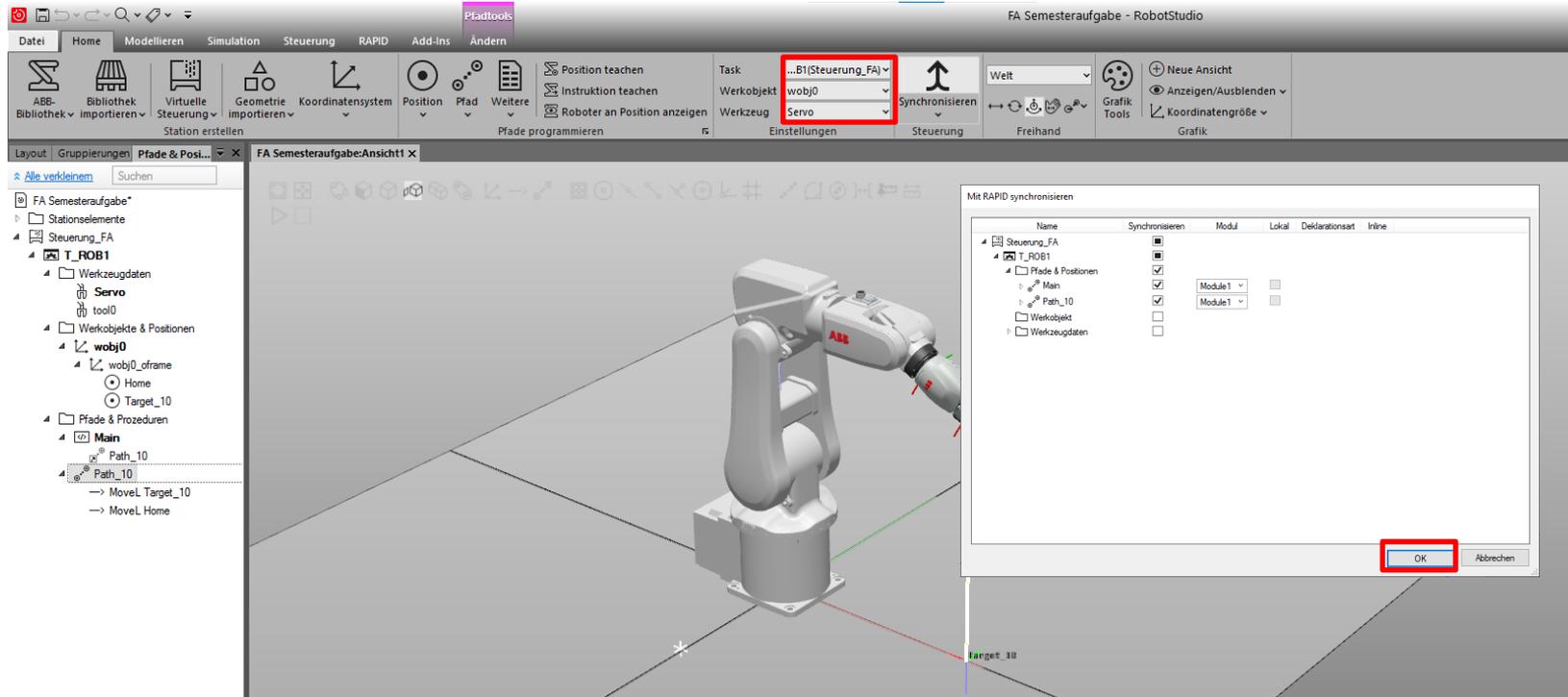
The screenshot shows the RobotStudio interface with a 3D model of a robot arm in the center. On the left, a tree view displays the project structure. A red box highlights the 'Main' folder under 'Path_10', which contains two sub-items: 'MoveL Target_10' and 'MoveL Home'. A red arrow points from the text 'Verknüpfen des Pfades mit der Main' to this box.

Verknüpfen des Pfades mit der Main

Ausgabe

Meldungen anzeigen:	Zeit	Kategorie
Steuerung_FA (Station): 10053 - Wiederanfahren fertig	02.04.2024 11:38:34	Ereignisprotokoll
Steuerung_FA (Station): 10151 - Programm gestartet	02.04.2024 11:38:34	Ereignisprotokoll
Steuerung_FA (Station): 10122 - Programm gestoppt	02.04.2024 11:38:34	Ereignisprotokoll
Pfad erstellt (Path_20)	02.04.2024 11:42:14	Allgemein
Automatische Speicherung abgeschlossen.	02.04.2024 11:43:16	Allgemein

Mit Steuerung synchronisieren

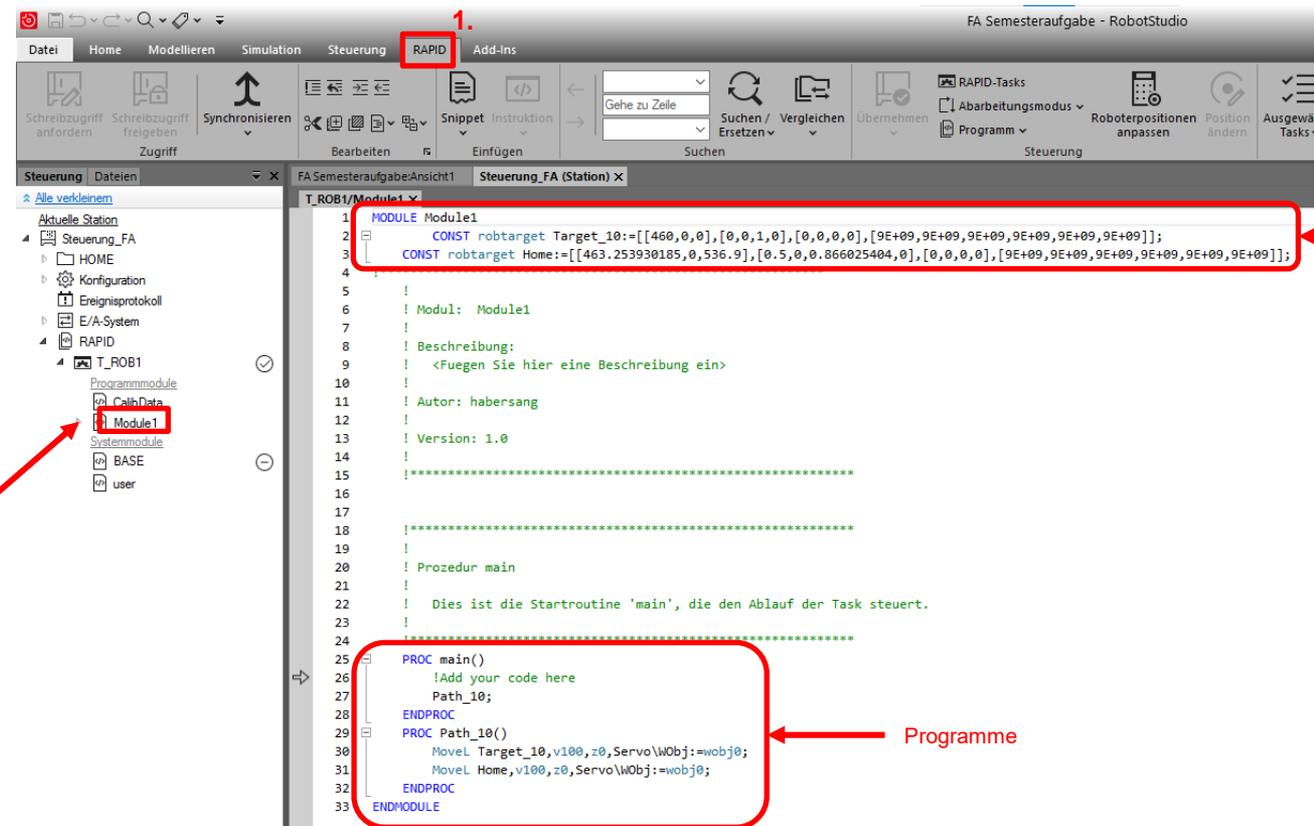


The screenshot shows the RobotStudio interface with the 'Synchronisieren' dialog box open. The dialog box is titled 'Mit RAPID synchronisieren' and contains a tree view of the project structure. The 'OK' button is highlighted with a red box.

The 'Synchronisieren' dialog box contains the following table:

Name	Synchronisieren	Modul	Lokal	Deklarationsart	Inline
Steuerung_FA	<input checked="" type="checkbox"/>				
T_ROB1	<input checked="" type="checkbox"/>				
Pfade & Positionen	<input checked="" type="checkbox"/>				
Main	<input checked="" type="checkbox"/>	Modul1	<input type="checkbox"/>		
Path_10	<input checked="" type="checkbox"/>	Modul1	<input type="checkbox"/>		
Werkobjekt	<input checked="" type="checkbox"/>				
Werkzeugdaten	<input type="checkbox"/>				

Ablaufsteuerung in Rapid-Code



The screenshot shows the RobotStudio interface with the RAPID code editor open. The file explorer on the left shows the project structure, with 'Module1' selected under 'T_ROB1'. The code editor displays the following RAPID code:

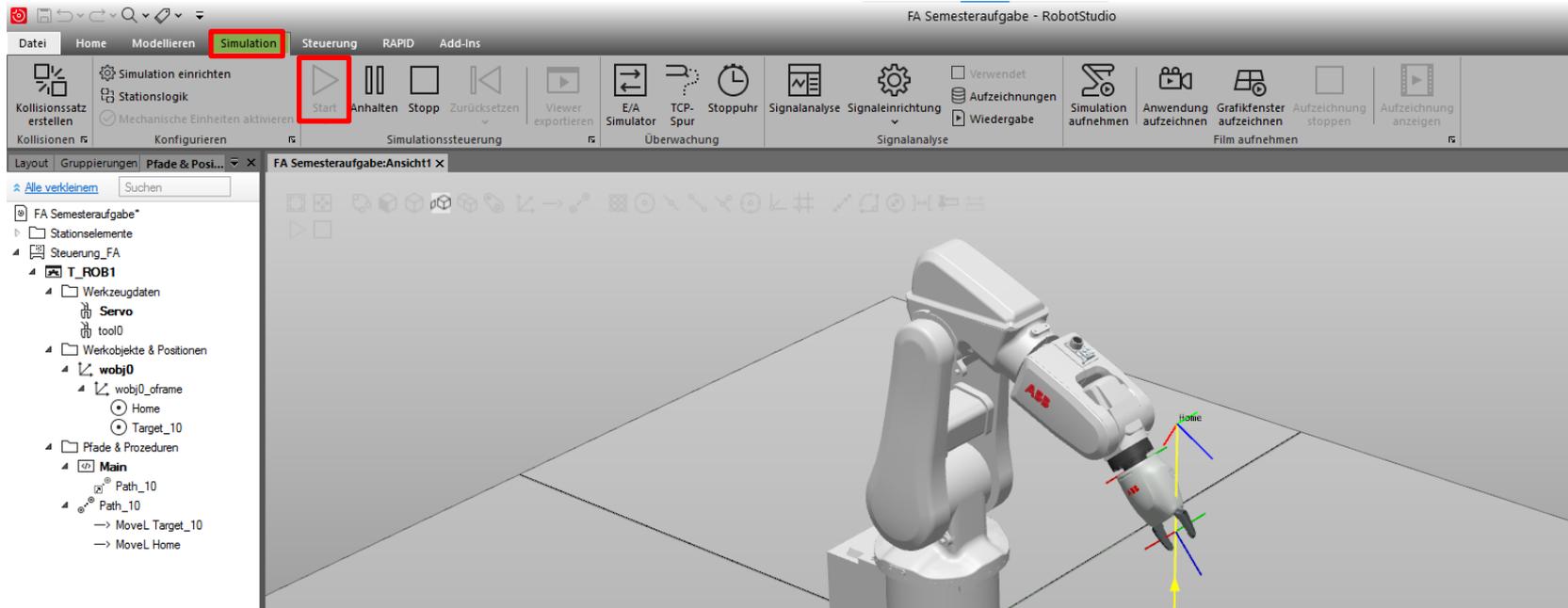
```
1 MODULE Module1
2   CONST robtarget Target_10:=[[460,0,0],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
3   CONST robtarget Home:=[[463.253930185,0,536.9],[0.5,0,0.866025404,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
4
5   !
6   ! Modul: Module1
7   !
8   ! Beschreibung:
9   ! <Fuegen Sie hier eine Beschreibung ein>
10  !
11  ! Autor: habersang
12  !
13  ! Version: 1.0
14  !
15  !-----
16
17
18  !-----
19  !
20  ! Prozedur main
21  !
22  ! Dies ist die Startroutine 'main', die den Ablauf der Task steuert.
23  !
24  !-----
25  PROC main()
26    !Add your code here
27    Path_10;
28  ENDPROC
29  PROC Path_10()
30    MoveL Target_10,v100,z0,Servo\Wobj:=wobj0;
31    MoveL Home,v100,z0,Servo\Wobj:=wobj0;
32  ENDPROC
33  ENDMODULE
```

Annotations in the image:

- 1.** A red box highlights the 'RAPID' tab in the top menu bar.
- 2. Doppelklick** (Double-click) with a red arrow pointing to 'Module1' in the file explorer.
- Positionen** (Positions) with a red arrow pointing to the 'CONST robtarget' lines in the code.
- Programme** (Programs) with a red arrow pointing to the 'PROC main()' and 'PROC Path_10()' lines in the code.

Ausführen der Simulation

Wichtig:
Vor dem Ausführen der Simulation
Synchronisieren nicht vergessen



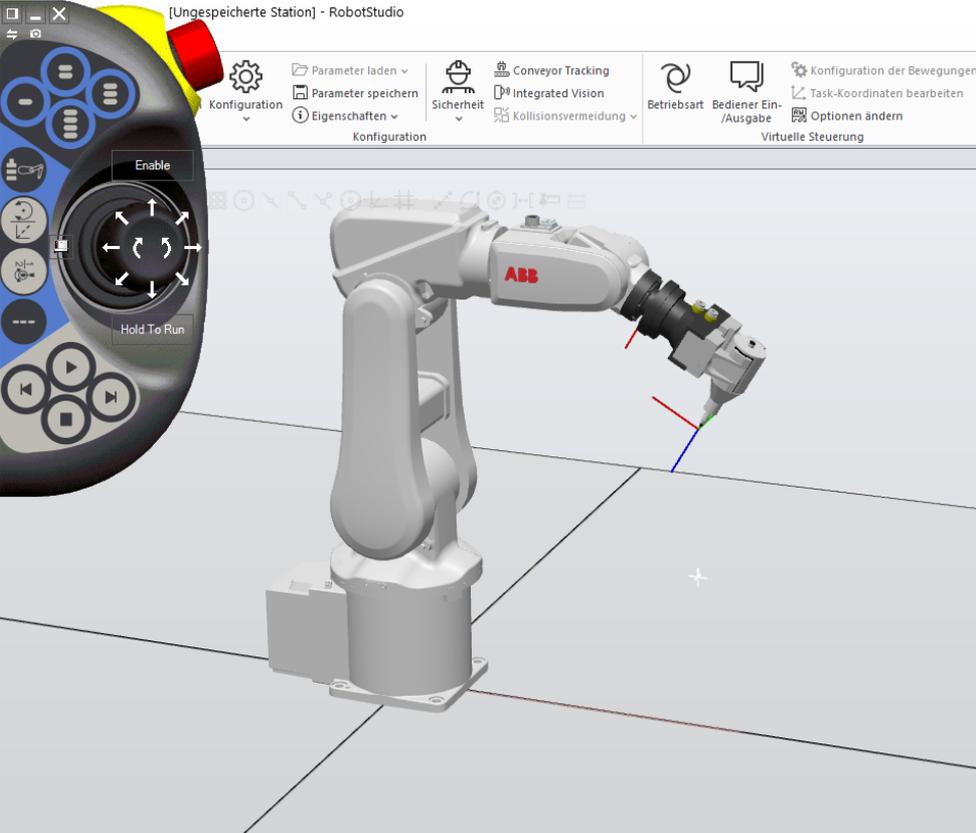
Einführung in Bedienfläche des YuMi

Wechsel in das Fenster „Bewegen“

1.   Einrichtung NBM-SSA1 Sicherheitsstopp
Prog. gestoppt (Geschw. 100%) 

2.  **Bewegen**

 HotEdit	 Backup und Restore
 Eingänge und Ausgänge	 Kalibrierung
 Produktion	 Systemeinstellungen
 Programm Editor	 Ereignislog
 Programm Daten	 FlexPendant Explorer
 Abmelden Default User	 Neustart
	 Systeminformationen

 [Ungespeicherte Station] - RobotStudio

Konfiguration

- Parameter laden
- Parameter speichern
- Eigenschaften
- Sicherheit
- Conveyor Tracking
- Integrated Vision
- Kollisionsvermeidung
- Betriebsart
- Bediener Ein-/Ausgabe
- Konfiguration der Bewegungen
- Task-Koordinaten bearbeiten
- Optionen ändern
- Virtuelle Steuerung

ROB_1
1/3



Bewegen: Bewegungsarten

Einrichtbetrieb
NBM-SSA1

Motoren ein
Prog. gestoppt (Geschw. 100%)

Bewegen

Tippen Sie auf eine Eigenschaft, um sie zu ändern.

Mech. Einheit: ROB_1...

Absolutgenauigkeit: Off

Bewegungsart: Achse 1-3...

Koordinatensystem: Welt...

Werkzeug: tool0...

Werkobjekt: wobj0...

Nutzlast: load0...

Joystick block.: Keine...

Inkremente: Groß...

Position	1:	2:	3:	4:	5:	6:
	0.00	0.00	0.00	0.00	30.00	0.00

Positionsformat...

Joystickrichtungen

2 1 3

Ausrichten... Bewegen zu Aktivieren...

Bewegen

ROB_1
1/3

Bewegungsarten:
Auswahl der Bewegungsmuster, die mit dem Joystick umgesetzt werden können

„Sicherheitstaster“:

Stellt den Sicherheitstaster an der Seite des FlexPendant nach. Motor kann im Handbetrieb nur mit betätigtem Taster betrieben werden.

Joystick:

Lenkt den Roboter an den jeweiligen Achsen (Siehe das Fenster „Joystickrichtungen“)

Einrichtbetrieb
NBM-SSA1

Motoren ein
Prog. gestoppt (Geschw. 100%)

Bewegen - Bewegungsart

Aktuelle Auswahl: Achse 1-3

Bewegungsart wählen:

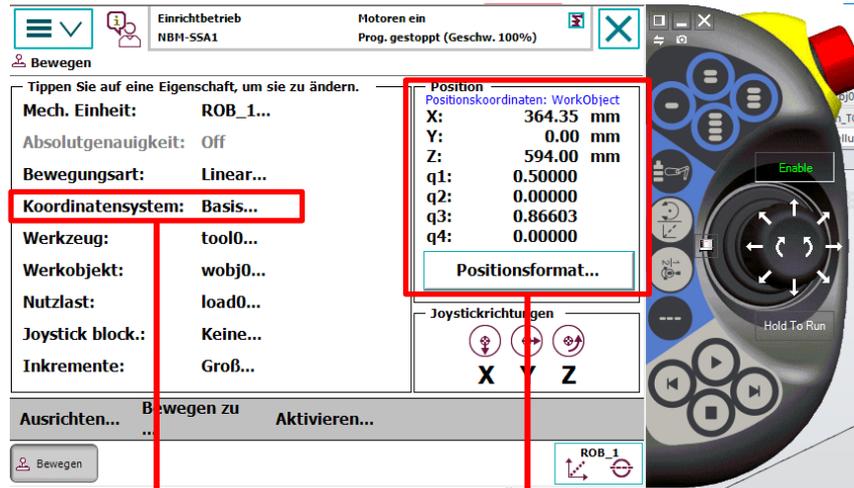
Achse 1-3 Achse 4-6 Linear Umorient.

OK Abbrechen

Bewegen

ROB_1
1/3

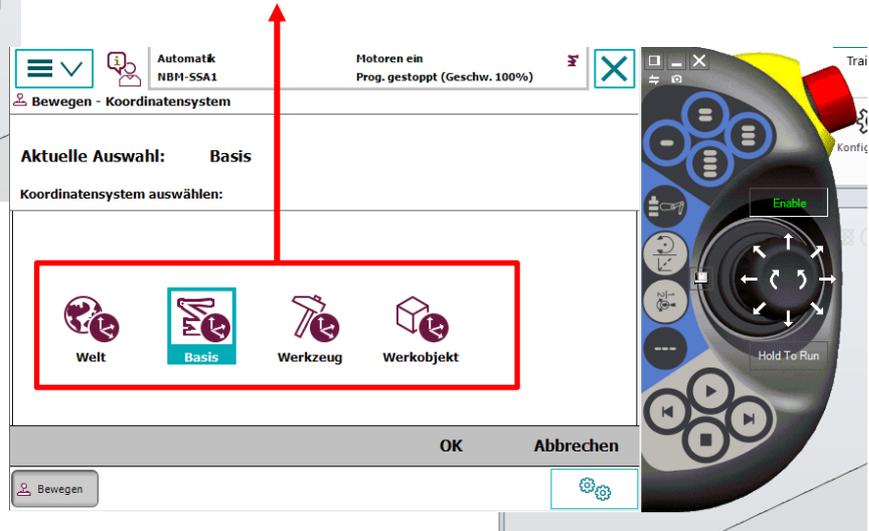
Bewegen: Koordinatensystem



Bezugskoordinatensystem:
In diesem Feld kann das Koordinatensystem ausgewählt werden, auf welches referenziert werden soll

Positionskoordinaten:
Hier werden die Positionsdaten in Relation zu dem ausgewählten Koordinatensystem aufgezeigt

Auswählbare Koordinatensysteme:
Hier ist die Auswahl der vier grundlegenden Koordinatensysteme aufgeführt, auf die Bezug genommen werden kann



Bewegen: Werkzeug

Einrichtbetrieb NBM-SSA1 Motoren ein Prog. gestoppt (Geschw. 100%)

Bewegen

Tippen Sie auf eine Eigenschaft, um sie zu ändern.

Mech. Einheit: ROB_1...

Absolutgenauigkeit: Off

Bewegungsart: Linear...

Koordinatensystem: Basis...

Werkzeug: tool0...

Werkobjekt: wobj0...

Nutzlast: load0...

Joystick block.: Keine...

Inkrement: Groß...

Position
Positionskordinaten: WorkObject
X: 364.35 mm
Y: 0.00 mm
Z: 594.00 mm
q1: 0.50000
q2: 0.00000
q3: 0.86603
q4: 0.00000

Positionsformat...

Joystickrichtungen
X Y Z

Ausrichten... Bewegen zu Aktivieren...

Werkzeugauswahl:
In diesem Feld kann das Werkzeug ausgewählt werden, dass an dem Roboterflansch angebracht ist

Auswählbare Werkzeuge:
Hier sind die auswählbaren Werkzeuge aufgeführt. Es besteht zudem die Möglichkeit, ein neues Werkzeug zu erstellen.

Einrichtbetrieb NBM-SSA1 Motoren ein Prog. gestoppt (Geschw. 100%)

Bewegen - Werkzeug

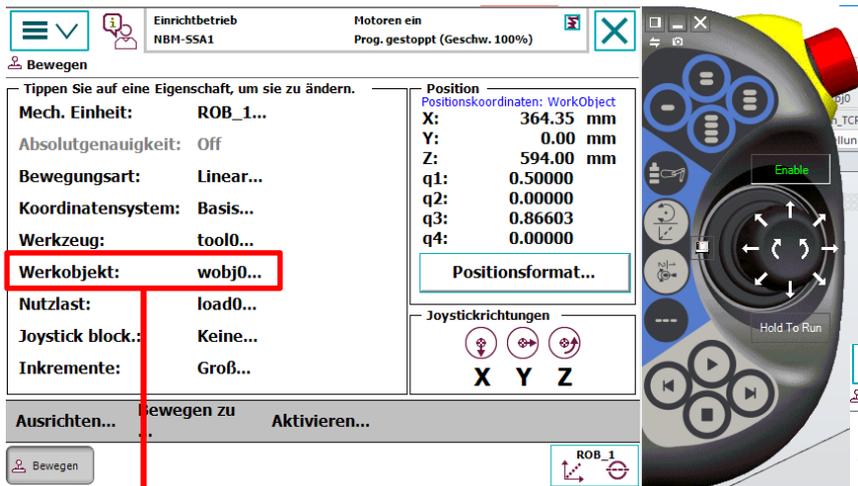
Aktuelle Auswahl: tPen

Ein Objekt aus der Liste auswählen.

Name des Werkzeugs	Modul	Bereich
tool0	RAPID/T_ROB1/BASE	Global
tPen	RAPID/T_ROB1/Module1	Task

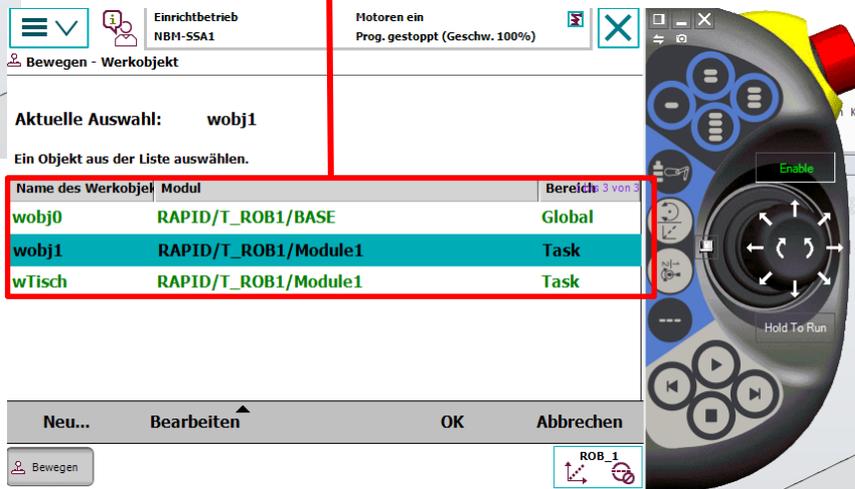
Neu... Bearbeiten OK Abbrechen

Bewegen: Werkobjekt

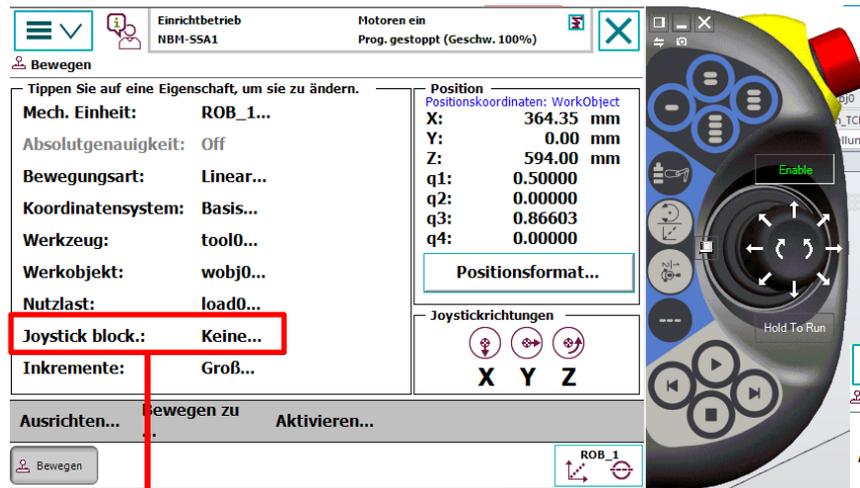


Referenziertes Werkobjekt:
Hier wird das Werkobjekt aufgeführt, auf welches sich referenziert wird.

Auswählbare Werkobjekte:
Hier werden die nutzbaren Werkobjekte angezeigt. Es besteht zudem die Möglichkeit, ein neues Werkobjekt zu erstellen.



Bewegen: Joystick blockieren

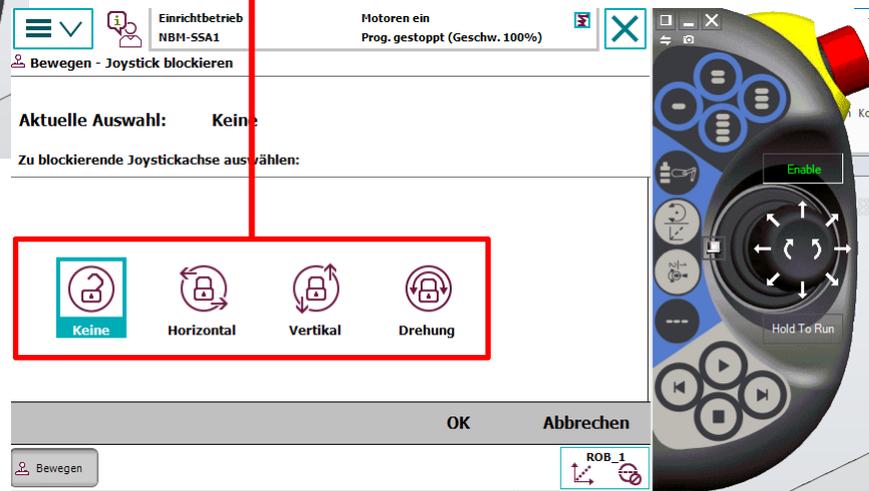


Richtungssperre:

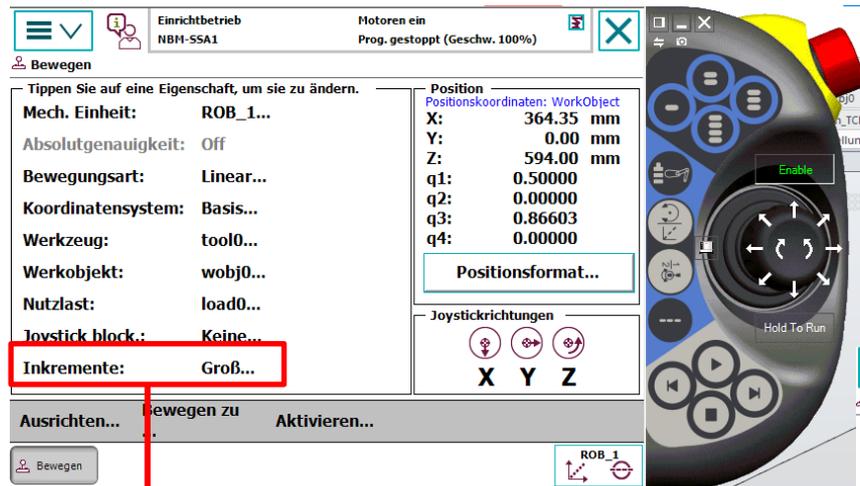
Hier können Achsen blockiert werden, in welche nicht verfahren werden soll. Dies kann zur Sicherheit dienen, wenn der Roboter beispielsweise nicht mit einem Objekt kollidieren soll.

Auswählbare Richtungen:
Richtungswahl des zu sperrenden
Verfahrweges.

Wichtig: Bei der linearen Bewegung soll die Z-Achse gesperrt werden, damit Kollisionen mit dem Tisch vermieden werden.

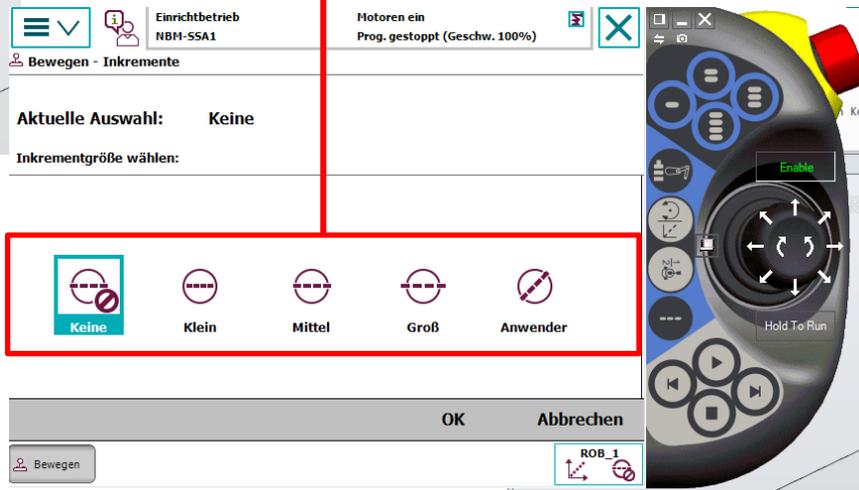


Bewegen: Inkremente



Bewegungssensitivität:
Die Sensitivität des Roboters auf Bewegungseingaben mit dem Joystick werden hier ausgewählt.

Wahl des Bewegungsinkrements:
Die genauen Schrittgrößen bei der Bewegung des Joysticks werden hier aufgezeigt.



Bewegen: Ausrichten

Einrichtbetrieb
NBM-SSA1

Motoren ein
Prog. gestoppt (Geschw. 100%)

Bewegen

Tippen Sie auf eine Eigenschaft, um sie zu ändern.

Mech. Einheit: ROB_1...

Absolutgenauigkeit: Off

Bewegungsart: Linear...

Koordinatensystem: Basis...

Werkzeug: tool0...

Werkobjekt: wobj0...

Nutzlast: load0...

Joystick block.: Keine...

Inkremente: Groß...

Position
Positionskoordinaten: WorkObject
X: 364.35 mm
Y: 0.00 mm
Z: 594.00 mm
q1: 0.50000
q2: 0.00000
q3: 0.86603
q4: 0.00000

Positionsformat...

Joystickrichtungen
X Y Z

Ausrichten... Bewegen zu Aktivieren...

Ausrichtungsfenster:
Hier kann das gewünschte Koordinatensystem ausgewählt und durch das Halten der „Ausrichten starten“ Schaltfläche die Positionsanpassung initiiert werden.

Einrichtbetrieb
NBM-SSA1

Sicherheitsstopp
Prog. gestoppt (Geschw. 100%)

Bewegen - Ausrichten

Aktuelles Werkzeug: tool0

1. Koordinatensystem zum Ausrichten des Werkzeugs auswählen:
KoordSys.: Welt

2. Zustimmungsschalter betätigen und auf 'Ausrichten starten' tippen.

Ausrichten starten

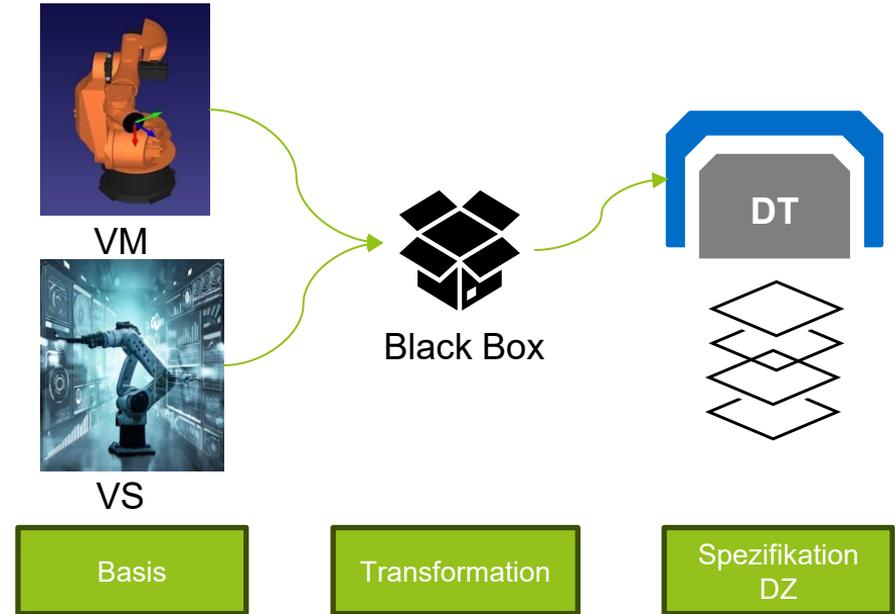
3. Nach Abschluss auf "Schließen" tippen.

Schließen

Automatische Ausrichtung:
In diesem Bereich kann der Roboter automatisch zu einem Koordinatensystem ausgerichtet werden.

Einführung:

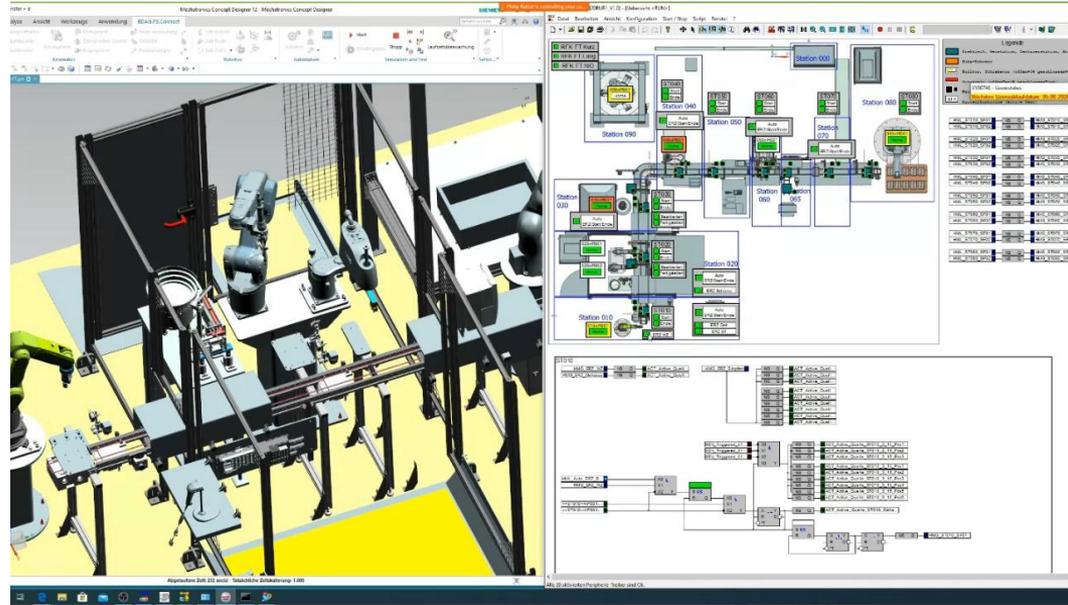
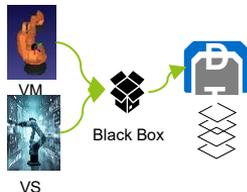
- Ausgangspunkt VM und VS
- Transformationsprozess
- Transformationsschritte im Detail oft unbekannt
- Spezifikation des DZ multidimensional
→ Abbildungsraum entscheidend



TRANSFORMATION ZUM DIGITALEN ZWILLING

Beispiel einer Roboterzelle

- Robotermodell RoboDK
- Virtueller Schatten (Echtzeit Sensordaten von Webcam)
- Vernetzung Simulation mit Robotersteuerung
- Präsentation Livestream Kamera, Robotersimulation und Roboterbewegung in der LFF



Datenerfassung

- Die Grundlage jedes DZ sind Informations- und Datenmodelle sowie Echtdaten
- Erfolg oder Misserfolg des DZ hängen stark von der Datenqualität ab
- Daten müssen korrekt und präzise sein, zur Gewährleistung eines erfolgreichen Einsatzes
- Bestandsdaten und mittels Sensoren generierte Daten werden erhoben und aufbereitet

Virtuelles Modell

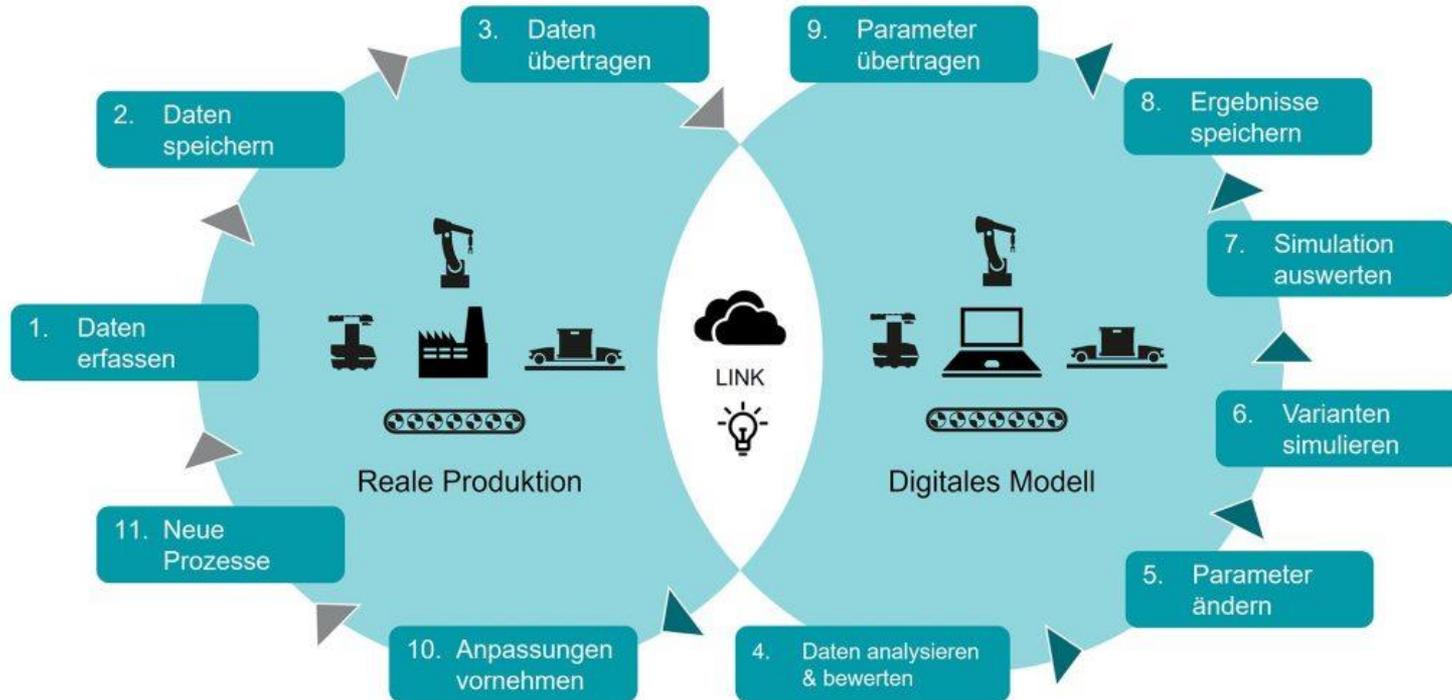
- Auf Basis der erhobenen Daten wird das virtuelle Modell erstellt
- Bietet die digitale Abbildung der Prozesse und Anlagen und gibt einen ersten Überblick über die Abläufe
- Bereits in dieser Phase unterstützt ein DZ bei der Auswahl der richtigen Anlage bzw. bei der Wahl des Systems

Simulation & Analyse

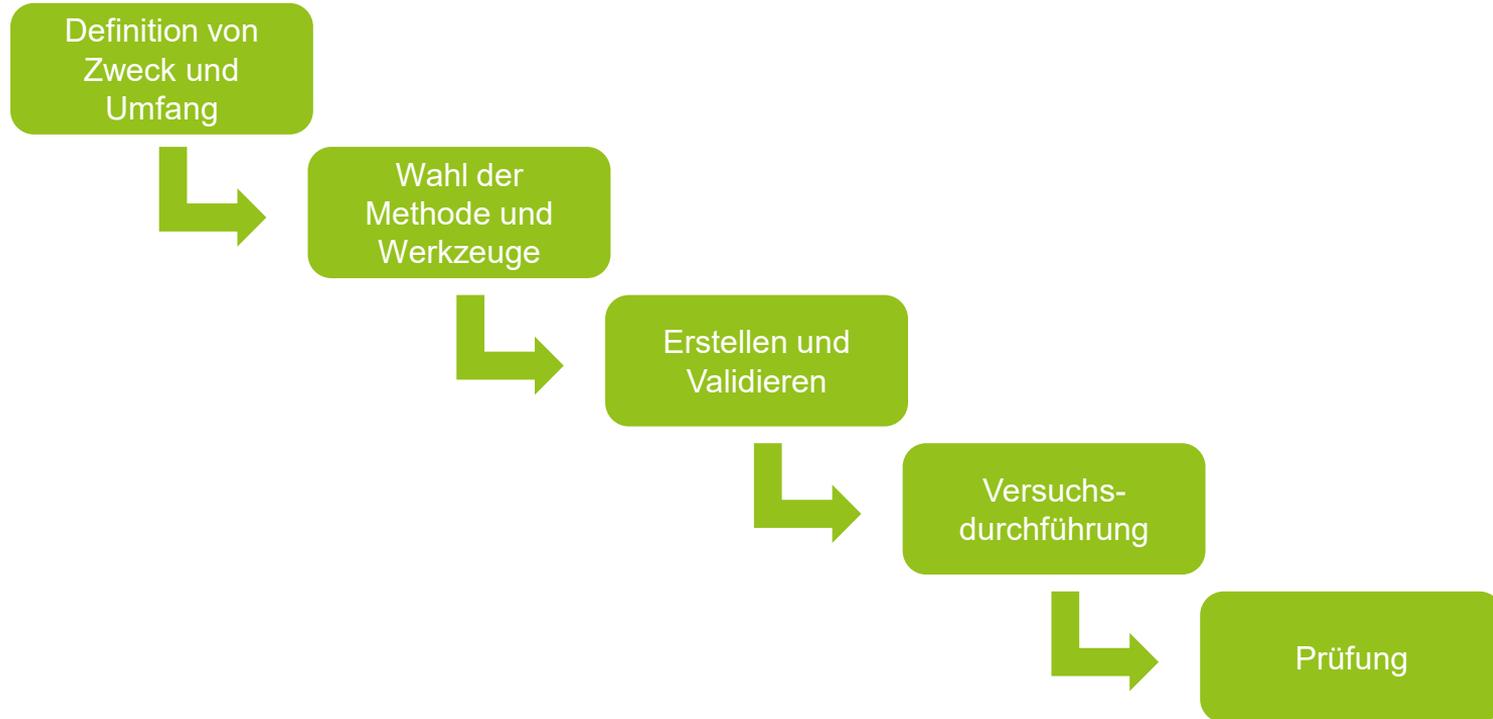
- Der DZ simuliert Prozesse und testet mit Hilfe von Algorithmen der künstlichen Intelligenz verschiedene Szenarien
- So können Unternehmen Abläufe in der virtuellen Umgebung optimieren, bevor sie in den Echtbetrieb übergehen

Überwachung & Optimierung

- Die Software überwacht kontinuierlich die Abläufe im Prozess und vergleicht die tatsächliche Leistung mit dem virtuellen Modell
- Bei Abweichungen können sofort Maßnahmen zur Verbesserung gestartet werden
- Änderungen in den Prozessen können parallel zum laufenden Betrieb evaluiert werden
- Effektivität und Effizienz können digital getestet werden



TRANSFORMATION STEP-BY-STEP ANLEITUNG



TRANSFORMATION

STEP-BY-STEP ANLEITUNG

1. Definition von Zweck und Umfang

- Es ist wichtig, klare Grenzen festzulegen und genau zu wissen, was modelliert werden soll und welchen Zweck dies erfüllt. Identifizieren Sie die Kernelemente, die das Verhalten des Systems wesentlich beeinflussen. Verdeutlichung der Systemdynamik.
- Gezielte Vereinfachung bei Entwurf und Planung. Wo liegen die Grenzen des Vorhabens und wo treten Probleme auf? Wie viel Detail und Komplexität wird benötigt?
- Priorisierung von Erkenntnissen gegenüber erschöpfenden Details. Welche relevanten Variablen möchte ich messen? Welche sind die wichtigsten Annahmen, die ich testen möchte? Welche Fragen möchte ich unbedingt beantwortet haben?
- Die Kernstruktur des Modells muss flexibel sein, um eine notwendige Erweiterung des Umfangs zu ermöglichen.
- Es sollte ein iterativer Verfeinerungsprozess verwendet werden, anstatt zu versuchen, das Modell beim ersten Versuch zu perfektionieren.

TRANSFORMATION

STEP-BY-STEP ANLEITUNG

2. Wahl der Methode und Werkzeuge

- Es steht eine Auswahl verschiedener Methoden und Werkzeuge für die Erstellung zur Verfügung. Beispiele sind Systemdynamik, agentenbasierte Modellierung, ereignisdiskrete Simulation oder Monte-Carlo-Simulation. Da jede Methode ihre eigenen Vor- und Nachteile hat, müssen Sie berücksichtigen, wie gut sie zu Ihrem Problem, Ihren Daten und Ihren Ressourcen passt.
- Die agentenbasierte Modellierung ermöglicht die Modellierung des Verhaltens und der Interaktionen einzelner Agenten innerhalb eines Systems.
- Die Systemdynamik konzentriert sich auf das Verständnis der Interaktion von Variablen im Laufe der Zeit, um das Verhalten komplexer Systeme zu simulieren.
- Die ereignisdiskrete Simulation modelliert die Abfolge von Ereignissen in einem System, um Prozesse und Arbeitsabläufe zu analysieren.

TRANSFORMATION

STEP-BY-STEP ANLEITUNG

3. Erstellen und Validieren

- Ein Modell ist eine vereinfachte Darstellung der Realität, die auf Annahmen, Daten und Logik basiert.
- Ein Modell sollte genau, konsistent und transparent sein sowie aussagekräftige und zuverlässige Ergebnisse liefern.
- Die Struktur, das Verhalten und die Ausgaben des Modells sollten anhand verschiedener Beweisquellen wie historischer Daten, Expertenmeinungen oder logischer Argumente überprüft und getestet werden.
- Zu Beginn der Erstellung werden die Systemkomponenten definiert, einschließlich Entitäten, Prozesse und Interaktionen.
- Die Elemente werden mit der gewählten Methode und dem gewählten Werkzeug visuell dargestellt.
- Anschließend werden Parameter, Variablen und Anfangsbedingungen basierend auf verfügbaren Daten und Annahmen angegeben.
- Im nächsten Schritt wird die Logik und die Algorithmen implementiert, um das Verhalten des Systems unter Berücksichtigung dynamischer Beziehungen und Rückkopplungsschleifen genau zu simulieren.
- Das Modell wird validiert, indem die simulierten Ergebnisse mit realen Beobachtungen oder historischen Daten verglichen werden.

TRANSFORMATION

STEP-BY-STEP ANLEITUNG

4. Versuchsdurchführung

- Ein Szenario ist eine Reihe von Bedingungen und Ereignissen, die eine mögliche zukünftige Situation definieren.
- Szenarien basieren auf Zielen, Unsicherheiten und Risiken.
- Durch den Vergleich und die Gegenüberstellung verschiedener Szenarien lassen sich die besten oder schlechtesten Ergebnisse, die wichtigsten Treiber und die kritischen Unsicherheiten identifizieren.

5. Prüfung

- Die Erstellung ist keine einmalige Aktivität, sondern ein fortlaufender Prozess.
- Sobald neue Informationen, Daten oder Feedback verfügbar sind, sind regelmäßige Überprüfungen und Aktualisierungen notwendig.
- So bleibt die Relevanz, Genauigkeit und Qualität erhalten und es wird sichergestellt, dass sich die sich ändernde Realität und die Bedürfnisse Ihres Problems und Ihrer Zielgruppe widerspiegeln.

-
1. Definieren Sie den klaren Unterschied zwischen einem Digitalen Zwilling und einer Simulation.
 2. Wie lässt sich die Umsetzung eines Digitaler Zwillingen realisieren? Skizzieren Sie alle Schritte.
 3. Welche Komponenten sind für die Entwicklung eines Digitalen Zwillingen entscheidend?

Erstellung eines digitalen Zwillings

Standardisierungskonzepte

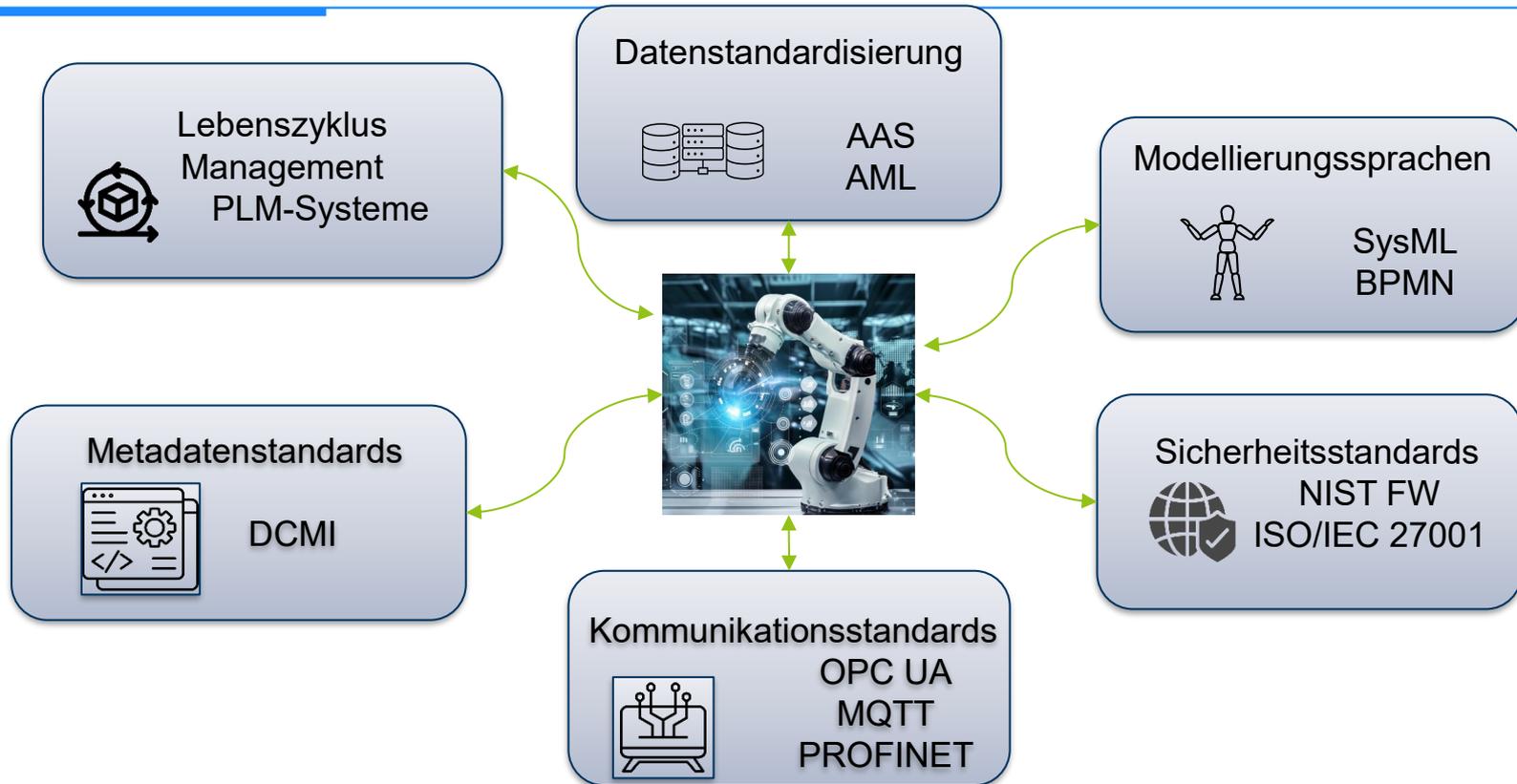
Datenstandardisierung

Metadatenstandards

Kommunikationsstandards

Modellierungssprachen

...





- [-] **AAS**AAS_Type_EMME_AS_40_S_LV_AS (2024-07-26 14:36:43.331)
 - [-] **Sub**Nameplate (2024-07-26 14:36:43.331)
 - Prop**URIOfTheProduct (2024-07-26 14:36:43.331)
 - Lang**ManufacturerName = de FESTO SE & Co. KG en (2024-07-26 14:36:43.331)
 - Lang**ManufacturerProductDesignation = de Zuverlässiger und leistungsstarker AC Synchron-Servomotor mit optimierter Anschlusstechnik für dynamische Anwendungen. en (2024-07-26 14:36:43.331)
 - Lang**ManufacturerProductRoot = de Servomotor en (2024-07-26 14:36:43.331)
 - Lang**ManufacturerProductFamily = de EMME-AS en (2024-07-26 14:36:43.331)
 - Lang**ManufacturerProductType = de EMME-AS-40-S-LV-AS en (2024-07-26 14:36:43.331)
 - Lang**OrderCodeOfManufacturer = en EMME-AS-40-S-LV-AS (2024-07-26 14:36:43.331)
 - Lang**ProductArticleNumberOfManufacturer = en 2082428 (2024-07-26 14:36:43.331)
 - Prop**SerialNumber (2024-07-26 14:36:43.331)
 - Prop**YearOfConstruction = 2024 (2024-07-26 14:36:43.331)
 - Prop**CountryOfOrigin = DE (2024-07-26 14:36:43.331)
 - File**CompanyLogo = /aasx/Nameplate/FestoLogo.png (2024-07-26 14:36:43.331)
 - [+] **Coll**ContactInformation #11 (2024-07-26 14:36:43.331)
 - [+] **Coll**Markings #5 (2024-07-26 14:36:43.331)

Die Verwaltungsschale

- Standardisiertes, digitales Abbild eines Assets (Roboter, Endeffektor, Motor etc.)
- Schnittstelle zwischen physischer und digitaler Welt
- Umfasst alle relevanten Informationen
→ Standardisierung erfolgt durch Verwendung von Teilmodellen



Teilmodell 1



Teilmodell 2



Teilmodell 3



Teilmodell 4

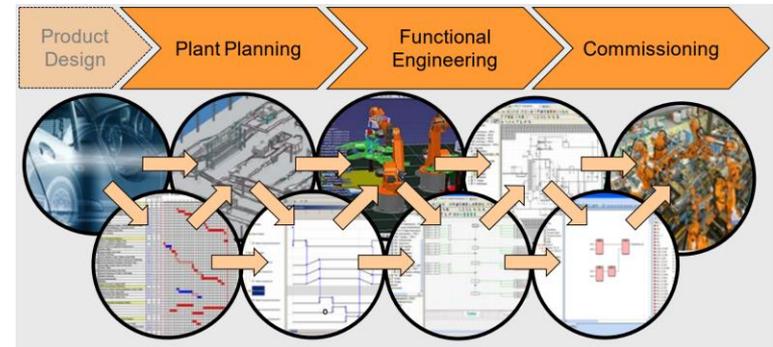


<AutomationML/>
Next Level Engineering

```
<InstanceHierarchy ID="AF138E59" Name="InstanceHierarchy">
  <InternalElement ID="AF138E59" Name="Station">
    <ExternalInterface Name="LogicInterfaceGantt,">
      <Attribute Name="refURI"
        AttributeDataType="xs:anyURI">
        <Value>file:///LogicXML/UUID_001</Value>
      </Attribute>
    </ExternalInterface>
  </InternalElement>
</InstanceHierarchy>
```

AutomationML

- XML-basierte Datenmodellierungssprache
- Ermöglicht das Modellieren, Speichern und Austauschen von Engineering Daten
- Einsatz im Bereich des Anlagenentstehungsprozesses für den herstellerunabhängigen Datenaustausch



<https://www.automationml.org/>



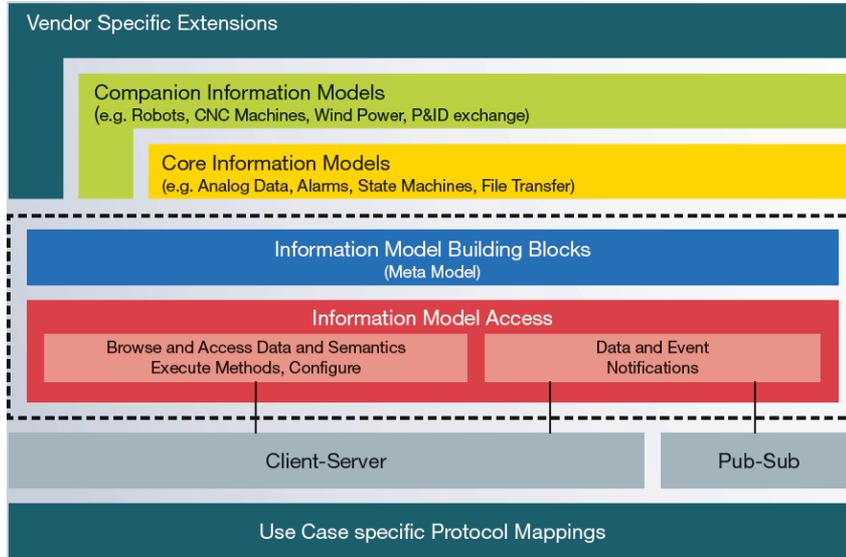
```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
    <dc:title>Homage to Catalonia,</dc:title>
    <dc:creator>Orwell, George, 1903-1950.</dc:creator>
    <dc:type>text</dc:type>
    <dc:publisher>London, Secker and Warburg</dc:publisher>
    <dc:date>[1938]</dc:date>
    <dc:language>eng</dc:language>
  </rdf:Description>
</rdf:RDF>
```

▪ DCMI (Dublin Core Metadata Initiative)

- Standard für die Beschreibung von Metadaten
- Besteht aus 15 Basiselementen (Titel, Ersteller, Thema, etc.)
- Erster Entwurf entstand im Jahr 1995

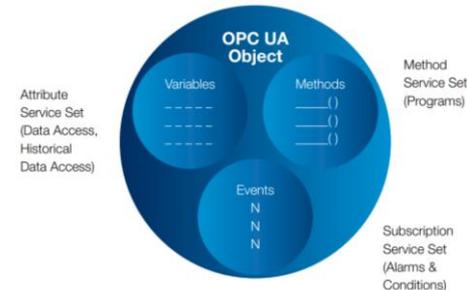
▪ Was sind Metadaten?

- Daten über Daten
- Informationen, die ein Datenobjekt (Asset, Website, Bild etc.) beschreiben

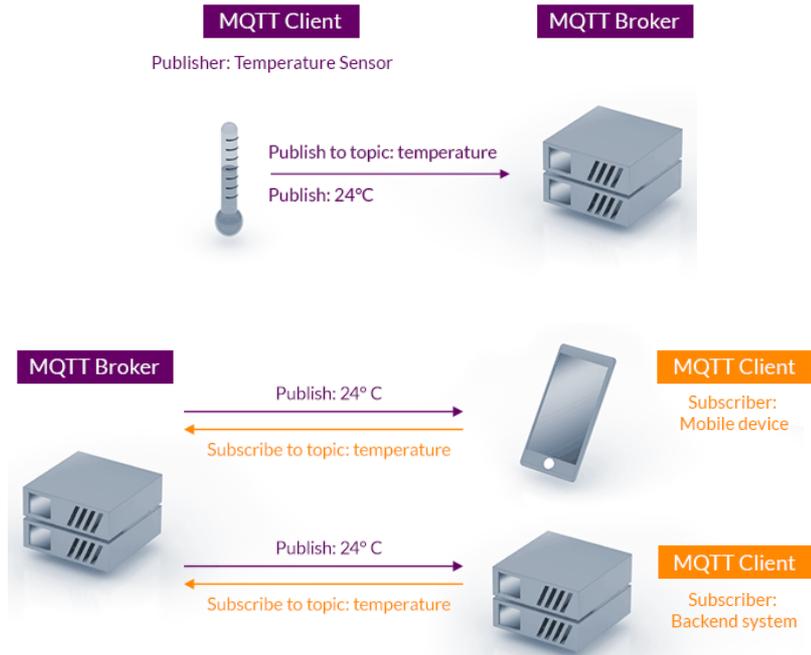


■ OPC Unified Architecture

- Standardisiertes Framework für die industrielle Interoperabilität und Kommunikation
- Gewährleistet die sichere Übertragung von Daten durch Protokolle und Services
- Spezifizieren die Schnittstellen zwischen Clients und Servern



ISBN 978-3-8343-3454-1, <https://opcfoundation.org/about/opc-technologies/opc-ua/>



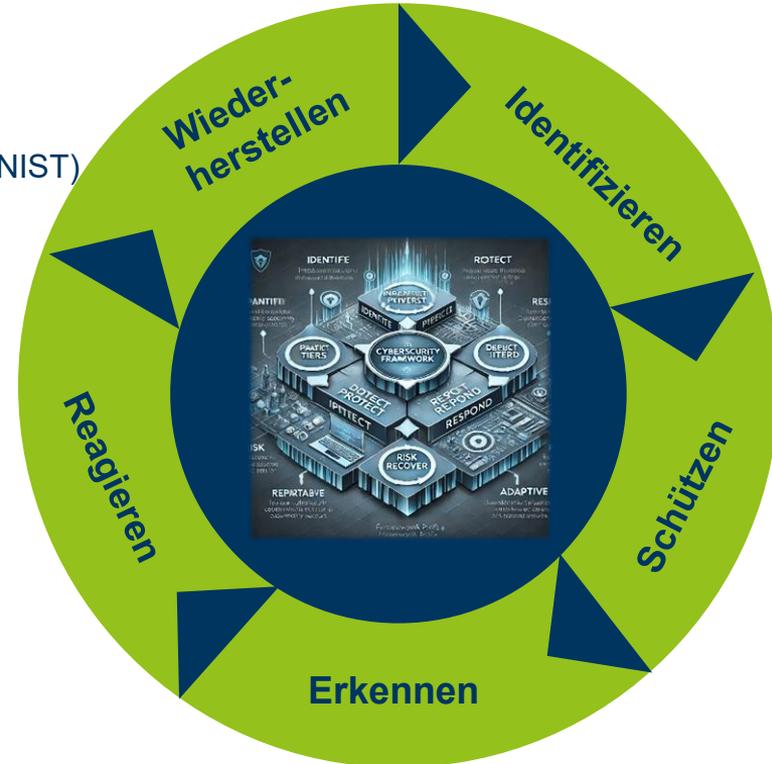
■ MQTT (Message Queuing Telemetry Transport)

- Standardisiertes Kommunikationsprotokoll
- Leichtgewichtiger Publish/Subscribe Mechanismus durch minimalen Ressourcenverbrauch
- MQTT-Client veröffentlicht Nachrichten an einen Broker und kann Nachrichten von anderen Clients abonnieren
- MQTT-Broker empfängt Nachricht und leitet diese an Abonnenten weiter

SICHERHEITSSTANDARDS

NIST FRAMEWORK

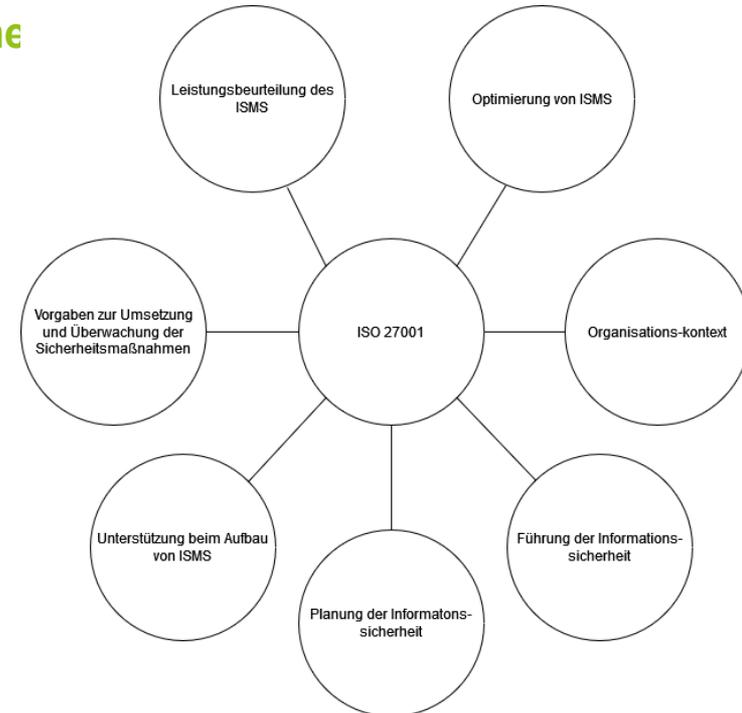
- **Strukturierter, ganzheitlicher Ansatz zur Verbesserung von Cybersicherheit**
 - Entwickelt vom National Institute of Standards and Technology (NIST)
- **3 Hauptkomponenten**
 - Framework Kern
 - Umsetzungsebenen
 - Framework Profil



SICHERHEITSSTANDARDS

ISO/IEC 27001

- **Standard für IT Sicherheitsmanagement Systeme (ISMS)**
 - Spezifikation von Anforderungen für ISMS
- **Umfasst Listen und Ergänzungen zu ...**
 - Kontrollzielen und Sicherheitsmaßnahmen
 - Informationsrichtlinien
 - Physische Sicherheitsmaßnahmen
 - Kryptografie
 - Notfallmanagement
- **Vorteile:**
 - KVP Unterstützung
 - Kundenzufriedenheit steigernd



Online ISBN: 978-3-658-27692-8,
<https://link.springer.com/ISO27001>

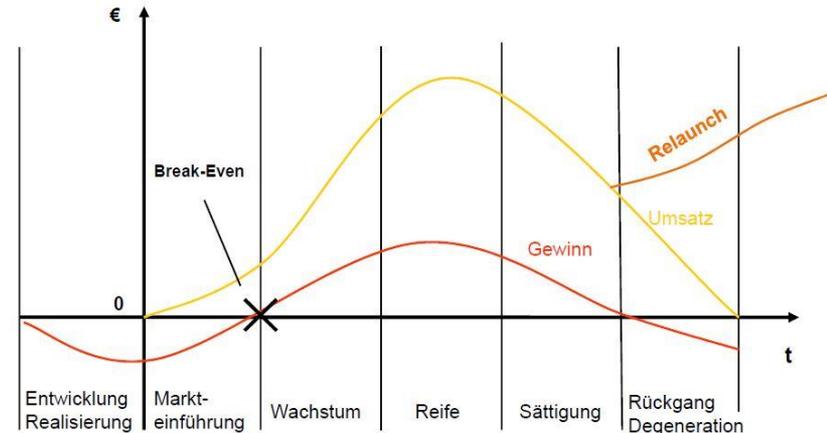
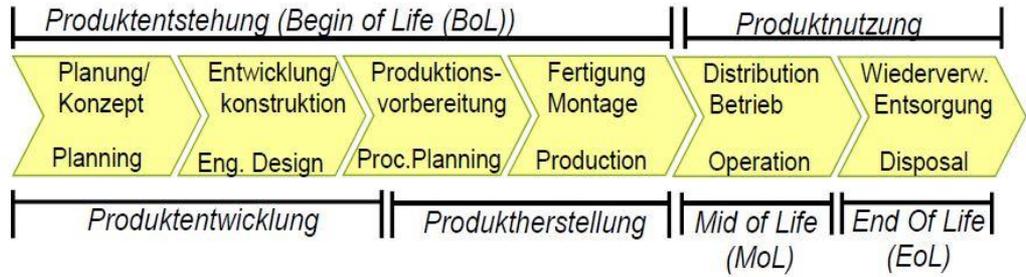
PRODUKT-LEBENSZYKLUS-MANAGEMENT (PLM)

- **verschiedene Sichten**

- Betriebswirtschaftlich
- Engineering
- Ökologisch (ISO 14001)

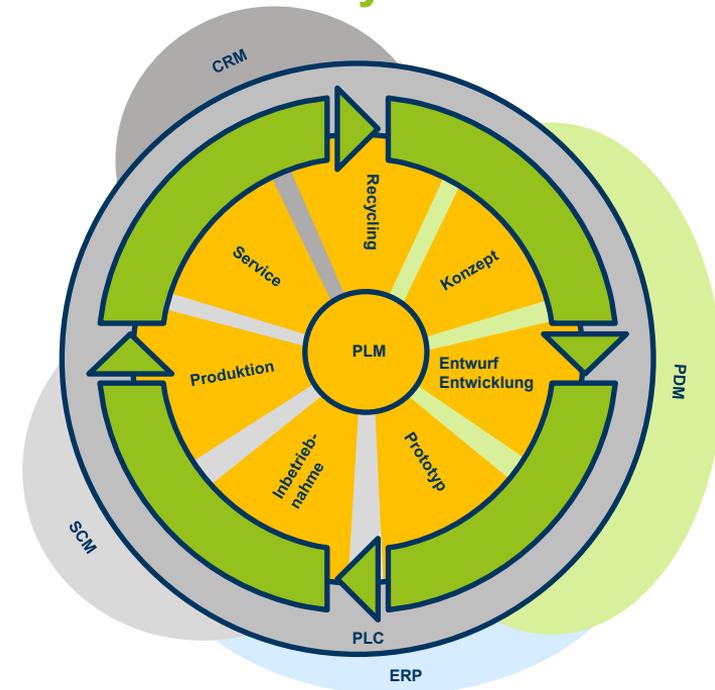
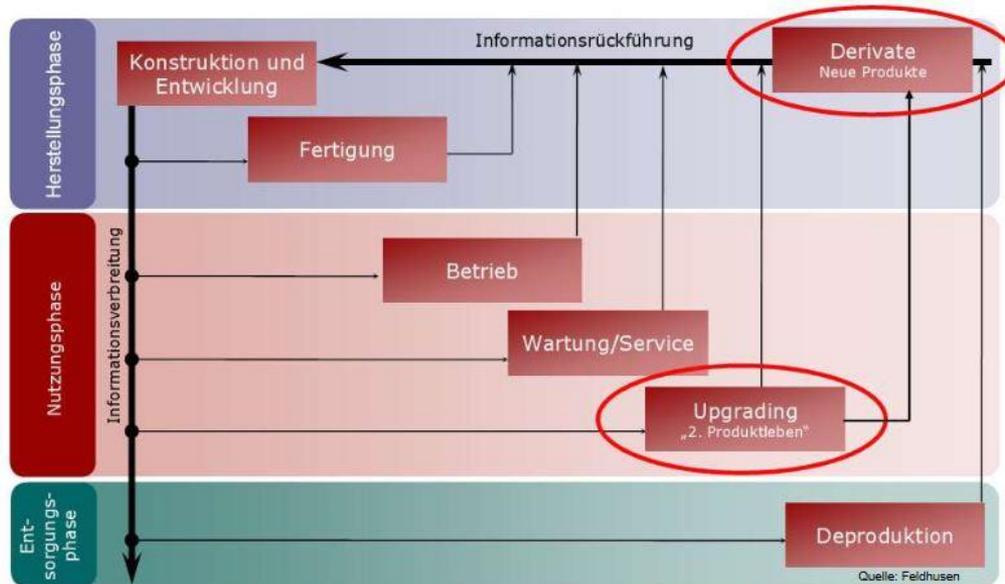
- **ganzheitliche Betrachtung einer Anlage**

- DZ kann Informationen aus allen Phasen kombinieren



PRODUKT LEBENSZYKLUS MANAGEMENT (PLM)

- Konzept zum Management von Wissen über Produktionssysteme über gesamten Lebenszyklus



1. Nennen Sie die 6 Arten von Standards, welche einen entscheidenden Beitrag zur Einführung eines DZ leisten können.
2. Was ist AAS für eine Art von Standard? Was besagt das Konzept der Verwaltungsschale?
3. Was versteht man unter Metadaten? Wofür steht DCIM?
4. Was ist MQTT für eine Art von Standard und wofür wird dieser verwendet?
5. Was sind Vorteile des PROFINET Standards?
6. Benennen sie die 5 Kernfunktionen des NIST Frameworks.
7. Welche Art von Standard wird in der ISO/IEC 27001 definiert und welche Bedeutung ist diesem zuzusprechen für den Einsatz von DZ?
8. Skizzieren Sie die Phasen des Produktlebenszyklus (betriebswirtschaftlich) und Erläutern Sie warum eine Lebenszyklusbetrachtung für den Einsatz eines DZ von besonderer Relevanz ist.
9. Wofür steht PLM? Was unterscheidet PLM-Systeme von dem Konzept eines DZ?
10. Wie kann der Einsatz einer PLM-Systemlösung und der Einsatz von DZ komplementär aussehen. Erläutern Sie die Synergien anhand eines Beispiels.

...

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

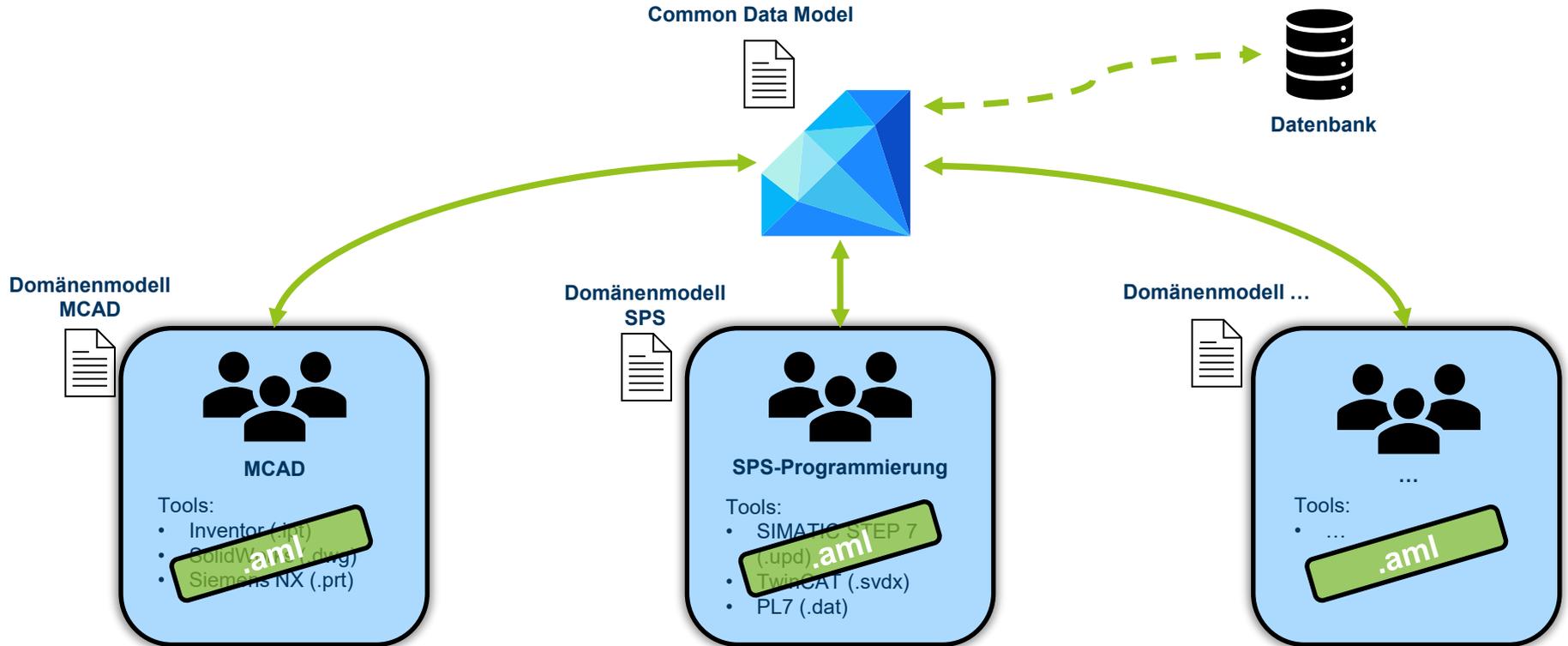
Erstellung eines digitalen Zwillings

Standardisierungskonzepte

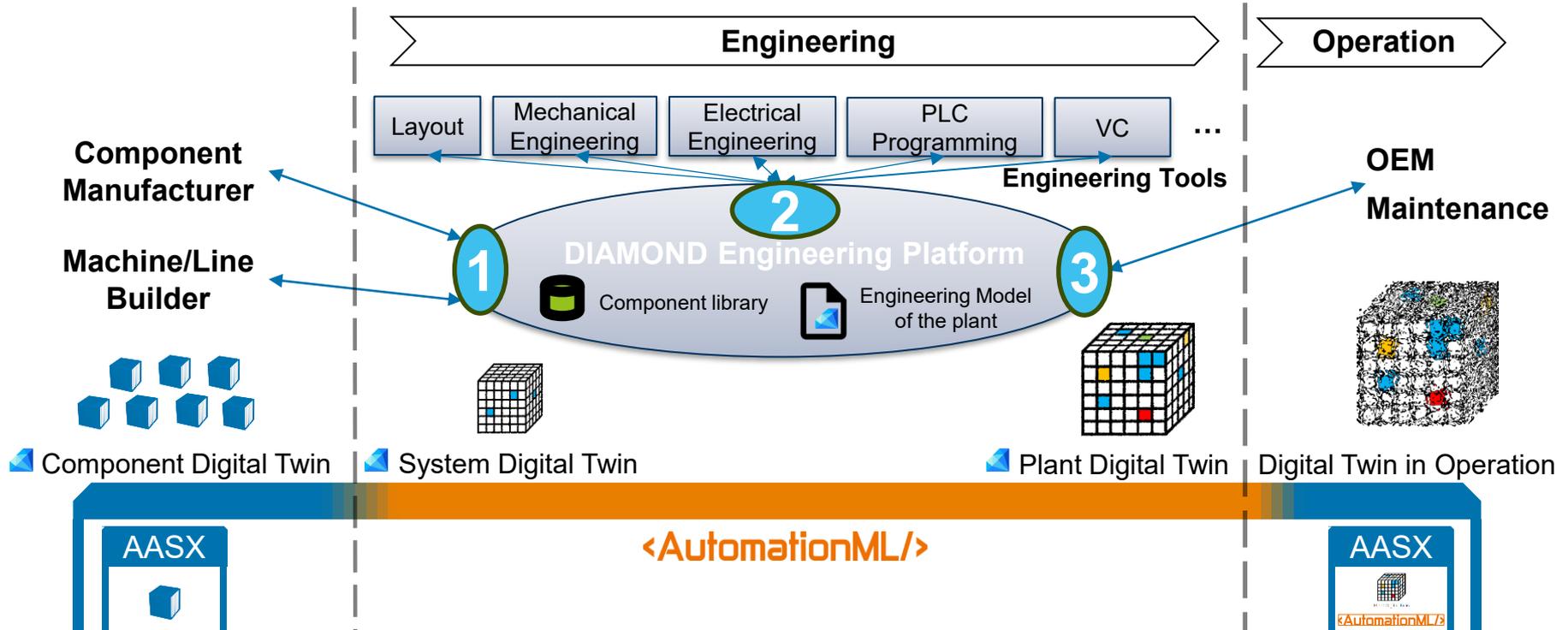
Ausblick

Literatur

ENTWICKLUNG EINES COMMON DATA MODELS



BIG PICTURE DIAMOND KONZEPT



...

Motivation

Der digitale Zwilling im Anlagenentstehungsprozess

Erstellung eines digitalen Zwillings

Standardisierungskonzepte

Ausblick

Literatur

1. [Digitaler Zwilling - Gesellschaft für Informatik e.V.](#)
2. https://link.springer.com/content/pdf/10.1007/978-3-662-58474-3_43.pdf
3. [Digital Twin: Manufacturing Excellence through Virtual Factory Replication \(researchgate.net\)](#)
4. [NASA Modeling, Simulation, Information Technology & Processing - TA11 \(emacromall.com\)\]](#)
5. https://link.springer.com/content/pdf/10.1007/978-3-662-63956-6_10.pdf
6. [Digitaler Zwilling für die Realisierung Utility 4.0 | SpringerLink](#)
7. [Der Digitale Zwilling – Probleme und Lösungsansätze \(degruyter.com\)\]](#)
8. [Was ist ein digitaler Zwilling? | IBM](#)
9. Roßmann J., Schluse M.; Experimentierbare Digitale Zwillinge im Lebenszyklus technischer Systeme, [10.1007/978-3-662-58474-3_43](#)
10. Soori M., Arezoo B., Dastres R.; Digital twin for smart manufacturing: A review, Sustainable Manufacturing and Service Economics, [10.1016/j.smse.2023.100017](#).
11. Stetter R., Grüble T., Till M.; Geometric and kinetic digital twin of a body-in-white assembly system for virtual commissioning, [10.1016/j.procir.2022.12.001](#).
12. <https://www.industrial-production.de/software-en/virtual-commissioning-929742.htm>
13. <https://blog.cadalyst.com/product-design-manufacturing-solutions/digital-twins-come-of-age-for-manufacturing>
14. <https://www.ibm.com/blog/how-digital-twins-optimize-the-performance-of-your-assets-in-a-sustainable-way/>
15. <https://www.fescreen-sim.de/en/sectors-and-ranges-of-application/logistics>



RUB

RUHR-UNIVERSITÄT BOCHUM

EINFÜHRUNGSSCHULUNG DIGITALER ZWILLING

Vielen Dank für Ihre Aufmerksamkeit