



Hochschule Neu-Ulm
University of Applied Sciences

Bachelorarbeit
im Bachelorstudiengang
Wirtschaftsingenieurwesen
an der Hochschule für angewandte Wissenschaften Neu-Ulm

**Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im
industriellen Metaverse am Beispiel der Drohnen GmbH**

Erstkorrektor/-in: Prof. Dr. Jürgen Grinninger

Verfasser/-in: León Kloos (Matrikel-Nr.: 272424)

Thema erhalten: 13.01.2025

Arbeit abgegeben: 12.05.2025

Gliederung:

1. Vorwort.....	1
2. Grundlagen: Digitaler Produktzwilling im industriellen Metaverse.....	3
2.1 Definition und Synergien	3
2.1.1 Industrielles Metaverse	3
2.1.2 Digitaler Produktzwilling.....	3
2.1.3 Synergien.....	4
2.2 Aufbau und Funktionsweise.....	6
2.2.1 Digitaler Produktzwilling.....	6
2.2.2 Industrielles Metaverse	8
2.2.3 Digitaler Produktzwilling im industriellen Metaverse.....	11
3. Technologie	13
3.1 SCARA Roboter.....	13
3.1.1 Mechanischer Aufbau und Bewegungsplanung.....	13
3.1.2 Ablauf der Robotersteuerung	15
3.2 Digitales Modell.....	16
3.2.1 Grundlagen und Ansätze der Modellierung technischer Systeme.....	16
3.2.2 Modellaufbau und Hilfsmittel	17
3.3 IIoT Kommunikationstechnologien	20
3.3.1 Grundlagen der industriellen Kommunikation.....	21
3.3.2 Ablauf industrieller Datenübertragung.....	22
3.4 3D-Integrationsplattform.....	24
3.4.1 Omniverse als Plattform.....	25
3.4.2 Datenstruktur und -verwaltung	25
3.4.2 Datenintegration.....	26

4. Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse	28
4.1 Begriffsdefinition Vorgehensmodell	28
4.2 Anforderungen an das Vorgehensmodell.....	29
4.3 Überblick bestehender Vorgehensmodelle	31
4.3.1 Auswahl und Beschreibung relevanter Modelle	31
4.3.2 Bewertung anhand der Anforderungen	33
4.4 Aufbau und Phasen des Vorgehensmodells.....	35
4.4.1 Zielsetzung und Rahmenbedingungen	35
4.4.2 Aufbau des Vorgehensmodells	35
4.4.3 Phasen des Vorgehensmodells.....	37
5. Anwendung des Vorgehensmodell am Beispiel Drohnen GmbH	44
5.1 Einordnung und Rahmenbedingungen der Anwendung	44
5.2 Anwendung des Vorgehensmodells: Konzeptionierungsphase	45
5.3 Anwendung der Vorgehensmodells: Ausarbeitungsphase.....	48
5.4 Kritische Reflexion	51
6. Fazit und Ausblick	53
Literaturverzeichnis:	54
Abbildungsverzeichnis:.....	60

1. Vorwort

Der Bundesverband der Deutschen Industrie e. V. beschreibt das industrielle Metaverse als eine neue Evolutionsstufe der digitalen Transformation in der Industrie, die signifikant zur Nachhaltigkeit, Wettbewerbsfähigkeit und zur Bewältigung gesamtgesellschaftlicher Herausforderungen beitragen kann (Bundesverband der Deutschen Industrie e. V. 2023). Laut einer Marktprognose, die im MIT Technology Review Insights veröffentlicht wurde, soll der weltweite Umsatz in diesem Bereich bis 2030 von derzeit 15 auf über 100 Milliarden US-Dollar ansteigen (Waxer 2023, S. 6). Dies unterstreicht die technologische und wirtschaftliche Relevanz dieser Entwicklung. Ullrich et al. sehen im industriellen Metaverse einen potenziellen „Game Changer“ für die moderne Industrie. Um dieses Konzept zum Leben zu erwecken, bedarf es einer Vielzahl zusätzlicher Technologien und Innovationen. Eine zentrale Rolle nimmt dabei der digitale Zwilling ein (Ullrich u. a. 2024, S. 186). Das industrielle Metaverse schafft einen vernetzten, immersiven Raum, in dem zahlreiche digitale Zwillinge miteinander interagieren und Mensch, Maschine sowie Daten in Beziehung zueinander treten können (Waxer 2023, S. 13).

Aktuell befindet sich das industrielle Metaverse noch in einer explorativen Phase (Ullrich u. a. 2024, S. 190). Technologische Konzepte werden erprobt, erste Pilotprojekte umgesetzt. Es fehlen jedoch bislang etablierte methodische Rahmenwerke, die die strukturierte Umsetzung konkreter Anwendungsfälle unterstützen, insbesondere im Hinblick auf die Integration digitaler Produktzwillinge. Für die Erstellung digitaler Zwillinge hingegen existieren verschiedene Vorgehensmodelle mit unterschiedlichen Schwerpunkten (Follath/Bross/Galka 2022, S. 691). Somit wird ein umfassender methodischer Rahmen nötig, der nicht nur die Entwicklung digitaler Zwillinge systematisch anleitet, sondern auch dessen Integration in das industrielle Metaverse berücksichtigt. Nur so kann jener vernetzte und immersive Raum entstehen, in dem digitale Zwillinge in Echtzeit miteinander interagieren.

Die vorliegende Arbeit adressiert diesen Bedarf. Ziel ist die Entwicklung eines strukturierten Vorgehensmodells zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse. Dieses Vorgehensmodell soll als methodische Brücke zwischen theoretischen Grundlagen und praktischer Umsetzung dienen. Die Validierung erfolgt exemplarisch anhand eines Anwendungsszenarios bei der Drohnen GmbH, bei dem die Echtzeitvisualisierung eines Roboters im Mittelpunkt steht.

Die Arbeit ist in vier Hauptabschnitte unterteilt. Im ersten Abschnitt werden die zentralen Begriffe sowie der konzeptionelle Aufbau und die Funktionsweise des industriellen Metaverse und des digitalen Produktzwillinges erläutert. Daran anschließend folgt eine technologische Analyse der für die Umsetzung relevanten Komponenten mit einem Fokus auf Robotik, digitale Modellierung, IIoT-Kommunikation und 3D-Integrationsplattformen. Im dritten Abschnitt wird ein Vorgehensmodell zur strukturierten Erstellung eines digitalen Produktzwillinges entwickelt. Abschließend erfolgt die Anwendung dieses Modells im Kontext der Drohnen GmbH, gefolgt von einer kritischen Reflexion der Ergebnisse und einem Ausblick auf weiterführende Potenziale und Entwicklungen.

2. Grundlagen: Digitaler Produktzwilling im industriellen Metaverse

Das industrielle Metaverse und der digitale Zwilling eröffnen Unternehmen entlang der gesamten Wertschöpfungskette neue Potenziale. Diese Technologien fördern eine beschleunigte Produktentwicklung durch digitale Zusammenarbeit, ermöglichen risikofreie Tests in virtuellen Umgebungen und erhöhen die Flexibilität bei der Anpassung an volatile Marktanforderungen (Siemens o. J.). Vor diesem Hintergrund werden in diesem Kapitel die grundlegenden Begriffe, die zentralen Zusammenhänge sowie der Aufbau und die Funktionsweise des industriellen Metaverse und des digitalen Produktzwillings systematisch erläutert.

2.1 Definition und Synergien

Das folgende Kapitel führt in die zentralen Begriffe industrielles Metaverse und digitaler Produktzwillinge ein und zeigt, wie durch ihr Zusammenspiel konkrete Synergien entstehen, die für industrielle Anwendungen von hoher Relevanz sind.

2.1.1 Industrielles Metaverse

Kreutzer und Klose erklären, dass der Begriff "Metaverse" aus dem griechischen Präfix "meta" für "über" und dem Wortteil "verse" oder "versum," das "universe" oder "Universum" bedeutet, hergeleitet wird. Daraus lässt sich ableiten, dass das Metaverse ein über der analogen Welt liegendes Universum darstellt (Kreutzer/Klose 2023, S. 8). Jensen Huang, CEO von NVIDIA, beschreibt das Metaverse als die nächste Evolutionsstufe des Internets, ein dreidimensionales Netzwerk aus verbundenen virtuellen Welten, das den Nutzern ermöglicht, in einer immersiven Umgebung zu interagieren und zu navigieren sowie ein nahezu grenzenloses Zusammenspiel von digitalem und analogem Leben zu erfahren (Caulfield 2021). Obwohl das Metaverse als eine umfassende digitale Welt beschrieben wird, wird es in drei Hauptsektoren unterteilt: das industrielle Metaverse, das Enterprise-Metaverse und das Verbraucher-Metaverse (Waxer 2023, S. 6). Die Zeitschrift für wirtschaftlichen Fabrikbetrieb erklärt, dass das industrielle Metaverse ein virtueller Raum sein wird, der Echtzeit-Simulationen und Visualisierungen von physischen Objekten ermöglicht. Es bietet eine immersive, digitale Umgebung für die Echtzeit-Kollaboration zwischen Menschen sowie zwischen Menschen und physischen Objekten. (Ullrich u. a. 2024, S. 186).

2.1.2 Digitaler Produktzwilling

Der digitale Produktzwilling stellt eine spezifische Ausprägung des übergeordneten Konzepts des digitalen Zwillings dar (Appelfeller/Feldmann 2023, S. 283). Dieses Konzept gilt als zentrales Element im Kontext von Industrie 4.0 (Stiebitz 2022, S. 293). Es hat sich im Laufe der Jahre, gestützt durch unterschiedliche Technologien und

Werkzeuge, kontinuierlich weiterentwickelt und findet in zahlreichen industriellen Bereichen Anwendung. Bei der Suche nach einer umfassenden Definition dieses Konzepts stößt man auf eine Vielzahl unterschiedlicher Interpretationen und Modelle. Ursache hierfür ist die Komplexität und breite Anwendbarkeit (Crespi/Drobot/Minverva 2023, S. 4). Der größte gemeinsame Nenner und somit die Quintessenz zahlreicher Definitionen lässt sich wie folgt zusammenfassen: „Ein digitaler Zwilling ist eine virtuelle Nachbildung eines physischen Produkts, Prozesses oder Systems.“ (Mockenhaupt/Rüttgers 2021, S. 260). Appelfeller und Feldmann unterscheiden digitale Zwillinge in vier Ebenen, basierend auf dem Abbildungsumfang des physischen Objekts. Auf der **Komponentenebene** steht die digitale Repräsentation einzelner Maschinenteile im Fokus. Die **Produktebene** bezieht sich auf die Abbildung eines vollständigen Produkts, beispielsweise eines Roboters. Die **Systemebene** umfasst vernetzte Produktionssysteme, bestehend aus mehreren Maschinen und Fördereinheiten. Die **Prozessebene** schließlich modelliert gesamte Fertigungsabläufe. Der digitale Produktzwilling ist dabei auf der Produktebene verortet. Er stellt eine vielseitig einsetzbare Technologie dar, die branchenübergreifend Anwendung findet (Appelfeller/Feldmann 2023, S. 283–285). Im Rahmen dieser Arbeit steht die Ausgestaltung eines digitalen Produktzwillings im Fokus. In Verbindung mit dem industriellen Metaverse eröffnen sich daraus neue Synergien, die in den folgenden Abschnitten näher erläutert werden.

2.1.3 Synergien

Synergien beschreiben das Zusammenwirken verschiedener Elemente, bei dem die entstehende Gesamtleistung größer ist als die bloße Summe der Einzelleistungen (Maier o. J.). Angerer et al. betonen, dass Synergien im industriellen Umfeld, das zunehmend durch Vernetzung, Echtzeitdaten und komplexe Abläufe geprägt ist, eine zentrale Rolle spielen. Sie helfen dabei, Prozesse effizienter zu gestalten, Innovationen zu fördern und die Wettbewerbsfähigkeit langfristig zu sichern. Unternehmen, die gezielt auf solches Zusammenwirken setzen, können Abläufe nicht nur automatisieren, sondern sie auch grundlegend verbessern. So lassen sich beispielsweise aktuelle Daten zur frühzeitigen Fehlererkennung nutzen und neue Lösungen schneller entwickeln (Angerer u. a. 2023, S. 4–5). Die im weiteren Verlauf dieses Abschnitts beschriebenen Synergien entstehen aus dem Zusammenspiel von digitalem Produktzwilling und industriellem Metaverse. In der Literatur wird häufig allgemein von digitalen Zwillingen gesprochen, da sich die beschriebenen Potenziale auf unterschiedliche Ausprägungen und Anwendungsebenen übertragen lassen.

Nach Waxer et al. bildet der digitale Zwilling das Fundament des industriellen Metaverse. Er stellt ein dynamisches, datenbasiertes Abbild eines realen Objekts dar, das kontinuierlich mit Informationen aus der physischen Welt versorgt wird. Digitale Zwillinge ermöglichen damit sowohl die Überwachung als auch die Simulation industrieller Prozesse in Echtzeit (Waxer 2023, S. 7). Das industrielle Metaverse geht jedoch über diese Funktion hinaus: Es schafft einen vernetzten, immersiven Raum, in dem zahlreiche digitale Zwillinge miteinander interagieren und Mensch, Maschine sowie Daten in Beziehung zueinander treten können (Waxer 2023, S. 13). Angerer et al. heben hervor, dass der digitale Zwilling die strukturierte Datengrundlage liefert, während das industrielle Metaverse diese Daten sichtbar, erlebbar und gemeinschaftlich nutzbar macht. Informationen können dadurch nicht nur analysiert, sondern intuitiv verstanden und in Handlungen übersetzt werden. Der digitale Zwilling liefert die präzise Zustandsabbildung. Das industrielle Metaverse stellt die Interaktions- und Kollaborationsumgebung bereit, etwa durch Visualisierung, Fernzugriff oder synchronisierte Zusammenarbeit in Echtzeit (Angerer u. a. 2023, S. 9–11). Im Folgenden werden drei zentrale Synergiepotenziale näher erläutert.

Zheng et al. sehen ein erstes Potenzial in der erweiterten **Interaktion zwischen Mensch und Maschine**. IngenieurInnen, TechnikerInnen und Produktionsmitarbeitende werden dabei nicht nur beobachtend eingebunden, sondern aktiv in digitale Prozesse integriert. Digitale Avatare ermöglichen eine direkte Interaktion mit virtuellen Modellen und Systemen. Durch die Verbindung menschlichen Erfahrungswissens mit datenbasierten Entscheidungsgrundlagen lassen sich insbesondere in komplexen oder dynamischen Situationen die Prozesssicherheit und Effizienz deutlich steigern (Zheng u. a. 2022, S. 242).

Angerer et al. zeigen, dass ein weiteres Potenzial in der **virtuellen, echtzeitbasierten Entscheidungsunterstützung** liegt. Durch die Kombination von Echtzeitdaten mit unternehmensspezifischem Wissen und intelligenter Datenverarbeitung können simulationsbasierte Entscheidungen getroffen werden. Dadurch werden industrielle Abläufe für Menschen besser nachvollziehbar, was eine schnelle und fundierte Reaktion auf veränderte Rahmenbedingungen ermöglicht (Angerer u. a. 2023, S. 10).

Zheng et al. beschreiben außerdem ein Synergiepotenzial im Bereich der **Zusammenarbeit**. Das industrielle Metaverse ermöglicht eine ortsunabhängige Interaktion über digitale Avatare. Mitarbeitende, KundInnen, externe Dienstleister oder Partnerunternehmen können in einem gemeinsamen virtuellen Raum zusammenarbeiten. Digitale Zwillinge externer Systeme, etwa Maschinenmodelle von Zulieferern, lassen sich direkt integrieren. Dies vereinfacht die Abstimmung,

unterstützt die Auswahl geeigneter Lösungen und fördert eine effiziente, kontextbezogene Kooperation (Zheng u. a. 2022, S. 242).

Die beschriebenen Synergien zeigen, welches Potenzial aus dem Zusammenspiel von digitalem Zwilling und industriellem Metaverse hervorgeht. Sie ermöglichen neue Formen der Interaktion, fundierte Entscheidungen auf Basis aktueller Daten sowie eine engere Zusammenarbeit über System- und Unternehmensgrenzen hinweg.

2.2 Aufbau und Funktionsweise

In diesem Kapitel werden der Aufbau und die Funktionsweise eines digitalen Produktzwillings im industriellen Metaverse dargestellt. Zunächst wird erläutert, welche Komponenten einen digitalen Produktzwilling (vgl. Kapitel 2.2.1) ausmachen und wie diese strukturiert sind. Anschließend wird die Referenzarchitektur des industriellen Metaverse (vgl. Kapitel 2.2.2) vorgestellt, bevor die Zusammenführung (vgl. Kapitel 2.2.3) beider Konzepte in einer übergreifenden Systemstruktur erfolgt.

2.2.1 Digitaler Produktzwilling

Im Folgenden wird der Aufbau und die Funktionsweise eines digitalen Produktzwillings erläutert. Das zugrunde liegende Modell (siehe Abbildung 1) besteht aus drei zentralen Elementen: dem physischen Objekt im realen Raum, dem digitalen Objekt im virtuellen Raum sowie einer dynamischen Verbindung, die den kontinuierlichen Datenaustausch zwischen beiden ermöglicht (Grieves 2015, S. 1).

Das **physische Objekt** beschreibt das reale Produkt mit all seinen Attributen, für das ein digitaler Zwilling erstellt werden soll. Über Sensoren werden Betriebsdaten und Umweltzustände erfasst (Zheng/Yang/Cheng 2019, S. 1143). Diese Sensoren sind über das Internet der Dinge (IoT) vernetzt und ermöglichen eine exakte digitale Abbildung des aktuellen Zustands des physischen Objekts (Kaur/Mishra/Maheshwari 2020, S. 5). Stark et al. erklären, dass die erfassten Daten zunächst im **digitalen Schatten** gespeichert werden. Dabei handelt es sich um eine Art Echtzeit-Datenspeicher, der die aktuellen Informationen über das physische Objekt enthält (Stark/Kind/Neumeyer 2017, S. 170). Zwischen dem physischen Objekt und dem digitalen Schatten besteht ein einseitiger Datenaustausch, bei dem eine Änderung im Zustand des physischen Objekts zu einer Aktualisierung der Informationen im Datenspeicher führt (Kritzinger u. a. 2018, S. 1017–1018). Das **digitale Modell** spiegelt nun das physische Objekt virtuell wider und beinhaltet Geometrie-, Physik-, Regel- und Verhaltensmodelle (Tao/Zhang/Nee 2019, S. 70). Kritzinger betont, dass zwischen dem digitalen Modell und dem physischen Objekt kein automatischer Datenaustausch stattfindet, weshalb Änderungen am physischen Objekt keine direkte Auswirkung auf das digitale Modell haben. Der Datenaustausch erfolgt manuell (Kritzinger u. a. 2018, S. 1017–1018). Tao,

Zhang und Nee erläutern, dass der eigentliche **digitale Produktzwilling** erst durch die Verknüpfung von digitalem Schatten und digitalem Modell entsteht. Dabei wird das digitale Modell mithilfe der im digitalen Schatten gespeicherten Daten aktualisiert, was Visualisierungen und Simulationen ermöglicht. Darüber hinaus können auch virtuelle Änderungen am digitalen Modell auf das physische Objekt zurückwirken. So lassen sich reale Zustände gezielt beeinflussen (Tao/Zhang/Nee 2019, S. 70–71). In diesem Fall liegt ein bidirektionaler Datenfluss vor, der die höchste Entwicklungsstufe eines digitalen Produktzwillings kennzeichnet (Kritzinger u. a. 2018, S. 1017–1018).

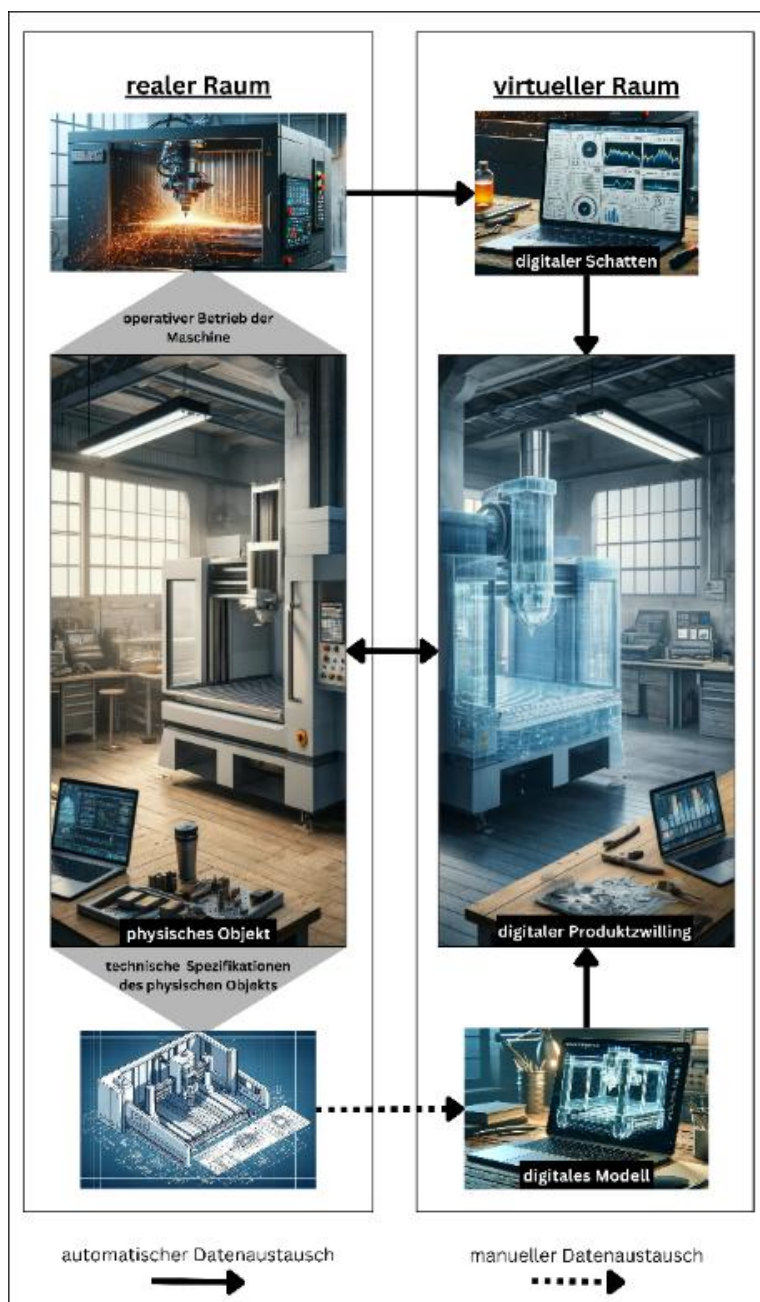


Abbildung 1: Modell des digitalen Produktzwilling

Quelle: Eigene Darstellung unter Verwendung von generierten Bildern durch OpenAI

2.2.2 Industrielles Metaverse

In der Zeitschrift für wirtschaftlichen Fabrikbetrieb wird eine Referenzarchitektur vorgestellt, die als Ausgangspunkt für die Entwicklung spezifischer Architekturen des industriellen Metaverse dienen kann. Sie vereint zentrale Technologien, die für den Aufbau eines solchen Systems erforderlich sind. Die von Ullrich et al. entwickelte Referenzarchitektur setzt sich aus drei Säulen zusammen, die zu einer zentralen Plattform gebündelt werden (siehe Abbildung 2) (Ullrich u. a. 2024, S. 187).

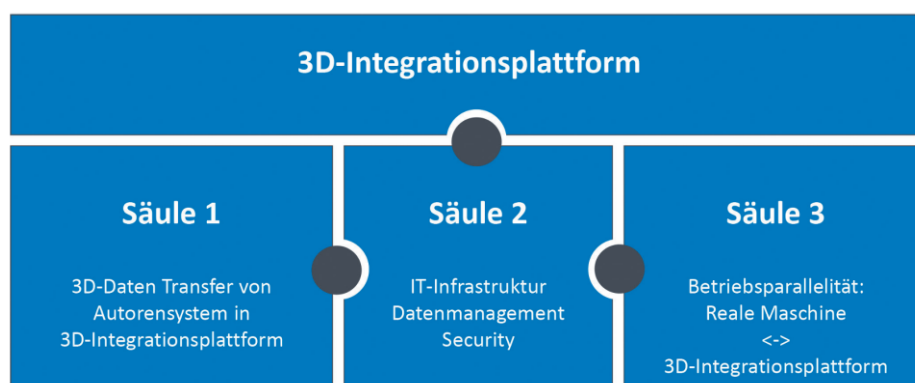


Abbildung 2: Referenzarchitektur eines industriellen Metaverse

Quelle: Ullrich u. a. 2024, S. 187

Ullrich et al. beschreiben, dass sich die **erste Säule** (vgl. Abbildung 2) der Referenzarchitektur mit dem Transfer von 3D-Daten aus sogenannten Autorensystemen in die 3D-Integrationsplattform befasst. Ziel ist es, sicherzustellen, dass alle relevanten Informationen wie Geometrie, Physik, Regelstrukturen und Verhaltensmodelle vollständig und korrekt übertragen werden. Diese Daten bilden die Grundlage für das digitale Modell des betrachteten digitalen Produktzwillings. Autorensysteme sind spezialisierte Softwareumgebungen, die zur Erstellung, Modellierung und Simulation digitaler Objekte dienen. Für die Integration dieser Daten in die 3D-Integrationsplattform werden **vier grundlegende Ansätze** unterschieden (siehe Abbildung 3). Im **ersten Ansatz** erfolgt eine statische Umwandlung der 3D-Daten in ein standardisiertes Format, das anschließend in die Plattform importiert wird. Der **zweite Ansatz** basiert auf einer unidirektionalen Live-Synchronisation. Dabei werden Änderungen, die im digitalen Modell des Autorensystems vorgenommen werden, automatisch an das Modell innerhalb der 3D-Integrationsplattform übermittelt. Im **dritten Ansatz** erfolgt eine bidirektionale Live-Synchronisation. Zusätzlich zur Übertragung aus dem Autorensystem in die Plattform werden auch Änderungen, die innerhalb der Plattform vorgenommen werden, zurück an das Ursprungsmodell im Autorensystem übermittelt. Bei beiden Synchronisationsverfahren muss das Autorensystem aktiv im Hintergrund laufen. Der

vierte Ansatz sieht vor, dass Simulationsabläufe direkt im Autorensystem aufgezeichnet und anschließend in ein standardisiertes Datenformat konvertiert werden. Diese sogenannten Aufzeichnungsdateien, auch „Recorded Files“ genannt, können innerhalb der Plattform ereignisbasiert abgespielt werden, ohne dass eine aktive Verbindung zur Simulationssoftware erforderlich ist. Eine durchgängig etablierte Architektur zur Umsetzung dieser Integrationsformen liegt bislang nicht vor. Der aktuelle Stand entspricht verschiedenen konzeptionellen Referenzansätzen, die sich noch in der Entwicklung befinden und kontinuierlich weiter erforscht werden. Im Rahmen dieser Arbeit wird die Anbindung von Autorensystemen an die 3D-Integrationsplattform exemplarisch anhand eines sogenannten Konnektors beschrieben, wie er beispielsweise im Kontext von NVIDIA Omniverse zum Einsatz kommt (Ullrich u. a. 2024, S. 187–188).

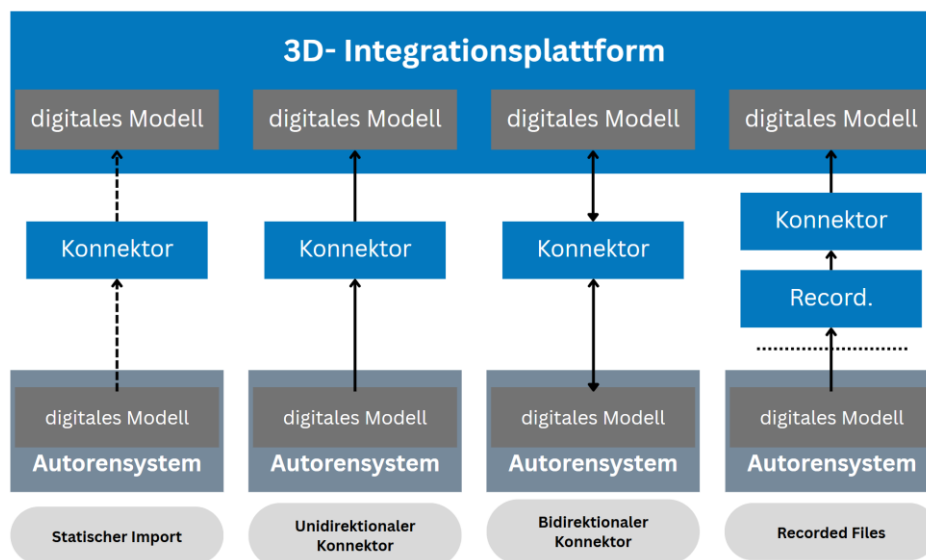


Abbildung 3: Möglichkeiten der Integration von digitalen Modellen in die 3D-Integrationsplattform

Quelle: in Anlehnung Ullrich u. a. 2024, 188

Ullrich et al. beschreiben die **zweite Säule** (vgl. Abbildung 2) der Referenzarchitektur als das technische Fundament des industriellen Metaverse. Sie umfasst zentrale Basiselemente wie die IT-Infrastruktur, das Datenmanagement und Sicherheitsmechanismen. Ziel ist es, eine Umgebung bereitzustellen, die sowohl die physischen Produktionssysteme als auch die digitalen Zwillinge zuverlässig miteinander verbindet (Ullrich u. a. 2024, S. 188). Diese infrastrukturelle Basis stellt sicher, dass Daten aus der realen Welt konsistent erfasst, verarbeitet und den digitalen Repräsentationen zur Verfügung gestellt werden können. Sie schafft die Voraussetzungen dafür, dass digitale Zwillinge in Echtzeit mit aktuellen Informationen versorgt und virtuelle Anwendungen stabil betrieben werden. Darüber hinaus spielt

diese Säule eine wichtige Rolle für die Koordination von Anwendungen innerhalb des Metaverse. Dazu gehören beispielsweise die Steuerung von Datenflüssen, die Verwaltung von Zugriffsrechten oder die Integration externer Systeme. Insgesamt bildet die zweite Säule das Rückgrat der Plattform. Ohne eine zuverlässige, leistungsfähige und sichere Infrastruktur ist der durchgängige Betrieb eines industriellen Metaverse mit Echtzeitfähigkeit, Datenkonsistenz und Interaktion nicht möglich (Ullrich u. a. 2024, S. 188–189).

Ullrich et al. beschreiben die **dritte Säule** (vgl. Abbildung 2) der Referenzarchitektur als sogenannte Betriebsparallelität. Sie bezeichnet die Echtzeitverbindung zwischen einem physischen Objekt, wie etwa einem Roboter, und dessen digitalem Modell innerhalb der 3D-Integrationsplattform. Ziel dieser Kopplung ist es, den laufenden Betrieb des realen Systems vollständig und synchron im virtuellen Raum abzubilden. Erst durch diese Verbindung wird es möglich, das Verhalten physischer Systeme nicht nur zu überwachen, sondern auch zu analysieren, vorherzusagen und bei entsprechender Ausbaustufe gezielt zu beeinflussen (Ullrich u. a. 2024, S. 189). Matysczok und Hohlweger zeigen, dass die Art der Betriebsparallelität maßgeblich den Funktionsumfang eines digitalen Produktzwilling bestimmt. Abhängig von der Nutzung der Daten und der Richtung des Informationsflusses lassen sich drei Entwicklungsstufen unterscheiden (siehe Abbildung 4). Auf der grundlegendsten Stufe befindet sich der sogenannte **Datenzwilling**. Dieser erfasst Betriebsdaten wie Temperatur-, Zustands- oder Positionswerte in Echtzeit und stellt sie im digitalen Modell strukturiert bereit. Der Informationsfluss verläuft dabei ausschließlich vom realen Objekt zur digitalen Umgebung, weshalb man von unidirektionaler Betriebsparallelität spricht. Eine Rückwirkung auf das physische System findet nicht statt. Der **Simulationszwilling** erweitert diese Funktionalität um analytische Komponenten. Die erfassten Echtzeitdaten werden nicht nur dokumentiert oder visualisiert, sondern auch genutzt, um das Systemverhalten zu simulieren und Vorhersagen über zukünftige Zustände zu treffen. Auch in dieser Variante bleibt der Informationsfluss einseitig, eine direkte Beeinflussung des realen Objekts erfolgt nicht. Die höchste Entwicklungsstufe stellt der sogenannte **Steuerungszwilling** dar. Hier findet eine bidirektionale Kommunikation statt. Neben der kontinuierlichen Übertragung von Betriebsdaten kann das digitale Modell auch aktiv auf das physische Objekt einwirken, zum Beispiel durch das Senden von Steuerbefehlen in Echtzeit. Es entsteht eine dynamische Wechselwirkung zwischen realem und virtuellem Raum, wodurch digitale Entscheidungen unmittelbar in der realen Welt umgesetzt werden können (Matysczok/Hohlweger 2021, S. 12). Ohne Betriebsparallelität handelt es sich

hingegen um eine klassische **Simulation**, bei der das virtuelle Modell vollständig losgelöst vom realen Objekt arbeitet (Matysczok/Hohlweber 2021, S. 7).

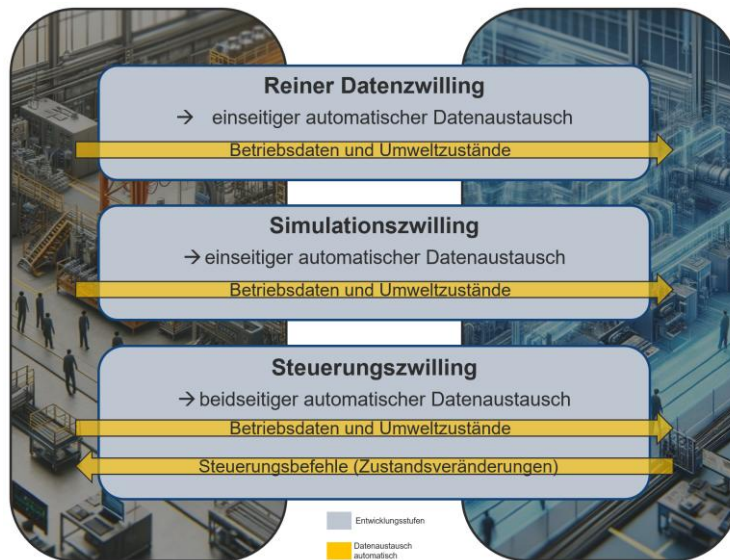


Abbildung 4: Unterscheidung digitaler Produktzwilling anhand der Art ihrer Anwendung

Quelle: Eigene Darstellung unter Verwendung von generierten Bildern durch OpenAI

Die **zentrale Plattform** (vgl. Abbildung 2), die 3D-Integrationsplattform, bündelt alle Daten und Prozesse, die durch die Säulen bereitgestellt werden und bietet eine zentrale Umgebung für die Visualisierung und Interaktion (Ullrich u. a. 2024, S. 187).

2.2.3 Digitaler Produktzwilling im industriellen Metaverse

Die zuvor getrennt betrachteten Konzepte des digitalen Produktzwillings (siehe Kapitel 2.2.1) und der Referenzarchitektur des industriellen Metaverse (siehe Kapitel 2.2.2) werden in diesem Abschnitt zusammengeführt. Die Abbildung 5 veranschaulicht die daraus resultierende Systemstruktur. Sie zeigt, wie physische und digitale Elemente über eine zentrale 3D-Integrationsplattform miteinander verbunden sind.

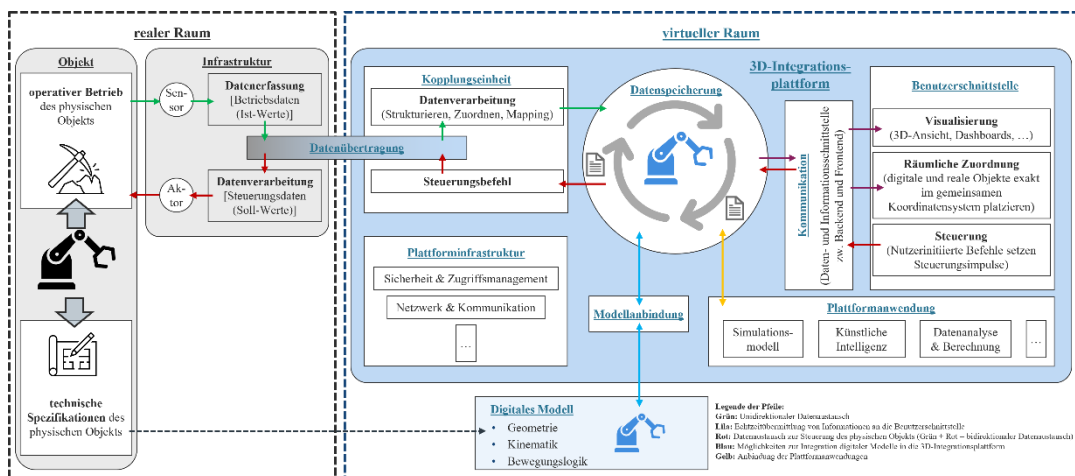


Abbildung 5: Systemstruktur - Digitaler Produktzwilling im industriellen Metaverse

Quelle: in Anlehnung Tu u. a. 2024, S. 968

Der **reale Raum** umfasst das physische Objekt, das mithilfe von Sensorik kontinuierlich überwacht wird. Die dabei erfassten Daten bilden die Grundlage für den digitalen Schatten und werden über eine Kopplungseinheit in den virtuellen Raum übertragen (Tu u. a. 2024, S. 969). Diese Kopplungseinheit entspricht sowohl der beschriebenen dynamischen Verbindung (vgl. Kapitel 2.2.1) als auch der dritten Säule der Referenzarchitektur (vgl. Kapitel 2.2.2), da sie die Echtzeitverbindung zwischen beiden Welten herstellt. Im **virtuellen Raum** befindet sich das digitale Modell. Es repräsentiert Geometrie, Verhalten und Logik des realen Objekts (vgl. Kapitel 2.2.1). Die Anbindung an die Plattform erfolgt über die Modellanbindung, die der ersten Säule der Referenzarchitektur entspricht (vgl. Kapitel 2.2.2). Dabei können verschiedene Daten aus Autorensystemen in standardisierter Form eingebunden werden. Die zweite Säule der Referenzarchitektur (vgl. Kapitel 2.2.2), die sich mit der technischen Infrastruktur, dem Datenmanagement und der Sicherheit befasst, findet ihre Umsetzung in der Plattforminfrastruktur und Datenspeicherung. Im Zentrum steht die 3D-Integrationsplattform. Sie vereint die Modellanbindung (Säule 1), die Echtzeitverbindung über die Kopplungseinheit (Säule 3), die technische Infrastruktur (Säule 2) sowie weitere zentrale Elemente wie Benutzerschnittstellen und Plattformanwendungen. Laut Tu et al. ermöglicht diese Umgebung die Nutzung des digitalen Produktzwillings in Echtzeit, sowohl zur Visualisierung als auch zur Interaktion zwischen Mensch und System (Tu u. a. 2024, S. 969–970).

Aufbauend auf dieser Systemstruktur wird im Folgenden die **Interaktion** zwischen physischer und digitaler Welt beschrieben. Wie in Kapitel 2.1.2 erläutert, liegt der Fokus dieser Arbeit auf einem digitalen Produktzwilling in Form eines reinen Datenzwillings. Es findet keine Rückkopplung zum realen Objekt statt. Im betrachteten Szenario handelt es sich beim physischen Objekt um ein Produkt, dessen Betriebs- und Zustandsdaten, wie von Tu et al. beschrieben, kontinuierlich durch Sensorik erfasst werden. Diese Daten werden über die Kopplungseinheit an die Integrationsplattform übermittelt, dort verarbeitet und gespeichert (siehe grüne Pfeile in Abbildung 5). Das digitale Modell innerhalb der Plattform wird kontinuierlich mit diesen Daten synchronisiert und bildet so den aktuellen Zustand des Produkts ab. Die Plattform stellt sicher, dass die Informationen in Echtzeit an die Benutzerschnittstelle übermittelt und dort entsprechend visualisiert werden können (siehe lila Pfeile in Abbildung 5) (Tu u. a. 2024, S. 971–972). Auf diese Weise entsteht ein digitaler Produktzwilling, der als Datenzwilling konzipiert ist.

3. Technologie

Dieses Kapitel erläutert die Technologien, die für die Erstellung eines digitalen Produktzwillings im industriellen Metaverse erforderlich sind. Um eine inhaltliche Kohärenz mit der späteren Anwendung in Kapitel 5 sicherzustellen, werden gezielt jene Technologien behandelt, die auch im praktischen Anwendungsbeispiel zur Anwendung kommen.

3.1 SCARA Roboter

In diesem Kapitel wird der SCARA-Roboter als zentrales Element des betrachteten digitalen Produktzwillings im industriellen Metaverse eingehend analysiert. Der Begriff SCARA steht für **S**elective **C**ompliance **A**ssembly **R**obot **A**rm und bezeichnet einen Industrieroboter mit vier Freiheitsgraden: Bewegungen entlang der X-, Y- und Z-Achse sowie eine Rotation um die Z-Achse (Chen u. a. 2012, S. 171). Diese kinematische Struktur ermöglicht eine hohe Dynamik und Wiederholgenauigkeit, wodurch sich SCARA-Roboter besonders für Montage- und Bestückungsaufgaben eignen, bei denen Präzision und Geschwindigkeit entscheidend sind (Shariatee u. a. 2014, S. 534). Seit seiner Entwicklung durch Professor Hiroshi Makino im Jahr 1979 hat sich der SCARA-Roboter als etablierte Lösung in automatisierten Fertigungsprozessen durchgesetzt (Barbosa u. a. 2020, S. 1570).

Um ein umfassendes Verständnis für Aufbau und Funktionsweise des eingesetzten Roboters zu ermöglichen, gliedert sich dieses Kapitel in zwei Abschnitte. Abschnitt 3.1.1 behandelt den mechanischen Aufbau sowie die zugrunde liegende Bewegungsplanung. Abschnitt 3.1.3 widmet sich anschließend dem Ablauf der Robotersteuerung. Beide Abschnitte beziehen sich konkret auf das Modell Omron i4L-550L, das im weiteren Verlauf als Grundlage für die Umsetzung des digitalen Produktzwillings dient.

3.1.1 Mechanischer Aufbau und Bewegungsplanung

Der SCARA-Roboter i4-550L von Omron zeichnet sich durch einen klar strukturierten **mechanischen Aufbau** aus, der auf mehrere funktionale Hauptkomponenten verteilt ist (siehe Abbildung 6). Die Basis bildet das Fundament und dient zur festen Montage des Roboters. Daran schließen sich zwei starre Armglieder an, der Inner Link und der Outer Link, welche gemeinsam den beweglichen Roboterarm in der Horizontalen formen (Omron 2023a, Abs. 1–4). Vertikal ist der sogenannte Quill angebracht, ein zylinderförmiges Element, das sowohl für die lineare Bewegung als auch für die Rotation um die Z-Achse verantwortlich ist. Am unteren Ende des Quills befindet sich der Tool Flange, der als Schnittstelle für Werkzeuge oder Greifer dient. Zur sicheren

Kabelführung zwischen Basis und Armstruktur wird der Flyover eingesetzt (Omron 2023a, S. 27). Die Bewegung der mechanischen Komponenten erfolgt über vier Gelenke (in Englisch Joints), die dem Roboter insgesamt vier Freiheitsgrade verleihen (siehe Abbildung 6). Joint 1 erlaubt die Rotation des Inner Links um die Basis, während Joint 2 die Rotation des Outer Links um den Inner Link ermöglicht. Joint 3 führt eine lineare Bewegung entlang der Z-Achse durch, und Joint 4 dreht ebenfalls um die Z-Achse (Omron 2023a, Abs. 1–5). Die Gelenke werden durch integrierte Servomotoren angetrieben, die innerhalb der Roboterstruktur verbaut sind und hochpräzise rotatorische bzw. translatorische Bewegungen ausführen.

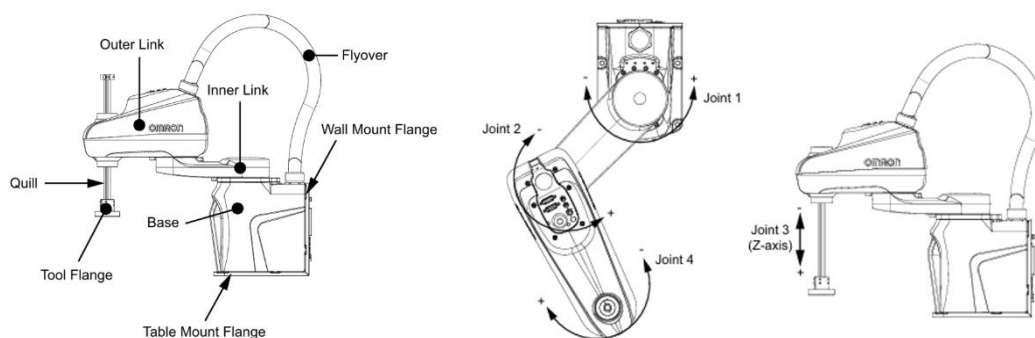


Abbildung 6: Mechanischer Aufbau und Freiheitsgrade

Quelle: Omron 2023a, Abs. 1-4

Laut Chen et al. beginnt die **Bewegungsplanung** des Roboters mit der Definition eines Zielpunkts im Raum, meist in Form kartesischer Koordinaten (X, Y, Z). Dieser Zielpunkt stellt die angestrebte Endposition beispielsweise eines Greifers dar. Um diese zu erreichen, muss das Robotersystem berechnen, welche Gelenkstellungen erforderlich sind. Dies geschieht mithilfe der inversen Kinematik, bei der die Koordinaten in Gelenkwerte umgerechnet werden (Chen u. a. 2012, S. 172–173). Horne et al. erklären, dass die Bewegungsplanung jedoch mehr als nur die Bestimmung der Zielposition umfasst, sie beinhaltet insbesondere auch die sogenannte Trajektorienplanung. Diese legt fest, wie sich die Gelenkparameter über die Zeit hinweg verändern müssen, um von der Start- zur Zielposition zu gelangen. Ziel ist eine kollisionsfreie und effiziente Bewegungsausführung, die zudem möglichst sanft und kontinuierlich verläuft (Horne/Jamshidi/Vadiee 1990, S. 58). Chen et al. zeigen, dass sich der Ablauf einer Trajektorienplanung in mehrere Schritte gliedert. Zunächst wird der Start- und der Zielpunkt definiert und auf Gültigkeit geprüft. Anschließend werden Bewegungsparameter wie Geschwindigkeit und Beschleunigung festgelegt. Um komplexe Bewegungen zu ermöglichen, können zusätzlich Zwischenpunkte entlang der gewünschten Bahn bestimmt werden. Für jeden dieser Punkte wird die inverse Kinematik angewendet, um die erforderlichen Gelenkstellungen zu berechnen. Da es

für eine Zielposition häufig mehrere mögliche Gelenkkonfigurationen gibt, wird anhand zuvor definierter Kriterien eine optimale Lösung ausgewählt. Dieser Vorgang wird für alle Trajektorienpunkte wiederholt, bis schließlich eine vollständige Bewegungslösung vorliegt (Chen u. a. 2012, S. 175). Zur Überwachung der Bewegung nutzt der Roboter die Vorwärtskinematik, wodurch die aktuellen Gelenkwerte und die Position und Orientierung des Greifers berechnet werden kann (Horne/Jamshidi/Vadiee 1990, S. 57).

3.1.2 Ablauf der Robotersteuerung

In Abbildung 7 (auf Seite 16) ist der vereinfachte Aufbau einer Pick-and-Place-Zelle dargestellt, der veranschaulicht, welche Komponenten am Steuerungsablauf des Roboters beteiligt sind. Die Steuerung erfolgt in dieser Anwendung über die Softwareumgebung ACE (Automation Control Environment) von Omron, die direkt auf einem industriellen PC (IPC Application Controller) ausgeführt wird (Omron 2020, Abs. 1–1). Zunächst werden die zu bearbeitenden Teile über einen Hopper auf die Bereitstellfläche des iPF Part Feeders gegeben. Eine über der Fläche angebrachte Vision-Kamera erfasst kontinuierlich die Position und Orientierung der einzelnen Teile. Die Bilddaten werden in Echtzeit an die ACE-Software übermittelt (Omron 2023b, Abs. 1–1). Dort verarbeitet die Software die eingehenden Informationen, identifiziert geeignete Teile und berechnet auf Basis der Kameradaten die optimale Greifposition sowie die zugehörige Bewegungstrajektorie des Roboters. Dabei berücksichtigt sie Parameter wie Geschwindigkeit und Beschleunigung, um eine kollisionsfreie und prozessgerechte Bewegung sicherzustellen. Die berechneten Bewegungsbefehle werden anschließend vom IPC Application Controller direkt an den Industrieroboter übertragen. Dieser führt die geplante Bewegung aus, positioniert sich über dem Zielteil, greift es und legt es an einer definierten Stelle ab. Zur Einrichtung, Simulation und Anpassung von Bewegungsprogrammen (basieren auf der Programmiersprache V+) kann zusätzlich ein Entwicklungs-PC mit der ACE-Software eingesetzt werden (Omron 2020, Abs. 1–1).

Der SCARA-Roboter von Omron verfügt über mehrere Kommunikationsschnittstellen. Für umfangreiche Datenflüsse kommen Netzwerkprotokolle wie TCP/IP zum Einsatz. Diese ermöglichen die Übertragung von Steuerbefehlen sowie den Austausch von Zustands- und Bewegungsdaten (Omron 2020, Abs. 2-1–3).

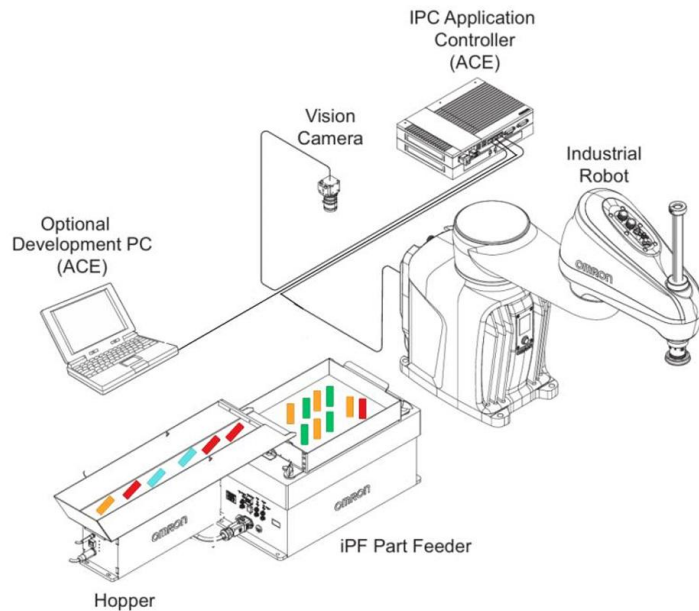


Abbildung 7: vereinfachter Aufbau einer Pick-and-Place-Zelle

Quelle: in Anlehnung Omron 2020, Abs. 1-1

3.2 Digitales Modell

Wie in Kapitel 2.2.1 beschrieben, stellt das digitale Modell das physische Objekt im virtuellen Raum dar. Es bildet Geometrie, Logik und Verhalten des realen Systems ab. In diesem Kapitel werden zunächst grundlegende Ansätze zur Modellierung technischer Systeme erläutert. Darauf aufbauend wird die Vorgehensweise zum Aufbau des Modells sowie der Einsatz geeigneter Hilfsmittel beschrieben.

3.2.1 Grundlagen und Ansätze der Modellierung technischer Systeme

Nach Zauner und Schrempf bildet die Modellierung technischer Systeme die Grundlage für deren digitale Repräsentation. Eine zentrale theoretische Basis hierfür liefert die **Systemtheorie**, die das Verständnis und die strukturierte Beschreibung komplexer technischer Zusammenhänge ermöglicht. Ein System wird dabei als eine Menge interagierender Komponenten definiert, die über Signale Informationen austauschen und so den Gesamtzustand des Systems definieren. Eine wesentliche Voraussetzung für die Modellbildung ist die klare Abgrenzung zur Umgebung, die durch eine definierte **Systemgrenze** erfolgt. Nur Elemente, die innerhalb dieser Grenze liegen, werden bei der Modellbildung berücksichtigt (Zauner/Schrempf 2009, S. 224). Ein weiterer entscheidender Aspekt ist der **zeitliche Bezug**. Ulbrich betont, dass bei Echtzeitsystemen nicht nur die funktionale Korrektheit, sondern auch die Reaktionszeit innerhalb definierter Fristen entscheidend ist. Dabei wird zwischen weichen, festen und harten Echtzeitanforderungen unterschieden. Während bei weichen Echtzeitsystemen Fristverletzungen tolerierbar sind, führen solche Überschreitungen bei harten Echtzeitanforderungen unweigerlich zum

Systemversagen. Feste Echtzeitsysteme bilden eine Zwischenform, hier sind verspätete Ergebnisse zwar unbrauchbar, jedoch nicht sicherheitskritisch. Die tolerierbare Reaktionszeit ist somit stets abhängig von den jeweiligen Systemanforderungen (Ulbrich 2017, S. 10–14).

Auf Basis dieser theoretischen Grundlagen beschreiben Zauner und Schrempf, **Modelle** als zweckgerichtete, vereinfachte Abbildungen der Realität. Sie konzentrieren sich auf diejenigen Eigenschaften eines Systems, die für die jeweilige Fragestellung relevant sind (Zauner/Schrempf 2009, S. 224–225). Ansätze für die **Modellierung** von Modellen unterscheiden sich besonders im Ausmaß des verfügbaren Systemwissens. Buchholz zeigt, dass sich die Modellbildung je nach Grad der Transparenz in Bezug auf die Systemstruktur in drei grundlegende Kategorien gliedern lässt. Der **White-Box-Ansatz** setzt vollständige Kenntnis über den inneren Aufbau und die Funktionsweise des Systems voraus. Der **Black-Box-Ansatz** hingegen verzichtet auf Einblicke in die interne Struktur. Stattdessen wird das Verhalten des Systems, allein auf Basis von Input-Output-Beziehungen analysiert (Buchholz 2006, S. 7). In der Praxis dominiert häufig der Grey-Box-Ansatz, der beide Sichtweisen kombiniert (Ehmer/Khan 2012, S. 14). Wie Langlotz et al. verdeutlichen, spiegeln sich diese theoretischen Modellierungsansätze in der Praxis insbesondere in zwei Formen wider: der **physikalischen** und der **datengetriebenen Modellierung**. Die physikalische Modellierung folgt dem White-Box-Prinzip, basiert auf einem fundierten Systemverständnis und verwendet mathematisch-physikalische Gleichungen, um das Systemverhalten zu beschreiben. Neben dem mechanischen Aufbau eines Systems werden hierbei auch kinematische Zusammenhänge und die Bewegungslogik berücksichtigt. Die datengetriebene Modellierung hingegen entspricht dem Black Box-Ansatz. Sie nutzt Methoden des maschinellen Lernens, um das Verhalten eines Systems ausschließlich auf Basis historischer oder aktueller Daten zu erschließen (Langlotz u. a. 2020, S. 340–341). Die Wahl der geeigneten Methode richtet sich nach Zielsetzung, Datenverfügbarkeit und Systemkenntnis.

Im Rahmen dieser Arbeit wird die physikalische Modellierung als methodischer Schwerpunkt verfolgt. Das digitale Modell wird mit importierten Geometrien, kinematischer Strukturen und bewegungsbasierter Logik realisiert. Der Fokus liegt somit auf einem White-Box-Ansatz zur präzisen und nachvollziehbaren Abbildung des realen Systems.

3.2.2 Modellaufbau und Hilfsmittel

Der Aufbau eines digitalen Modells folgt einem strukturierten, phasenorientierten Vorgehen, das sich an etablierten Modellbildungsmethoden orientiert. Ziel ist es, das

reale System, in diesem Fall einen SCARA-Roboter, in ein digitales Modell zu überführen. Gutenschwager et al. erklären, dass sich die Modellbildung in die **drei aufeinander aufbauenden Phasen** Systemanalyse, Formalisierung und Implementierung gliedert.

Ausgangspunkt der Modellbildung ist die **Systemanalyse**, in der die relevanten Strukturen, Funktionen und Grenzen des physischen Systems bestimmt werden (Gutenschwager u. a. 2017, S. 150–151). Diese Phase dient der Abgrenzung zur Umgebung, der Auswahl der modellrelevanten Komponenten sowie der Festlegung des erforderlichen Detaillierungsgrads (vgl. Kapitel 3.2.2). Als Ergebnis der Systemanalyse entsteht ein Konzeptmodell, das Systemgrenzen, Komponentenstruktur, Interaktionen sowie erste logische Abläufe dokumentiert (Gutenschwager u. a. 2017, S. 151).

Die zweite Phase umfasst laut Gutenschwager et al. die **Formalisierung** des zuvor erarbeiteten Konzepts. Ziel ist die Erstellung eines eindeutigen, widerspruchsfreien Modells (Gutenschwager u. a. 2017, S. 155). Die Erstellung des Modells erfolgt in der Software Visual Components. Dazu werden die CAD-Daten des SCARA-Roboters in die Software importiert und ein geometrisches Modell erstellt. Die Komponenten werden hierarchisch organisiert und im Anschluss werden auf Basis von Herstellinformationen, die kinematischen Bewegungsarten der Gelenke sowie ihre mechanischen Grenzen definiert (Visual Components o. J.). Diese Phase stellt eine wichtige Grundlage für die spätere Ausführung und Interaktion dar, da sie die strukturellen und mechanischen Rahmenbedingungen für alle logischen Abläufe schafft.

Die abschließende Phase ist die **Implementierung**. Die Umsetzung roboterbasierter Bewegungsabläufe im digitalen Modell basiert auf einer strukturierten Herangehensweise, die sich funktional in die Bewegungsplanung, -steuerung und -ausführung unterteilen lässt (Fang u. a. 2019, S. 127–128). In Visual Components werden diese Ebenen durch ein Zusammenspiel von Python-Skript, Value-Signalen, Controllern, Behaviors und der vordefinierten Kinematik realisiert (Visual Components 2018). Die **Bewegungsplanung** beschreibt den mathematischen und logischen Prozess, in dem festgelegt wird, wie sich ein Gelenk über die Zeit hinweg von einer Start- zu einer Zielposition bewegen soll (Fang u. a. 2019, S. 128). Diese erfolgt im Python-Skript mit dem Ziel, eine kontinuierliche, ruckfreie und realitätsnahe Bewegung zu erzeugen (Visual Components 2018). In dieser Arbeit erfolgt die Bewegungsplanung auf Grundlage von den eingelesenen Trajektorien Daten des physischen Objekts. Das sind Daten, welche den Verlauf einer Bewegung über die

Zeit hinweg beschreiben (König 2024). Vom physischen Objekt erhalten wir die Startpositionen der vier Gelenke (Joint 1–4), die Zielpositionen der vier Gelenke sowie die sieben Rampenzeiten, welche jeweils die Dauer einer Bewegungsphase innerhalb der Trajektorie definieren. Nach Fang et al. kommt statt einer linearen Bewegung eine sogenannte 7-Phasen-S-Kurve zum Einsatz. Diese besteht aus sieben aufeinanderfolgenden Bewegungsabschnitten, die für einen sanften Anstieg und Abfall von Beschleunigung und Geschwindigkeit sorgen. Die Rampenzeiten bestimmen dabei die zeitliche Länge jeder Phase und steuern so maßgeblich die Dynamik der Bewegung. Die folgende Abbildung 8 veranschaulicht den Bewegungsverlauf einer solchen S-Kurve anhand eines typischen Trajektorienprofils. Sie zeigt, wie sich eine Bewegung physikalisch sauber und gleichmäßig von einem Startpunkt zu einem Zielpunkt entwickeln kann, genau das, was in der Bewegungsplanung angestrebt wird (Fang u. a. 2019, S. 129).

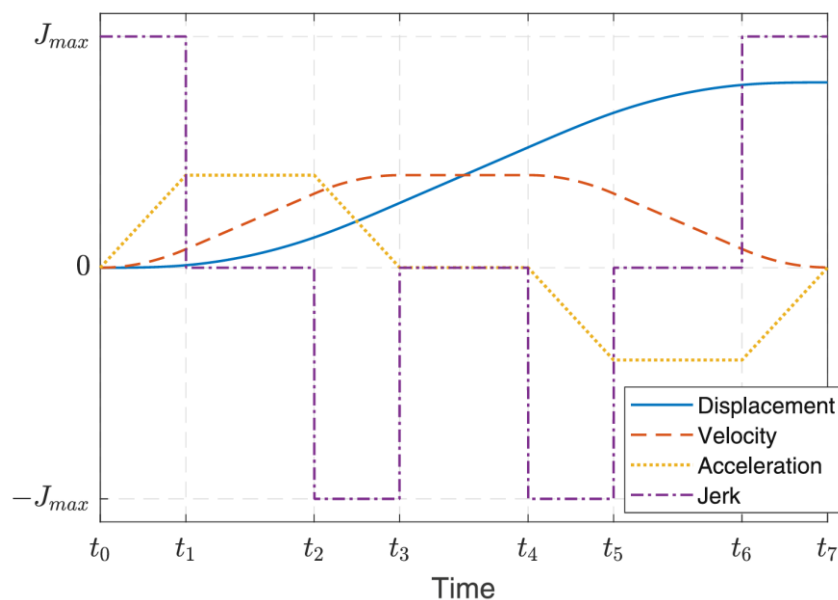


Abbildung 8: Bewegungsverlauf einer S-Kurve

Quelle: Fang u. a. 2019, S. 130

Gemäß Fang et al. repräsentieren die vier dargestellten Kurven (vgl. Abbildung 8) zentrale physikalische Größen, die sich im Verlauf der Bewegung über die Zeit verändern. Der sogenannte **Ruck** (Jerk), dargestellt durch die lila gestrichelte Kurve, beschreibt die zeitliche Veränderung der Beschleunigung. Er ist abschnittsweise konstant und prägt den grundsätzlichen Charakter der Bewegung. Die orange gepunktete Kurve stellt die **Beschleunigung** (Acceleration) dar, die zunächst ansteigt, anschließend für eine gewisse Zeit konstant bleibt und schließlich wieder abfällt, wodurch die Bewegung sanft beginnt und endet. Die rot gestrichelte Kurve repräsentiert die **Geschwindigkeit** (Velocity), die kontrolliert zunimmt, im mittleren

Abschnitt gleichmäßig verläuft und zum Ende hin wieder abnimmt. Die **Positionskurve** (Displacement), dargestellt in Blau, zeigt den stetigen Bewegungsfortschritt des Gelenks und verläuft dabei kontinuierlich ohne abrupte Sprünge. Zwischen den Zeitpunkten t_0 bis t_7 erkennt man die sieben Bewegungsphasen, die exakt den sieben Rampenzeiten entsprechen. Diese Zeiten geben vor, wie lange die jeweiligen Abschnitte, wie zum Beispiel die Beschleunigungs-, konstante Geschwindigkeits- oder Bremsabschnitte der Bewegung dauern. Die **blaue Kurve** (S-Kurve) in der Abbildung 8 zeigt den Verlauf der Gelenkposition über die Zeit. Sie stellt die geplante Bewegung von der Start- bis zur Zielposition dar (Fang u. a. 2019, S. 129–130). Um diese Bewegung im digitalen Modell umsetzen zu können, wird sie nach Fang et al. durch Interpolation in viele einzelne Punkte zerlegt. Jeder dieser Trajektorienpunkte gibt an, welchen Positionswert das Gelenk zu einem bestimmten Zeitpunkt einnehmen soll. Durch die zeitliche Auflösung entsteht eine Punktfolge entlang der Kurve, die die Bewegung in kleinen Schritten beschreibt. Je enger die Punkte beieinander liegen, also je kürzer die Zeitabstände sind, desto flüssiger und realistischer wird die Bewegung ausgeführt. Somit wird die Bewegung vom Startpunkt zur Zielposition durch die Trajektorienpunkte vollständig beschrieben und kann im digitalen Modell präzise und gleichmäßig dargestellt werden (Fang u. a. 2019, S. 132–134). Dieser Vorgang wird für jedes Gelenk separat durchgeführt, jeweils bezogen auf die individuelle Bewegungsstrecke.

Die **Bewegungssteuerung** übergibt die berechneten Positionswerte an das digitale Modell. In Visual Components erfolgt dies über sogenannte Value-Signale die durch das Python-Skript fortlaufend aktualisiert werden. Sie geben an, welche Position ein Gelenk zu einem bestimmten Zeitpunkt einnehmen soll. Ein zentraler Controller verarbeitet diese Signale und ordnet sie den richtigen Gelenken zu. Er sorgt dafür, dass die Werte aus der Bewegungsplanung korrekt an die Bewegungsausführung weitergeleitet werden.

Die **Bewegungsausführung** erfolgt über sogenannte Behaviors, die mit den Gelenken verbunden sind. Sie setzen die vom Controller bereitgestellten Positionswerte in sichtbare Bewegungen um. Die Art der Bewegung ergibt sich dabei aus der im Modell hinterlegten Kinematikstruktur. So wird die geplante Bewegung im digitalen Modell realitätsnah dargestellt (Visual Components 2018).

3.3 IIoT Kommunikationstechnologien

Das industrielle Internet der Dinge (IIoT) verfolgt im Kontext von Industrie 4.0 das Ziel, Maschinen, Anlagen und Menschen digital zu vernetzen, um einen reibungslosen

Informationsaustausch zu ermöglichen und Produktionsprozesse effizienter zu gestalten. Es bildet die Grundlage für die übergreifende Überwachung und Steuerung von Fertigungsprozessen – auch standortübergreifend. Ermöglicht wird dies durch moderne Kommunikationstechnologien, die eine durchgängige und zuverlässige Verbindung zwischen allen beteiligten Systemen sicherstellen (Babel 2024, S. 95). Der folgende Abschnitt beleuchtet zunächst die Grundlagen der industriellen Kommunikation, bevor anschließend der Ablauf der Datenübertragung beschrieben wird.

3.3.1 Grundlagen der industriellen Kommunikation

Um die technische Umsetzung der im IIoT angestrebten Vernetzung besser zu verstehen, ist ein Blick auf die zugrunde liegenden Kommunikationsstrukturen erforderlich. Laut Babel bildet die industrielle Kommunikation die Grundlage für den verlässlichen Austausch von Informationen. Zur systematischen Beschreibung dieser Prozesse haben sich zwei ergänzende Modelle etabliert: die Automatisierungspyramide und das OSI-Schichtenmodell. (Babel 2023, S. 21).

Wie Babel erläutert, gliedert die **Automatisierungspyramide** die industrielle Kommunikation in fünf hierarchisch aufgebaute Ebenen innerhalb eines Unternehmens: die Feldebene (Sensoren und Aktoren), die Steuerungsebene (z. B. speicherprogrammierbare Steuerungen), die Leitebene (Prozessvisualisierung und -überwachung mittels SCADA/HMI), die Betriebsebene (Manufacturing Execution Systems) und die Unternehmensebene (Enterprise Resource Planning) (Babel 2023, S. 24). Babel weist darauf hin, dass diese Struktur sowohl eine vertikale als auch eine horizontale Kommunikation innerhalb des Unternehmens ermöglicht. Damit liefert die Automatisierungspyramide eine strukturierende Antwort auf die Frage, wo und zwischen welchen Systemen Kommunikation stattfindet.

Ergänzend dazu beschreibt das **OSI-Schichtenmodell** (Open Systems Interconnection Model) die technische Seite der Datenübertragung. Es unterteilt die Kommunikation in sieben aufeinander aufbauende Schichten, von der physikalischen Signalübertragung bis zur Anwendung. Die untersten Schichten (Schicht 1 und 2), die Bitübertragungsschicht und die Sicherungsschicht, regeln die physikalische Übertragung von Daten sowie die fehlerfreie Kommunikation. Darüber liegt die Vermittlungsschicht (Schicht 3), die für die Adressierung und Weiterleitung der Daten durch das Netzwerk verantwortlich ist. Die Transportschicht (Schicht 4) stellt sicher, dass Daten vollständig und korrekt beim Empfänger ankommen. Die Sitzungsschicht (Schicht 5) organisiert den Austausch zwischen zwei Kommunikationspartnern. Die Darstellungsschicht (Schicht 6) sorgt dafür, dass Daten in ein einheitliches Format

übersetzt werden. Außerdem übernimmt sie Aufgaben wie die Verschlüsselung von Daten. Die Anwendungsschicht (Schicht 7) stellt schließlich die Verbindung zu Benutzerprogrammen her. Damit beantwortet das OSI-Modell die Frage, wie industrielle Kommunikation technisch realisiert wird (Babel 2023, S. 30–32).

Das **Zusammenspiel** von der Automatisierungspyramide und dem OSI-Schichtenmodell ermöglicht ein umfassendes Verständnis industrieller Kommunikationssysteme. Während die Automatisierungspyramide die organisatorische Struktur der Kommunikation im Unternehmen beschreibt, bietet das OSI-Modell ein technisches Referenzsystem für deren Umsetzung. Industrielle Netzwerkprotokolle wie PROFINET, EtherCAT, OPC UA oder MQTT orientieren sich an den OSI-Schichten und ermöglichen die Datenübertragung zwischen den in der Automatisierungspyramide definierten Ebenen (Babel 2023, S. 20). Grundsätzlich lassen sich drei Kommunikationsrichtungen unterscheiden. Die vertikale Kommunikation beschreibt den Informationsaustausch zwischen verschiedenen Ebenen der Automatisierungspyramide, die horizontale Kommunikation erfolgt zwischen Systemen auf derselben Ebene und systemübergreifende Kommunikation umfasst den Austausch mit externen Systemen (Babel 2023, S. 31).

3.3.2 Ablauf industrieller Datenübertragung

Nachdem der vorherige Abschnitt die theoretischen Grundlagen der industriellen Kommunikation behandelt hat, widmet sich dieser Abschnitt dem konkreten Ablauf der Datenübertragung. Ziel ist es, das abstrakte OSI-Schichtenmodell mit realen Technologien der industriellen Praxis zu verknüpfen und dadurch ein vertieftes Verständnis der zugrunde liegenden Kommunikationsmechanismen zu ermöglichen. Im Fokus steht eine systemübergreifende Datenübertragung, bei der beispielsweise Betriebsdaten aus der Leitebene (Ebene 3 der Automatisierungspyramide) an ein externes System übermittelt werden. Laut Gronau lässt sich dieser Prozess mithilfe eines sogenannten IoT-Stacks strukturieren, der die einzelnen Schichten des OSI-Modells konkreten Technologien zuordnet. Abbildung 9 veranschaulicht diese Zuordnung: Die unteren Schichten (1–4) nutzen weit verbreitete Netzwerktechnologien wie Ethernet, IPv4 und TCP für die physikalische und transporttechnische Übertragung. Der Schwerpunkt liegt jedoch auf den anwendungsnahen Schichten 5 bis 7, die durch das Protokoll MQTT realisiert werden und den eigentlichen Informationsaustausch zwischen den Systemen ermöglichen (Gronau 2018, S. 5–7).

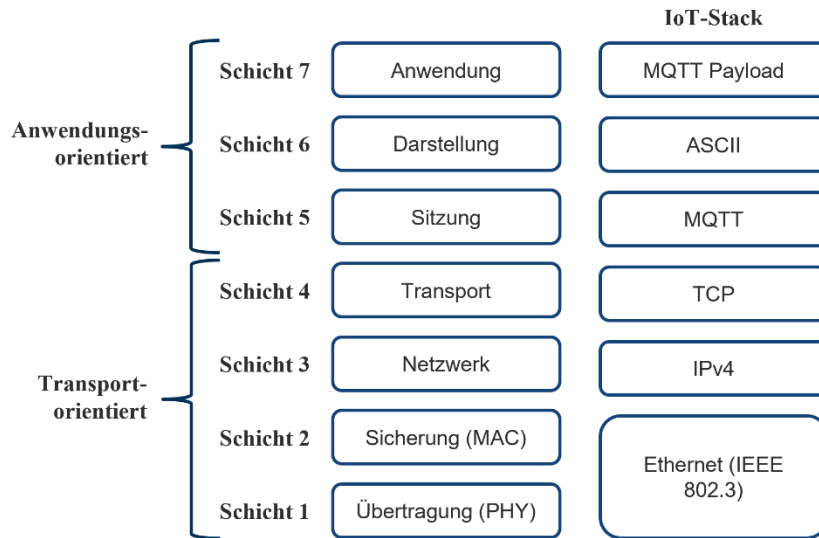


Abbildung 9: Technologiezuordnung im OSI-Modell zur IIoT-basierten Datenübertragung

Quelle: in Anlehnung Gronau 2018, S. 5

Laut EMQX ist MQTT (Message Queuing Telemetry Transport) ein Kommunikationsprotokoll, das speziell für die Anforderungen des industriellen Internets der Dinge (IIoT) entwickelt wurde. Es basiert auf dem Publish-Subscribe-Prinzip, bei dem Nachrichten nicht direkt an einen Empfänger gesendet werden, sondern an einen zentralen Vermittlungsdienst, den sogenannten Broker (siehe Abbildung 10). Ein sendendes Gerät, der sogenannte Publisher, übermittelt seine Nachricht, auch Payload genannt, an den Broker und weist sie einem bestimmten Topic zu. Ein Topic ist ein thematisch strukturierter Nachrichtenkanal, der als thematische Kennzeichnung dient. Empfangende Geräte, sogenannte Subscriber, registrieren sich beim Broker für ein oder mehrere Topics. Der Broker prüft bei jeder eingehenden Nachricht, welchem Topic sie zugeordnet ist, und leitet sie gezielt an alle Subscriber weiter, die dieses Topic abonniert haben. Auf diese Weise wird eine entkoppelte Kommunikation ermöglicht, bei der Sender und Empfänger weder gleichzeitig aktiv sein noch sich gegenseitig kennen müssen (EMQX Team 2025).

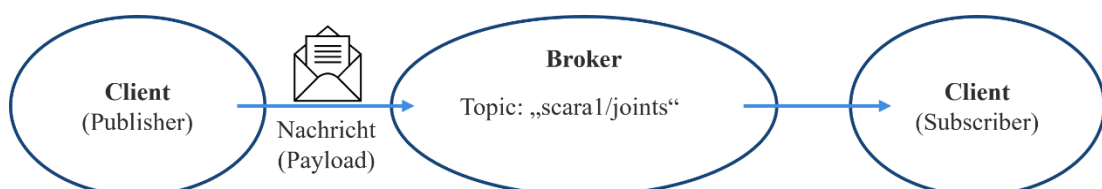


Abbildung 10: Publisher/Subscriber Modell

Quelle: Eigene Darstellung

Im Folgenden wird exemplarisch erläutert, wie eine Nachricht im Rahmen einer MQTT-basierten Kommunikation den OSI-Schichtenstapel vom Sender bis zum Empfänger durchläuft. Der Ablauf orientiert sich an der von Babel beschriebenen OSI-

Modellstruktur mit den jeweiligen Schichtfunktionen und wird auf das Protokoll MQTT übertragen. Ausgangspunkt ist eine Anwendung auf der Senderseite, die beispielsweise Positionsdaten erzeugt, etwa in der Form eines Payloads wie „32.0;55.1;12.9“. Dieser Payload wird in der Anwendungsschicht (Schicht 7) generiert. In der Darstellungsschicht (Schicht 6) erfolgt die Kodierung in ein standardisiertes Format, in diesem Fall als ASCII-Zeichenkette. Anschließend stellt die Sitzungsschicht (Schicht 5) die Verbindung über das MQTT-Protokoll her und überträgt die Nachricht an einen MQTT-Broker. Die darunterliegenden Schichten übernehmen die technische Übertragung. Die Transportschicht (Schicht 4) verwendet TCP, um eine zuverlässige Ende-zu-Ende-Kommunikation sicherzustellen. Die Vermittlungsschicht (Schicht 3) nutzt IPv4 zur Adressierung der Zielsysteme. Schließlich erfolgt auf den Schichten 2 und 1 (Sicherung und Bitübertragung) die Datenrahmung sowie die physikalische Übertragung über Ethernet. Auf Empfängerseite durchläuft die Nachricht den OSI-Stapel in umgekehrter Richtung: Die physikalisch übertragenen Bits werden empfangen und durch die Sicherungsschicht auf Fehlerfreiheit geprüft. Danach sorgen Netzwerk- und Transportschicht für korrekte Adressierung und vollständige Datenübertragung. In der Sitzungsschicht erkennt das System anhand des zugehörigen Topics, etwa „scaral/joints“, ob die Nachricht relevant ist. Die Darstellungsschicht konvertiert die Zeichenkette zurück in ein lesbares Format, bevor die Anwendungsschicht die Daten schließlich verarbeitet. Dieser Ablauf verdeutlicht die Rolle des OSI-Modells als strukturierendes Kommunikationskonzept, das durch konkrete Technologien wie MQTT, TCP/IP und Ethernet in der industriellen Praxis umgesetzt wird (Babel 2023, S. 31–36).

3.4 3D-Integrationsplattform

In diesem Kapitel wird die 3D-Integrationsplattform als zentrale Umgebung vorgestellt, in der digitale Modelle, Echtzeitdatenströme und Interaktionsmöglichkeiten zu einem digitalen Produktzwilling zusammengeführt werden. Obwohl es im industriellen Umfeld bislang keine universell standardisierte Plattform für das industrielle Metaverse existiert, bietet NVIDIA mit Omniverse eine beispielhafte und praxisnahe Lösung (Ullrich u. a. 2024, S. 187). Die folgenden Abschnitte beleuchten die zentralen Konzepte, die technischen Strukturen und Integrationsmechanismen der Plattform.

3.4.1 Omniverse als Plattform

Omniverse ist eine offene, modulare Entwicklungs- und Kollaborationsplattform von NVIDIA. In Anlehnung an Abschnitt 2.2.3 fungiert Omniverse als zentrale 3D-Integrationsumgebung für den digitalen Produktzwilling. Die Plattform vereint 3D-Modelle, IoT-Daten, Simulationen sowie Künstliche Intelligenz in einer gemeinsamen, skalierbaren Struktur. Mithilfe von Erweiterungstools und integrierten Services entstehen anpassbare Anwendungen, die präzise, dynamische und kollaborative digitale Zwillinge in Echtzeit ermöglichen (NVIDIA 2025c). Der Zugriff erfolgt über verschiedene Benutzerschnittstellen wie lokale oder virtuelle Workstations, webbasierte Oberflächen oder XR-Geräte. Diese Interfaces ermöglichen eine kontextbezogene und rollenbasierte Nutzung der digitalen Inhalte. Grundlage aller Schnittstellen ist der Omniverse Nucleus, der als zentrale Daten- und Kommunikationsplattform dient und die Synchronisation zwischen Anwendungen, Nutzern und Datenquellen steuert (NVIDIA 2025f).

3.4.2 Datenstruktur und -verwaltung

Im Rahmen der in Abschnitt 2.2.3 beschriebenen Systemstruktur des digitalen Produktzwillings im industriellen Metaverse kommt der Datenstruktur und der Datenverwaltung eine zentrale Bedeutung zu. Die technische Grundlage für den Austausch und die Speicherung digitaler Inhalte bildet das Universal Scene Description (USD)-Format, das innerhalb der Plattform von NVIDIA als einheitliche Datenstruktur dient. Ergänzend übernimmt die Komponente Omniverse Nucleus die Speicherung und Verwaltung dieser Inhalte (NVIDIA 2025h).

Für den erfolgreichen Aufbau eines industriellen Metaverse sind offene, standardisierte und interoperable Lösungen erforderlich, die eine reibungslose Zusammenarbeit in Echtzeit ermöglichen (Waxer 2023, S. 19). USD erfüllt diese Anforderungen, indem es ein offenes Framework sowie ein plattformunabhängiges Dateiformat bereitstellt. Es erlaubt die strukturierte, modulare und versionierbare Beschreibung komplexer 3D-Szenen. Ursprünglich von Pixar entwickelt, bietet USD eine gemeinsame Sprache für unterschiedliche Tools und Systeme. Damit unterstützt es sowohl die Interoperabilität als auch die Wiederverwendbarkeit und Erweiterbarkeit digitaler Inhalte (Pixar 2021).

Die Speicherung und Verwaltung der USD-basierten Inhalte erfolgt in Omniverse über die zentrale Komponente Nucleus. Dabei handelt es sich um eine datenbankgestützte Kollaborations-Engine, die den gleichzeitigen Zugriff auf digitale Inhalte durch verschiedene Anwendungen und Nutzer ermöglicht. Nucleus organisiert die Daten und stellt essenzielle Funktionen wie Versionierung, Zugriffskontrolle und

Nutzerverwaltung bereit. Auf diese Weise bleiben Änderungen am digitalen Produktzwilling nachvollziehbar (NVIDIA 2023).

3.4.2 Datenintegration

Im industriellen Metaverse ist die **Integration digitaler Modelle** ein zentraler Bestandteil für die Umsetzung von digitalen Produktzwillingen (vgl. Kapitel 2.2.3). NVIDIA Omniverse stellt hierfür mit dem sogenannten USD-Konnektor eine technische Lösung bereit, um Modelle aus Autorensystemen nahtlos in die Plattform zu übertragen (Ullrich u. a. 2024, S. 187). Der Konnektor konvertiert anwendungsspezifische Formate in das standardisierte USD-Format, das als Basis der 3D-Datenstruktur in Omniverse dient (NVIDIA 2025j). Nach Angaben von NVIDIA besteht ein USD-Konnektor aus zwei Hauptkomponenten: dem File Format Plugin und dem Asset Resolver Plugin. Das File Format Plugin übersetzt das Modell in eine USD-Datei und erhält dabei Geometrien, Materialien und Metadaten. Der Asset Resolver sorgt dafür, dass Dateien unabhängig vom Speicherort gefunden und geladen werden können (NVIDIA 2025g). Wie bereits in Abschnitt 2.2.2 erläutert, sehen Ullrich et al. einen wesentlichen Vorteil des USD-Konnektors in der Unterstützung unterschiedlicher Integrationsmechanismen. Besonders die Live-Synchronisierung erlaubt es, Änderungen im Ursprungsmodell in Echtzeit in Omniverse zu übernehmen. Dadurch entsteht eine konsistente, medienbruchfreie Entwicklungsumgebung über mehrere Tools hinweg (Ullrich u. a. 2024, S. 187). Der typische Ablauf beginnt mit dem Modellieren in einem 3D-Tool wie Visual Components, gefolgt von der Aktivierung des Konnektors, der Konvertierung ins USD-Format und dem Upload in Omniverse Nucleus. Dort wird das Modell automatisch geladen und kontinuierlich aktualisiert (NVIDIA 2025e).

Ergänzend zur Modellintegration spielt auch die **Integration von Echtzeitdaten** eine zentrale Rolle. In Abschnitt 2.2.3 wird diese Verbindung als Kopplungseinheit beschrieben. In NVIDIA Omniverse erfolgt die Anbindung solcher Datenströme über spezialisierte Microservices wie die IoT Services, die Daten beispielsweise über das Protokoll MQTT erfassen, verarbeiten und an Nucleus übermitteln (NVIDIA 2025b). Ariansyah et al. betonen, dass die empfangenen Daten korrekt mit dem digitalen Modell verknüpft werden müssen, um eine präzise räumliche und zeitliche Synchronisierung zu gewährleisten. Dieses sogenannte **Daten-Mapping** stellt sicher, dass alle Informationen aus der realen Welt eindeutig den entsprechenden Komponenten des virtuellen Modells zugewiesen werden (Ariansyah u. a. 2020, S. 2). Möglich wird dies durch die Struktur des USD-Formats, in dem jede Modellkomponente als eindeutig benannter Prim dargestellt wird. Diese Prims

verfügen über Eigenschaften wie Position oder Bewegungswinkel, denen die Echtzeitdaten direkt zugewiesen werden. Das System erkennt dadurch automatisch, welches Datenpaket zu welchem Element des digitalen Modells gehört. Die kontinuierliche Datenübertragung sorgt dafür, dass das Modell stets mit aktuellen Werten aus der realen Umgebung versorgt wird (NVIDIA 2025i). Auf diese Weise entsteht ein vollwertiger digitaler Produktzwilling in Form eines Datenzwillings innerhalb des industriellen Metaverse.

4. Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse

Das industrielle Metaverse befindet sich aktuell in einer explorativen Phase, in der technologische Konzepte erprobt, potenzielle Anwendungen identifiziert und erste Pilotprojekte umgesetzt werden (Ullrich u. a. 2024, S. 190). Vor diesem Hintergrund lässt sich feststellen, dass bislang keine etablierten Vorgehensmodelle vorliegen, die explizit auf die Erstellung eines digitalen Produktzwillings im industriellen Metaverse ausgerichtet sind. Ziel dieses Kapitels ist es, auf dieser Ausgangslage ein Vorgehensmodell zu entwickeln, das die strukturierte und methodisch fundierte Erstellung eines digitalen Produktzwillings im industriellen Metaverse ermöglicht. Dazu wird zunächst erläutert, was unter einem Vorgehensmodell zu verstehen ist. Im Anschluss werden die Anforderungen an ein solches Modell systematisch abgeleitet, sowohl aus der Perspektive des digitalen Produktzwillings als auch unter Berücksichtigung der spezifischen Rahmenbedingungen des industriellen Metaverse. Danach werden bestehende Vorgehensmodelle vorgestellt und hinsichtlich ihrer Eignung bewertet. Abschließend wird auf Basis der identifizierten Lücken ein geeignetes, erweitertes Vorgehensmodell entwickelt.

4.1 Begriffsdefinition Vorgehensmodell

Projekte zeichnen sich laut Jakoby durch sechs wesentliche Merkmale aus: ein klares Ziel, komplexe Aufgaben, prozessorientierte Abläufe, organisatorischen Bedarf, begrenzte Ressourcen und feste Zeitvorgaben. Diese Merkmale bringen spezifische Herausforderungen mit sich, die gezielte und projektspezifische Maßnahmen erforderlich machen. Trotz individueller Unterschiede lassen sich bewährte Methoden und Werkzeuge oft auf verschiedene Projekte übertragen. Im Laufe der Zeit wurden solche Methoden zu ganzheitlichen Vorgehensmodellen weiterentwickelt, die durch ihr Zusammenspiel eine breite Anwendbarkeit gewährleisten (Jakoby 2021, S. 27–28).

Ein Vorgehensmodell beschreibt einen strukturierten und standardisierten Projektablauf, der alle Phasen eines Projekts von der Planung bis zur Fertigstellung systematisch organisiert (Jenny 2014, S. 968). Es handelt sich um eine projektübergreifende Methode, mit der sich Vorhaben gezielt umsetzen lassen (Jenny 2014, S. 142). Man unterscheidet zwischen übergreifenden und spezialisierten Vorgehensmodellen, wobei Letztere gezielt auf die spezifischen Anforderungen eines Unternehmens oder einer Branche zugeschnitten sind (Timinger 2017, S. 30). Lindemann hebt hervor, dass Vorgehensmodelle zur Transparenz beitragen, Risiken minimieren, eine klare Struktur schaffen und die Erfolgsaussichten eines Projekts

erhöhen. Zudem fördern sie ein effektives Wissensmanagement, indem sie sowohl die Dokumentation als auch die Weitergabe von Erfahrungen und Know-how systematisieren (Lindemann 2009, S. 33).

Zur genaueren Strukturierung bestehen Vorgehensmodelle aus mehreren zentralen Elementen. Timinger beschreibt als wesentliches Merkmal das Phasenmodell, das den Projektverlauf in aufeinanderfolgende Abschnitte gliedert. Meilensteine dienen als Kontrollpunkte zur Überprüfung des Projektfortschritts. Aktivitäten bezeichnen konkrete Arbeitsschritte innerhalb einer Phase, die zur Zielerreichung notwendig sind. Ergänzt wird dies durch die Integration geeigneter Methoden, Werkzeuge und Regelungen, die den Projektverlauf strukturieren. Auch Arbeits- und Hilfsmittel unterstützen die Umsetzung. Schließlich definieren Vorgehensmodelle klar die Rollen der Beteiligten, um Verantwortlichkeiten transparent zuzuordnen (Timinger 2017, S. 30).

4.2 Anforderungen an das Vorgehensmodell

Anforderungen spielen eine zentrale Rolle bei der Entwicklung eines Vorgehensmodells, da sie den Rahmen für dessen Struktur, Inhalte und Funktionalitäten vorgeben. Im Allgemeinen bezeichnen Anforderungen die Eigenschaften oder Funktionen, die ein Objekt erfüllen muss, um seinem Zweck gerecht zu werden (Jakoby 2021, S. 78). Übertragen auf den vorliegenden Kontext stellt das Objekt das zu entwickelnde Vorgehensmodell dar, dessen Ziel es ist, die Erstellung eines digitalen Produktzwillings im industriellen Metaverse systematisch zu unterstützen. Ausgehend davon lassen sich **drei wesentliche Anforderungsebenen** identifizieren: methodische Anforderungen, inhaltliche Anforderungen aus Sicht des digitalen Produktzwillings sowie zusätzliche Anforderungen durch dessen Einbettung in das industrielle Metaverse.

Zur Sicherstellung eines strukturierten, nachvollziehbaren und zugleich anpassungsfähigen Entwicklungsprozesses sind zunächst grundlegende **methodische Anforderungen** zu definieren. In Anlehnung an die Prinzipien von Trauer et al. lassen sich drei zentrale Kriterien benennen: Verständlichkeit, Universalität und Flexibilität (Follath/Bross/Galka 2022, S. 694). Die inhaltliche Herleitung folgt den in Abschnitt 4.1 dargestellten Strukturelementen. Verständlichkeit bezieht sich auf eine klare, logisch aufgebaute Struktur des Vorgehens, untergliedert in aufeinanderfolgende Phasen und ergänzende Aktivitäten. Universalität beschreibt die übergreifende Anwendbarkeit des Modells auf unterschiedliche Branchen- und Technologieszenarien. Voraussetzung dafür ist ein modularer Aufbau, der die

Integration etablierter Methoden und Werkzeuge ermöglicht. Flexibilität wiederum meint die Fähigkeit des Modells, sich an neue Erkenntnisse und veränderte Rahmenbedingungen anzupassen. Dies umfasst etwa die Möglichkeit, zwischen Phasen zu iterieren sowie Risiken frühzeitig zu identifizieren und zu adressieren.

Auf einer **zweiten Ebene** ergeben sich spezifische Anforderungen aus den Charakteristika des **digitalen Produktzwillings**. Da in dieser Arbeit ein reiner Datenzwilling betrachtet wird, stehen in Anlehnung an Stockhausen und Lambertz dabei insbesondere die folgenden **drei Aspekte** im Vordergrund. **Erstens** erfordert der digitale Produktzwilling eine frühzeitige Sicherstellung der Verfügbarkeit und Qualität der Betriebsdaten. Das Vorgehensmodell muss hierzu geeignete Prozesse zur Identifikation, Anbindung und Prüfung relevanter Datenquellen vorsehen. **Zweitens** ist eine realitätsgetreue Abbildung des physischen Objekts erforderlich (Stockhausen/Lambertz 2024, S. 15). Das digitale Modell muss reale Eigenschaften und Verhalten korrekt abbilden, wobei geeignete Modellierungsmethoden sowie eine Validierung der Abbildungsgenauigkeit wesentliche Schritte darstellen (Tao/Zhang/Nee 2019, S. 70). **Drittens** muss ein stabiler Datenfluss gewährleistet werden (Stockhausen/Lambertz 2024, S. 14). Das Vorgehensmodell muss die zuverlässige Übertragung und Synchronisation der Daten zwischen physischem Objekt und digitalem Modell sicherstellen, wobei Aspekte wie Kommunikationsprotokolle und Datenintegrität zu berücksichtigen sind (Follath/Bross/Galka 2022, S. 693).

Die **dritte Ebene** betrifft die erweiterten Anforderungen, die sich aus der Einbindung in das **industrielle Metaverse** ergeben. Dieses verlangt nicht nur die Erstellung eines digitalen Produktzwillings, sondern auch dessen Integration in eine vernetzte, immersive Plattformumgebung (Ullrich u. a. 2024, S. 187). Auf Basis der in Kapitel 2.2.2 dargestellten Referenzarchitektur und der in Kapitel 2.2.3 beschriebenen Systemstruktur lassen sich **drei zentrale Anforderungen** ableiten. **Erstens** muss das Vorgehensmodell eine durchgängige Datenintegration und -verarbeitung sicherstellen. 3D-Daten des digitalen Modells sowie Betriebsdaten des physischen Objekts müssen strukturiert erfasst, über geeignete Schnittstellen übertragen und auf der 3D-Integrationsplattform zusammengeführt werden. **Zweitens** ist eine technische Infrastruktur erforderlich, die eine stabile, latenzarme und flüssige Visualisierung des digitalen Produktzwillings in Echtzeit ermöglicht. **Drittens** sind Plattformanwendungen und Benutzerschnittstellen integraler Bestandteil. Das Vorgehensmodell muss die spätere Interaktion mit dem digitalen Produktzwilling in immersiven, multiuserfähigen Umgebungen ermöglichen.

4.3 Überblick bestehender Vorgehensmodelle

Ziel dieses Abschnitts ist es, geeignete bestehende Vorgehensmodelle zu analysieren und deren Potenzial für die Entwicklung eines anforderungsgerechten Modells zu bewerten. Die Auswahl erfolgte gezielt auf Grundlage der in Kapitel 4.2 formulierten Anforderungen. Der Fokus liegt dabei auf Modellen, die sowohl grundlegende methodische Kriterien als auch zentrale strukturelle Elemente eines digitalen Produktzwillings berücksichtigen. Vorgehensmodelle aus Bereichen wie dem klassischen Projektmanagement oder der Fabrikplanung wurden bewusst nicht einbezogen, da sie häufig keine ausreichenden Aussagen zu Datenintegration, Schnittstellenmanagement und Echtzeitfähigkeit enthalten (Follath/Bross/Galka 2022, S. 694). Nachfolgend werden zwei ausgewählte Modelle vorgestellt und dahingehend bewertet, inwieweit sie den identifizierten Anforderungen gerecht werden. Ziel dieser Bewertung ist es, bestehende Lücken sichtbar zu machen und herauszuarbeiten, in welchen Bereichen eine gezielte Erweiterung erforderlich ist, um die spezifischen Anforderungen zu erfüllen.

4.3.1 Auswahl und Beschreibung relevanter Modelle

Die Auswahl konzentriert sich auf zwei Vorgehensmodelle (Modell 1 und Modell 2), die auf dem etablierten CRISP-DM-Zyklus (Cross Industry Standard Process for Data Mining) basieren, jedoch unterschiedliche Erweiterungsperspektiven verfolgen.

Modell 1, das von Follath, Bross und Galka vorgestellte Vorgehensmodell, adressiert die strukturierte Erstellung digitaler Zwillinge im Kontext von Produktion und Logistik. Es baut methodisch auf dem CRISP-DM-Zyklus auf, wurde jedoch gezielt um spezifische Anforderungen aus der industriellen Praxis ergänzt. Ziel des Modells ist es, einen systematischen Entwicklungsprozess für digitale Zwillinge bereitzustellen, der sowohl den technischen Aufbau als auch die datengetriebene Ausgestaltung berücksichtigt. Das Modell gliedert sich in sechs iterativ durchlaufbare Phasen: Betriebsverständnis, Systemverständnis, Systemvorbereitung, Modellierung, Evaluierung und Anwendung. Jede Phase ist durch klar definierte Aufgaben und Zielstellungen charakterisiert. Die iterative Struktur erlaubt es, bei neuen Erkenntnissen gezielt in frühere Phasen zurückzugehen. Eine Besonderheit des Modells liegt im Fokus auf dem Datenaspekt. Bestehende Datenquellen, ihre Qualität und Struktur sowie deren Nutzarmachung für die Modellbildung stehen im Zentrum. Allerdings adressiert das Modell explizit nicht die Bereitstellung fehlender technischer Infrastrukturen. Diese werden als Voraussetzungen betrachtet, die außerhalb des dargestellten Vorgehens liegen und gegebenenfalls durch vorgelagerte Projekte realisiert werden müssen. Insgesamt bietet das Modell eine methodisch transparente

und praxisnahe Grundlage für die Entwicklung datenbasierter digitaler Zwillinge (Follath/Bross/Galka 2022, S. 691–695).

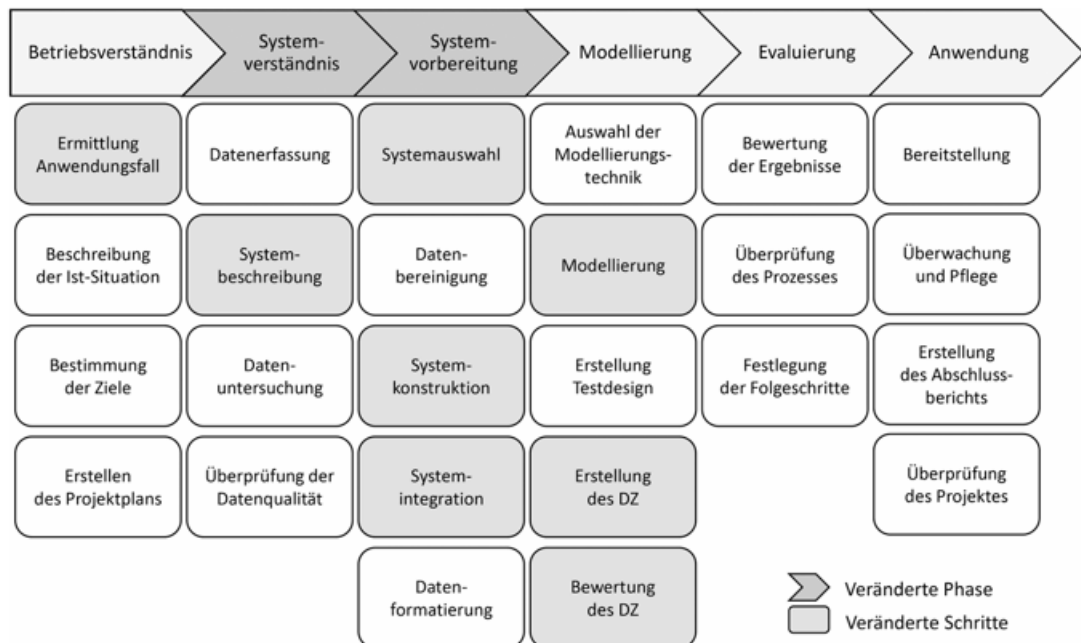


Abbildung 11: Vorgehensmodell 1 nach Follath Bross und Galka

Quelle: Follath/Bross/Galka 2022, S. 694

Modell 2, das von Huber et al. vorgestellte Vorgehensmodell „Data Mining Methodology for Engineering Applications“ (DMME), erweitert den klassischen CRISP-DM-Zyklus gezielt um ingenieurwissenschaftliche Anforderungen. Es wurde mit dem Ziel konzipiert, datengetriebene Analyseprojekte im technischen Umfeld methodisch zu unterstützen und dabei die notwendige physisch-technische Infrastruktur aktiv zu berücksichtigen. Neben den bekannten Phasen von CRISP-DM führt das Modell drei zusätzliche Phasen ein: technisches Verständnis, technische Realisierung und technische Implementierung. Diese betreffen unter anderem die Identifikation technischer Lücken, die Entwicklung entsprechender Konzepte sowie deren Umsetzung. DMME bietet damit eine praxisorientierte Erweiterung für Projekte, bei denen nicht nur Daten analysiert, sondern zuvor eine tragfähige technische Grundlage geschaffen werden muss (Wiemer/Drowatzky/Ihlenfeldt 2019, S. 406–407).

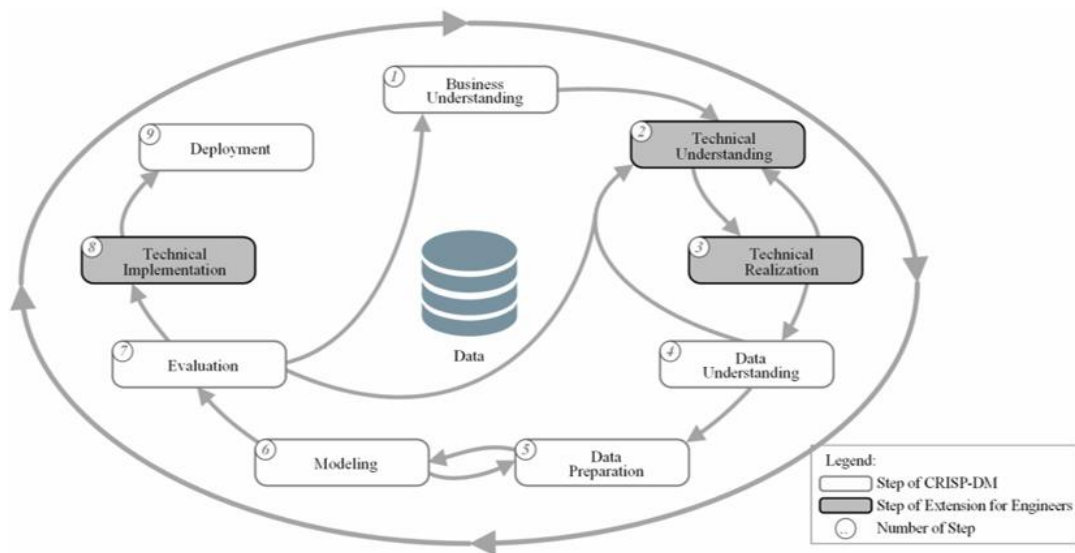


Abbildung 12: Vorgehensmodell 2 nach Wiemer, Drowatzky und Ihlenfeldt

Quell: Wiemer/Drowatzky/Ihlenfeldt 2019, S. 406

4.3.2 Bewertung anhand der Anforderungen

Auf Basis, der in Kapitel 4.2 definierten Anforderungen werden die beiden ausgewählten Modelle im Folgenden bewertet. Untersucht wird, inwieweit sie die methodischen Anforderungen, die Anforderungen des digitalen Produktwillings sowie die des industriellen Metaverse erfüllen. Ziel ist es, bestehende Lücken systematisch zu identifizieren.

Hinsichtlich der Verständlichkeit bietet Modell 1 eine klar strukturierte, sechsstufige Struktur mit definierten Phasen und Aufgaben (Follath/Bross/Galka 2022, S. 694). Die Orientierung am CRISP-DM-Zyklus unterstützt die Nachvollziehbarkeit und sorgt für eine transparente Projektstruktur. Modell 2 erweitert den CRISP-DM-Zyklus um zusätzliche technische Phasen und erhöht dadurch die inhaltliche Tiefe (Huber u. a. 2019, S. 406). Beide Modelle erfüllen somit das Kriterium der Verständlichkeit. Bezüglich der Universalität zeigt Modell 1 eine breite Anwendbarkeit in datengetriebenen Projekten, setzt jedoch eine bestehende IT- und Dateninfrastruktur voraus (Follath/Bross/Galka 2022, S. 695). In Umgebungen mit unzureichender Infrastruktur stößt es daher an seine Grenzen. Modell 2 adressiert diese Schwäche, indem es die Schaffung technischer Grundlagen explizit in das Vorgehen integriert (Huber u. a. 2019, S. 407). Während Modell 1 große Teile der Anforderung erfüllt, ermöglicht erst die Ergänzung durch Modell 2 eine vollständige Abdeckung. Im Hinblick auf die Flexibilität ermöglichen beide Modelle Rücksprünge zwischen den Phasen und damit eine adaptive Reaktion auf neue Erkenntnisse (Follath/Bross/Galka 2022, S. 694). Modell 2 geht jedoch darüber hinaus, indem es auch infrastrukturelle Anpassungen innerhalb des Modells vorsieht und damit eine höhere

Anpassungsfähigkeit gegenüber dynamischen Rahmenbedingungen bietet. Beide Modelle erfüllen die Anforderung der Flexibilität, wobei Modell 2 einen erweiterten Spielraum eröffnet.

Neben den methodischen Anforderungen müssen Vorgehensmodelle auch spezifische Anforderungen des digitalen Produktzwillings erfüllen. Dazu zählen, wie in Kapitel 4.2 erarbeitet, die Datenverfügbarkeit und -qualität, die Repräsentationsgenauigkeit sowie ein stabiler Datenfluss. Modell 1 berücksichtigt vorhandene Datenquellen und deren Qualität, geht jedoch nicht auf fehlende Infrastrukturen ein (Follath/Bross/Galka 2022, S. 695). Modell 2 schließt diese Lücke durch die Integration technischer Aufbauprozesse (Huber u. a. 2019, S. 404). Bezüglich der Repräsentationsgenauigkeit stützt sich Modell 1 auf eine physikalische Modellierung des physischen Objekts und sieht Test- und Bewertungsschritte vor, wodurch eine realitätsnahe Abbildung gewährleistet wird (Follath/Bross/Galka 2022, S. 695). Modell 2 fokussiert sich auf datengetriebene Modelle und betrachtet keine physikalischen Aspekte (Huber u. a. 2019, S. 404). Beim Datenfluss adressiert Modell 1 den unidirektionalen Austausch zwischen physischem Objekt und digitalem Modell (Follath/Bross/Galka 2022, S. 695). Modell 2 behandelt Datenflüsse vorrangig im Kontext datenbasierter Analysen, ohne eine konkrete Anbindung zwischen physischem Objekt und digitalem Modell auszuführen.

Die Anforderungen des industriellen Metaverse gehen über klassische digitale Produktzwillings-Konzepte hinaus und betreffen insbesondere die Integration in vernetzte, immersive Plattformstrukturen. Bewertet wurden die Aspekte Datenintegration und -verarbeitung, Echtzeitvisualisierung sowie Plattformanwendungen und Benutzerschnittstellen. Modell 1 bietet eine klare Struktur zur Erfassung und Verarbeitung technischer und betrieblicher Daten. Auch die Modellbildung ist methodisch verankert (Follath/Bross/Galka 2022, S. 695). Allerdings bleibt die Integration dieser Daten in eine übergreifende 3D-Integrationsplattform unberücksichtigt. Anforderungen an Visualisierung in Echtzeit sowie die spätere Interaktion mit dem digitalen Produktzwillig im Plattformkontext werden nicht adressiert. Modell 2 fokussiert sich auf den Aufbau technischer Infrastrukturen und die Umsetzung datenbasierter Anwendungen (Huber u. a. 2019, S. 406). Dadurch werden grundlegende Voraussetzungen für Datenintegration geschaffen. Eine methodische Abbildung von Echtzeitvisualisierung oder multiuserfähiger Plattformanbindung fehlt jedoch auch hier. Insgesamt zeigt sich, dass keines der beiden Modelle die Anforderungen des industriellen Metaverse vollständig erfüllt. Insbesondere die Verbindung zwischen physischem Objekt, digitalem Modell

und immersiver Umgebung bleibt methodisch unbehandelt. Daraus ergibt sich ein klarer Bedarf zur Erweiterung der bestehenden Modelle.

Die Bewertung zeigt, dass die Modelle 1 und 2 wertvolle methodische Grundlagen bieten, jedoch zentrale Anforderungen des industriellen Metaverse bislang unberücksichtigt lassen. Daraus ergibt sich die Notwendigkeit, ein erweitertes Vorgehensmodell zu entwickeln, das diese Lücken gezielt adressiert. Der folgende Abschnitt stellt den Aufbau und die Phasen dieses Modells vor.

4.4 Aufbau und Phasen des Vorgehensmodells

Basierend auf den in Abschnitt 4.2 definierten Anforderungen und der Bewertung der bestehenden Vorgehensmodelle in Abschnitt 4.3 wird in diesem Abschnitt ein Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse vorgestellt.

4.4.1 Zielsetzung und Rahmenbedingungen

Im Zentrum dieser Arbeit steht die Entwicklung eines digitalen Produktzwillings in Form eines reinen Datenzwillings. Das vorgestellte Vorgehensmodell orientiert sich an der in Abschnitt 4.1 erläuterten Begriffsdefinition und konzentriert sich gezielt auf die Kernelemente Phasenmodell und Aktivitäten. Weitere Bestandteile klassischer Vorgehensmodelle, wie Methoden, Werkzeuge, Regelungen sowie Arbeits- und Hilfsmittel werden bereits in Kapitel 3 kontextuell eingeordnet und sind daher nicht Gegenstand des folgenden Kapitels. Ebenso bleiben die Aspekte Rollen und Meilensteine bewusst unberücksichtigt. Ziel ist es, ein schlankes und auf die wesentlichen Entwicklungsaktivitäten fokussiertes Vorgehensmodell zu entwerfen, das sowohl den methodischen Anforderungen als auch den spezifischen technologischen und strukturellen Anforderungen eines digitalen Produktzwillings im industriellen Metaverse gerecht wird.

4.4.2 Aufbau des Vorgehensmodells

Das entwickelte Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse basiert auf einer vierstufigen Hauptstruktur, bestehend aus Analysephase, Konzeptionierungsphase, Ausarbeitungsphase und Projektabschluss (Timinger 2017, S. 40). Diese Gliederung dient der strukturierten Einordnung der Unterphasen, die sich aus den identifizierten Anforderungen (vgl. 4.2) der bestehenden Modellen (vgl. 4.3.1) sowie deren Bewertung (vgl. 4.3.2) ableiten lassen.

Analysephase: Diese Phase umfasst drei aufeinanderfolgende Unterphasen. Das Betriebsverständnis und das Systemverständnis stammen aus Modell 1 und dienen der betriebsbezogenen sowie technischen Ausgangsanalyse (Follath/Bross/Galka 2022, S.

694). Ergänzt werden sie durch das technische Verständnis aus Modell 2, das bereits in dieser frühen Phase konkrete technische Zielstellungen und bestehende Defizite identifiziert (Huber u. a. 2019, S. 406).

Konzeptionierungsphase: In dieser Phase werden Lösungsansätze konkretisiert. Die technische Realisierung basiert auf Modell 2 (Huber u. a. 2019, S. 406). Die Zielsystementwicklung geht auf die „Systemvorbereitung“ aus Modell 1 zurück, wurde jedoch umbenannt, um den inhaltlichen Fokus klarer zu betonen (Follath/Bross/Galka 2022, S. 694).

Ausarbeitungsphase: Diese Phase bildet das Herzstück der technischen Implementierung und wurde gezielt erweitert, um den Anforderungen des industriellen Metaverse gerecht zu werden. Die Unterphase Datenbereitstellung bezieht sich auf das physische Objekt und beschreibt die strukturierte Erfassung relevanter Betriebsdaten. Die Unterphase Modellierung adressiert das digitale Modell des Objekts und wurde aus Modell 1 übernommen (Follath/Bross/Galka 2022, S. 694). Die Unterphase Datenintegration und -verarbeitung bezieht sich auf die 3D-Integrationsplattform. Sie unterteilt sich in zwei Teilbereiche: die Kopplungseinheit, die den Betriebsdatenfluss sicherstellt, und die Modellanbindung, welche die Übernahme der digitalen Modelldaten aus Autorensystemen ermöglicht. Die nachfolgende Unterphase Einrichtung von Plattformanwendungen und Benutzerschnittstellen gewährleistet, dass der digitale Produktzwilling interaktiv nutzbar wird. Sie untergliedert sich in die drei funktionalen Komponenten Visualisierung, Interaktion und Kollaboration und adressiert zentrale Anforderungen wie Multiuser-Fähigkeit und Echtzeitverfügbarkeit (NVIDIA 2025c).

Projektabschluss: Der Projektabschluss umfasst die beiden Unterphasen Evaluierung und Anwendung, die aus Modell 1 übernommen wurden (Follath/Bross/Galka 2022, S. 694). Sie dienen der Qualitätssicherung, der Prüfung der Zielerreichung und der Übergabe des digitalen Produktzwillings in die betriebliche Nutzung.

Auch wenn das Vorgehensmodell grundsätzlich einem sequenziellen Ablauf folgt, ist es nicht als strikt linear zu verstehen. Vielmehr erlaubt es gezielte Rücksprünge in vorherige Phasen, um flexibel auf neue Erkenntnisse und veränderte Rahmenbedingungen reagieren zu können (Wiemer/Drowatzky/Ihlenfeldt 2019, S. 4).

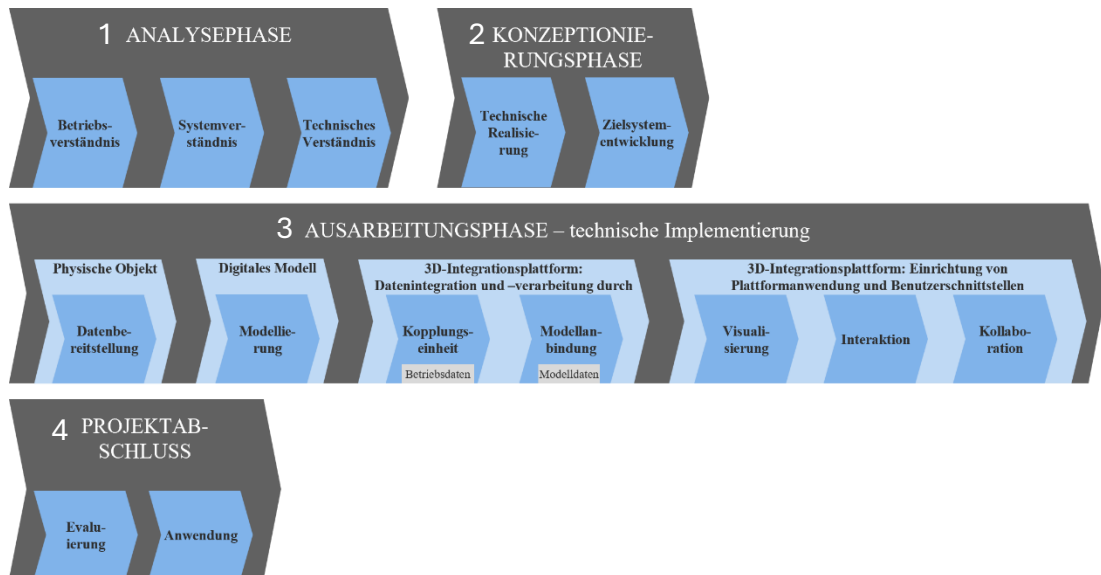


Abbildung 13: Aufbau des Vorgehensmodells

Quelle: Eigene Darstellung

4.4.3 Phasen des Vorgehensmodells

In diesem Kapitel werden die Phasen und Aktivitäten des Vorgehensmodells im Detail erläutert. Eine Übersicht über das vollständige Modell bietet Abbildung 14 (auf Seite 42-43).

Die **Analysephase** besteht aus den drei Unterphasen Betriebsverständnis, Systemverständnis und technisches Verständnis.

Betriebsverständnis: Diese Phase bildet die Grundlage für den gesamten Projektverlauf (Follath/Bross/Galka 2022, S. 694). Ziel ist es, ein umfassendes Verständnis für den betrieblichen Kontext, die bestehenden Rahmenbedingungen sowie die angestrebten Zielsetzungen zu entwickeln (Chapman u. a. 1999, S. 13). Follath et al. beginnen mit der Identifikation der zugrunde liegenden Problemstellung, aus der sich ein spezifischer Anwendungsfall systematisch ableiten lässt. Dadurch wird die zu lösende Aufgabe greifbar gemacht und eine belastbare Ausgangsbasis für die weitere Analyse geschaffen. Die Ist-Analyse beinhaltet eine systematische Erhebung aller relevanten Informationen zum aktuellen Zustand des betrachteten Objekts (Follath/Bross/Galka 2022, S. 694). Dabei werden insbesondere technische Infrastrukturen, bestehende Datenquellen, organisatorische Strukturen sowie verfügbare IT- und Softwarelösungen dokumentiert (Ariansyah u. a. 2020, S. 4). Zur strukturierten Erfassung empfiehlt sich ein Ressourceninventar, das personelle, technische und datenseitige Kapazitäten zusammenfasst (Follath/Bross/Galka 2022, S. 694). Zusätzlich sollten Anforderungen, Annahmen und Einschränkungen systematisch dokumentiert werden (Caulfield 2021, S. 4). Auf Basis des identifizierten Anwendungsfalls und der Ist-Analyse erfolgt die Formulierung konkreter Projektziele

(Chapman u. a. 1999, S. 18). Abschließend wird ein strukturierter Projektplan erstellt, der als operativer Fahrplan für die Umsetzung dient (Follath/Bross/Galka 2022, S. 695). Dieser umfasst Zeitpläne, Arbeitspakete, Zuständigkeiten und den geplanten Ressourceneinsatz (Chapman u. a. 1999, S. 19).

Systemverständnis: Follath et al. definieren diese Phase als eine strukturierte Analyse des Status quo, die den Ausgangspunkt für die Umsetzung bildet. Ziel ist es, die Datenquellen sowie den Aufbau und die Funktion des betrachteten Objekts zu analysieren, um ein fundiertes Bild des Ist-Zustands zu erhalten (Follath/Bross/Galka 2022, S. 695). Chapman et al. beschreiben, dass zu Beginn sowohl vorhandene als auch potenziell benötigte Datenquellen systematisch erfasst und analysiert werden. Dabei wird geprüft, welche Informationen bereits vorliegen, welche zur Zielerreichung erforderlich sind und in welchen Bereichen Defizite bestehen. Wichtige Aspekte wie Speicherorte, Datenformate, Zugriffsmöglichkeiten sowie technische und organisatorische Rahmenbedingungen werden erfasst (Chapman u. a. 1999, S. 20–22). Anschließend erfolgt eine detaillierte Beschreibung des bestehenden Systems, basierend auf den zuvor identifizierten Daten. Analysiert werden physische und digitale Strukturen, technische Komponenten, Abläufe und Schnittstellen. Ergänzend können vorhandene Modelle, Visualisierungen oder Prozessdarstellungen zur Einordnung herangezogen werden. Parallel erfolgt eine erste Bewertung geeigneter Methoden und Werkzeuge (Follath/Bross/Galka 2022, S. 695). Die erfassten Datenbestände werden abschließend hinsichtlich Struktur, Vollständigkeit, Relevanz und Qualität bewertet, um eine belastbare Datengrundlage für die weiteren Phasen zu schaffen (Chapman u. a. 1999, S. 22).

Technisches Verständnis: Die Phase stellt den Übergang von der Analyse zur technischen Ausgestaltung dar. Huber et al. beschreiben sie als zentralen Punkt, an dem die zuvor erarbeiteten Anforderungen und Zielsetzungen in konkrete technische Lösungsansätze überführt werden. Sie bildet die Brücke zwischen Bedarf und Realisierbarkeit. Ziel ist es, eine belastbare technische Basis zu schaffen, auf der die spätere Umsetzung systematisch aufbauen kann. Ausgehend von den identifizierten Abweichungen werden technische Aufgabenstellungen abgeleitet und priorisiert (Huber u. a. 2019, S. 405). Im Fokus dieser Phase steht nach Wiemer et al. die Frage, welche technischen Probleme gezielt zu lösen sind, um die angestrebten Projektziele wirkungsvoll zu erreichen. Gleichzeitig erfolgt die Übersetzung der Anforderungen in technisch erfüllbare Zielgrößen (Wiemer/Drowatzky/Ihlenfeldt 2019, S. 5).

Im Anschluss wird geprüft, ob die technischen Zielstellungen mit der bestehenden Infrastruktur umsetzbar sind (Huber u. a. 2019, S. 406). Dabei werden mögliche

Defizite identifiziert, um frühzeitig realistische Voraussetzungen zu schaffen und notwendige Erweiterungen gezielt zu planen.

Die **Konzeptionierungsphase** umfasst die beiden Unterphasen technische Realisierung und Zielsystementwicklung. Sie dient der systematischen Entwicklung und Ausgestaltung technischer Lösungen, die auf den in der Analysephase identifizierten Anforderungen und Defiziten aufbauen.

Technische Realisierung: Die Phase baut unmittelbar auf den zuvor erkannten technischen Defiziten auf (Huber u. a. 2019, S. 406). Wie Wiemer et al. verdeutlichen ist es das Ziel, auf Grundlage der formulierten Zielgrößen sowie der Ergebnisse der Infrastrukturanalyse geeignete technische Konzepte zu entwickeln, um die festgestellten Lücken systematisch zu schließen (Wiemer/Drowatzky/Ihlenfeldt 2019, S. 6). Ein zentrales Ziel besteht darin, frühzeitig Rückschlüsse auf den Reifegrad möglicher Lösungen zu ziehen, um eine fundierte Entscheidungsbasis für den weiteren Projektverlauf zu schaffen (Huber u. a. 2019, S. 406).

Zielsystementwicklung: Follath et al. erklären, dass diese Phase den Übergang von der Auswahl technischer Lösungen zur konkreten Ausgestaltung des zukünftigen Zielsystems bildet. Dabei werden die Erkenntnisse aus den vorhergehenden Phasen zusammengeführt, um das Zielsystems gezielt zu gestalten (Follath/Bross/Galka 2022, S. 695). Ariansyah et al. unterstreichen die Relevanz der Auswahl geeigneter technischer Komponenten, Technologien und Systeme als entscheidende Grundlage für die Umsetzung des definierten Anwendungsfalls. Ein besonderes Augenmerk liegt dabei auf der Kompatibilität mit der bestehenden IT- und Systemlandschaft im Unternehmen. Auf Basis dieser Auswahl wird ein technisches Systemkonzept entwickelt, das beschreibt, wie die gewählten Komponenten umgesetzt und miteinander verbunden werden sollen. Dieses Konzept umfasst insbesondere die Definition der grundlegenden Systemarchitektur, die Beschreibung von Schnittstellen, Datenflüssen sowie die Gestaltung von Zugriffs- und Berechtigungskonzepten. Abschließend wird festgelegt, wie die konzipierten Lösungen effizient in die vorhandene Infrastruktur integriert werden können (Ariansyah u. a. 2020, S. 4–5).

Die **Ausarbeitungsphase** umfasst die technische Implementierung des digitalen Produktzwillings. Sie besteht aus den vier Unterphasen: Datenbereitstellung am physischen Objekt, Modellierung des digitalen Modells, Datenintegration und -verarbeitung in die 3D-Integrationsplattform und Einrichtung von Plattformanwendungen und Benutzerschnittstellen.

Datenbereitstellung: Die Bereitstellung der Daten erfolgt auf Grundlage der zuvor definierten Zielvorgaben. Das physische Objekt wird mit geeigneten Erfassungseinheiten ausgestattet, um Betriebsdaten und Umweltzustände in Echtzeit zu erfassen (Zheng/Yang/Cheng 2019, S. 1143). Diese Daten werden anschließend strukturiert aufbereitet und der Zugriff für die Übertragung vorbereitet (Follath/Bross/Galka 2022, S. 695).

Modellierung: Ziel der Modellierungsphase ist die Erstellung eines digitalen Abbilds des physischen Objekts, das dessen zentrale Eigenschaften, Strukturen und Verhaltensweisen realitätsnah repräsentiert (Follath/Bross/Galka 2022, S. 695). Das daraus entstehende digitale Modell bildet das Herzstück des digitalen Produktzwillings und schafft die Grundlage für dessen Integration in eine übergeordnete 3D-Integrationsplattform (Ullrich u. a. 2024, S. 187). Zu Beginn erfolgt die Auswahl geeigneter Modellierungstechniken (Follath/Bross/Galka 2022, S. 695). Wie Rabe et al. betonen, richtet sich die Auswahl nach Faktoren wie der Komplexität des Objekts, der angestrebten Zielsetzung sowie der Verfügbarkeit relevanter Daten. In dieser Arbeit kommt ein physikalischer Modellierungsansatz zum Einsatz. Auf dieser Basis wird das physische Objekt in einer geeigneten Modellierungsumgebung nachgebildet. Dabei werden sowohl statische Merkmale als auch dynamische Prozesse so modelliert, dass ein konsistentes, funktionsfähiges und erweiterbares digitales Objekt entsteht (Rabe/Spieckermann/Wenzel 2008, S. 46–50). Die Validierung und Verifikation des Modells erfolgt mithilfe gezielter Test- und Vergleichsszenarien. Ziel ist es, die Funktionsfähigkeit zu überprüfen und sicherzustellen, dass das Modell mit dem Verhalten des realen Objekts übereinstimmt (Follath/Bross/Galka 2022, S. 695–696).

Datenintegration und -verarbeitung: Der Schwerpunkt dieser Phase liegt auf der strukturierten Einbindung der Betriebs- und Modelldaten in die 3D-Integrationsplattform sowie deren anschließender Verarbeitung innerhalb dieser Umgebung (Ullrich u. a. 2024, S. 187). Nach Tu et al. übernimmt die sogenannte Kopplungseinheit die technische Verantwortung für die Echtzeit-Betriebsparallelität zwischen physischem Objekt und digitalem Produktzwillings. Hierfür werden geeignete Kommunikationsschnittstellen implementiert und der Zugriff auf die relevanten Datenquellen realisiert. Neben dem Aufbau des Übertragungsmechanismus auf Seiten des physischen Objekts beinhaltet dies auch die Implementierung einer Empfangsstruktur innerhalb der 3D-Plattform. Die Ausgestaltung dieser Kopplungseinheit richtet sich nach den technischen Rahmenbedingungen wie unterstützten Protokollen, Standards und Werkzeugen (Tu u. a. 2024, S. 969). Die konzeptionellen Grundlagen dafür wurden bereits in der Konzeptionierungsphase

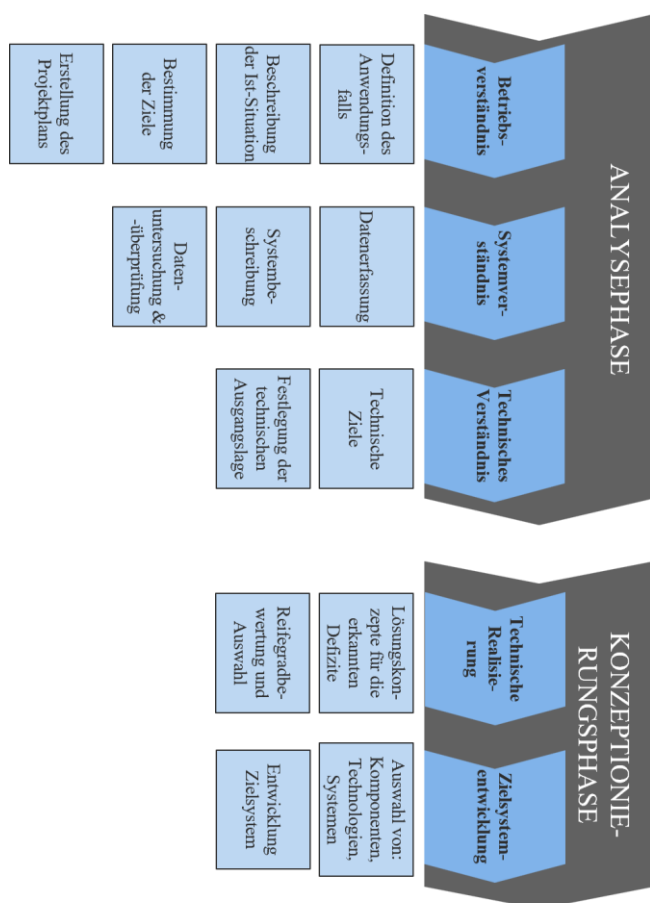
gelegt. Das vollständig entwickelte digitale Modell wird anschließend über einen Konnektor in die 3D-Integrationsplattform überführt, wodurch die technische Integration sichergestellt wird (Ullrich u. a. 2024, S. 187). Tu et al. erklären, nach erfolgreicher Anbindung werden sowohl Betriebs- als auch Modelldaten organisiert, verarbeitet und in einer strukturierten Datenbasis abgelegt. Ein zentraler Schritt ist das sogenannte Daten-Mapping, mit dem die dynamische Echtzeitverbindung zwischen physischem Objekt und digitalem Modell hergestellt wird. Dabei werden relevante Betriebsdaten gezielt modellbezogenen Elementen zugeordnet (Tu u. a. 2024, S. 969). Im Rahmen der anschließenden Validierung und Verifikation wird überprüft, ob der kontinuierliche Datenfluss korrekt interpretiert wird und ob das digitale Modell in Echtzeit auf Veränderungen des physischen Objekts reagieren kann.

Plattformanwendungen und Benutzerschnittstellen: In dieser letzten Unterphase der Ausarbeitungsphase wird sichergestellt, dass der digitale Produktzwilling interaktiv nutzbar wird. Diese Phase gliedert sich in die drei funktionalen Komponenten Visualisierung, Interaktion und Kollaboration, welche die spätere Anwendung in einer immersiven, multiuserfähigen 3D-Umgebung ermöglichen (Ullrich u. a. 2024, S. 189). Im Bereich der **Visualisierung** wird festgelegt, wie die digitalen Inhalte dargestellt werden sollen. Möglich sind Dashboards, Webinterfaces oder XR-Technologien, abhängig vom konkreten Einsatzzweck und den Anforderungen an die Darstellungstiefe (NVIDIA 2025c). Die Komponente **Interaktion** beschreibt, wie Benutzer mit dem digitalen Produktzwilling in Kontakt treten (NVIDIA 2025d). In der vorliegenden Arbeit erfolgt die Interaktion unidirektional, das physische Objekt sendet Daten an das digitale Modell, eine Rückkopplung oder Steuerung des physischen Objekts ist nicht vorgesehen. Die **Kollaboration** stellt im Omniverse sicher, dass mehrere Nutzer gleichzeitig und standortunabhängig mit dem digitalen Produktzwilling arbeiten können. Hierfür sind geeignete Plattformdienste, ein belastbares Datenmanagement sowie Rechte- und Rollenkonzepte erforderlich (NVIDIA 2025a). Im Rahmen der Validierung und Verifikation wird überprüft, ob Visualisierung, Interaktion und Kollaboration stabil und in Echtzeit funktionsfähig sind. Ziel ist es, sicherzustellen, dass alle digitalen Repräsentationen korrekt ablaufen und der digitale Produktzwilling den erwarteten Systemzustand zuverlässig wiedergibt.

Die letzte Phase des Vorgehensmodells, der **Projektabschluss**, umfasst die beiden Unterphasen Evaluierung und Anwendung und dient der systematischen Sicherung der Projektergebnisse sowie der Überführung des digitalen Produktzwillings in den produktiven Betrieb.

Evaluierung: Laut Follath et al. dient diese Phase der Bewertung der bisherigen Ergebnisse und der Reflexion des gesamten Projektverlaufs. Dabei erfolgt ein Abgleich zwischen den erreichten Resultaten und den zuvor definierten Projektzielen (Follath/Bross/Galka 2022, S. 695).

Anwendung: Die abschließende Unterphase, die von Chapman et al. als Anwendung beschrieben wird, markiert den Übergang von der Entwicklung in den operativen Einsatz des digitalen Produktzwillings. Diese Phase beschreibt den Übergang, in dem der digitale Produktzwilling dauerhaft in die betriebliche Praxis überführt wird. Durch kontinuierliche Überwachung, regelmäßige Systemupdates und bedarfsgerechte Anpassungen wird ein stabiler Betrieb sichergestellt. Zudem wird ein Abschlussbericht erstellt, in dem die wesentlichen Ergebnisse, Herausforderungen und Erkenntnisse des Projekts dokumentiert werden. Eine abschließende Projektüberprüfung dient dazu, den Gesamtverlauf kritisch zu bewerten und Empfehlungen für zukünftige Implementierungen abzuleiten (Chapman u. a. 1999, S. 32–44).



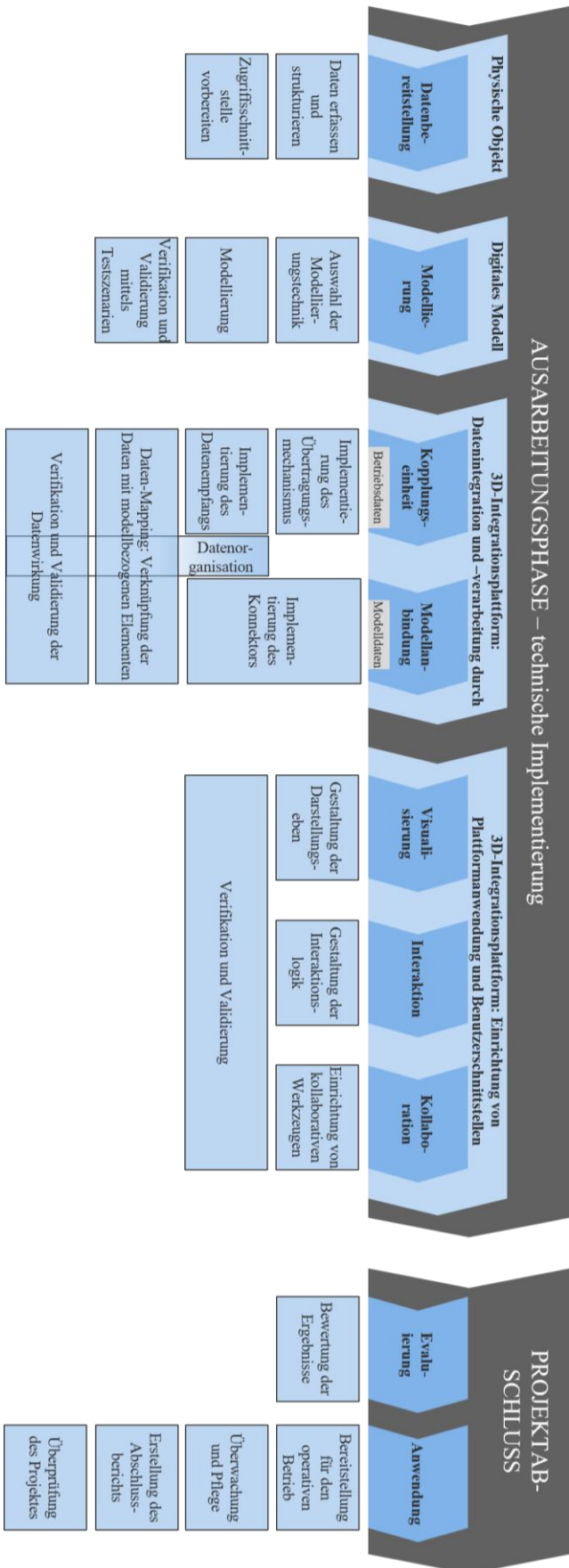


Abbildung 14: Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse

Quelle: Eigene Darstellung

5. Anwendung des Vorgehensmodell am Beispiel Drohnen GmbH

Im folgenden Kapitel wird das zuvor entwickelte Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse exemplarisch an der Drohnen GmbH angewendet.

5.1 Einordnung und Rahmenbedingungen der Anwendung

Die Drohnen GmbH ist ein fiktives Unternehmen, das als integraler Bestandteil des Technologietransferzentrums (TTZ) in Leipheim agiert. Das TTZ verfolgt die Mission, den Wandel zur intelligenten Fabrik als Impulsgeber aktiv mitzugestalten, indem es den Wissens- und Technologietransfer zwischen Hochschule und Industrie fördert. Als Einrichtung der Hochschule Neu-Ulm unterstützt es regionale Unternehmen durch Schulungen, Beratungen, gemeinsame Projekte sowie anwendungsorientierte Forschung bei der Weiterentwicklung smarter Produktions- und Logistiklösungen. Im Rahmen dessen dient die Drohnen GmbH als Lern- und Innovationsfabrik, die ein industrielles Wertschöpfungssystem mit realem, global ausgerichtetem Lieferkettennetzwerk abbildet. Sie bietet eine praxisnahe Umgebung zur Fertigung flugfähiger Drohnen und ermöglicht es, zentrale Konzepte der smarten Produktion und Logistik realitätsnah zu erproben. Damit schafft sie eine ideale Ausgangsbasis zur Anwendung und Validierung digitaler Zukunftstechnologien im Kontext der Industrie 4.0 (Hochschule Neu-Ulm o. J.).

Das in dieser Arbeit entwickelte Vorgehensmodell wird exemplarisch auf die Drohnen GmbH angewendet. Im Mittelpunkt steht dabei die Umsetzung eines digitalen Produktzwillings in Form eines Datenzwillings, der die Bewegungen des SCARA-Roboters i4-550L von Omron in Echtzeit visuell abbildet. Das digitale Modell wird mit der Software Visual Components erstellt (vgl. Kapitel 3.2). Als Zielplattform für die Integration in eine 3D-Umgebung ist NVIDIA Omniverse vorgesehen (vgl. Kapitel 3.4).

Für die Anwendung des Vorgehensmodells gelten spezifische Rahmenbedingungen. Um Wiederholungen zu vermeiden und den Fokus auf das Wesentliche zu richten, wird die Analysephase in dieser Arbeit nicht weiter vertieft. Stattdessen beginnt die Darstellung direkt mit der Konzeptionierungsphase. In diesem Zusammenhang wird der entwickelte Zielsystemaufbau vorgestellt, der Bestandteil dieser Phase ist und auf die Realisierung des zuvor definierten Ziels ausgerichtet wurde. In der darauffolgenden Ausarbeitungsphase ergeben sich Einschränkungen aufgrund fehlender Lizenzrechte für NVIDIA Omniverse. Daraus resultieren folgende spezifische Anpassungen: Die Unterphase Datenintegration und -verarbeitung wird an die bestehenden technischen Rahmenbedingungen angepasst. Eine unidirektionale

Kopplungseinheit zwischen dem SCARA-Roboter und Visual Components wird realisiert, während die ursprünglich geplante Modellanbindung an Omniverse entfällt. Das Daten-Mapping sowie die Datenorganisation erfolgen vollständig innerhalb von Visual Components. Auch die Unterphase Einrichtung von Plattformanwendungen und Benutzerschnittstellen wird lediglich in vereinfachter Form berücksichtigt. Die abschließende Projektabschlussphase wird aus Umfangsgründen nicht weiter behandelt. Verifikations- und Validierungsschritte werden in dieser Arbeit nicht näher betrachtet, da sie den vorgegebenen Rahmen überschreiten.

In Abschnitt 5.2 wird zunächst das vollständige Zielsystem vorgestellt, das auch NVIDIA Omniverse umfasst. Im Anschluss folgt eine angepasste Variante ohne Omniverse, deren Umsetzung in Abschnitt 5.3 detailliert betrachtet wird. Alle entwickelten Skripte werden der Drohnen GmbH in digitaler Form zur Verfügung gestellt.

5.2 Anwendung des Vorgehensmodells: Konzeptionierungsphase

Die folgende Abbildung 15 veranschaulicht den praktischen Aufbau des angestrebten Zielsystems. Sie stellt das Ergebnis der Konzeptionierungsphase, konkret der Unterphase Zielsystementwicklung, dar und wird im Folgenden erläutert. Das Zielsystem orientiert sich an der in Kapitel 2.2.3 beschriebenen Systemstruktur (vgl. Abbildung 5) und überführt deren konzeptionelle Darstellung in eine konkrete technische Umsetzung. Das Zielsystem gliedert sich in drei zentrale Komponenten:

Das digitale Modell wird mithilfe der Software Visual Components realisiert und umfasst sämtliche Elemente der Modellbildung (vgl. Kapitel 3.2). Dazu zählen die geometrische Struktur des Roboters, seine kinematischen Freiheitsgrade sowie die Bewegungslogik, die durch ein Python-Skript implementiert wird (vgl. Kapitel 3.1). Die Modellierung bildet die Grundlage für die spätere Echtzeitvisualisierung des Roboters im virtuellen Raum. Das physische Objekt ist ein realer SCARA-Roboter, dessen relevante Betriebsdaten über geeignete Sensorik erfasst und mithilfe von Programmen (in der Programmiersprache V+- und C#) strukturiert und für die Übertragung vorbereitet werden (vgl. Kapitel 3.1 und 3.3). Die SCARA-Roboter fungiert damit als Datenquelle für den digitalen Produktzwilling. Die 3D-Integrationsplattform NVIDIA Omniverse übernimmt die zentrale Rolle bei der Verknüpfung des digitalen Modells mit den Echtzeitdaten (vgl. Kapitel 3.4). Die Modellanbindung erfolgt über den USD-Konnektor (vgl. Kapitel 3.4.2). Die Kopplungseinheit stellt über das MQTT-Protokoll die Verbindung zwischen physischem Objekt und Plattform her (vgl. Kapitel 3.3.2 und 3.4.2). Die empfangenen Daten werden im Rahmen eines Daten-Mappings strukturiert und in die

Datenorganisation von Omniverse überführt. Diese basiert auf dem USD-Format und wird über den Nucleus-Server verwaltet (vgl. Kapitel 3.4.2). Diese systematische Darstellung verdeutlicht, wie die in den theoretischen Grundlagen entwickelten Strukturelemente in der praktischen Anwendung zusammengeführt und technisch umgesetzt werden.

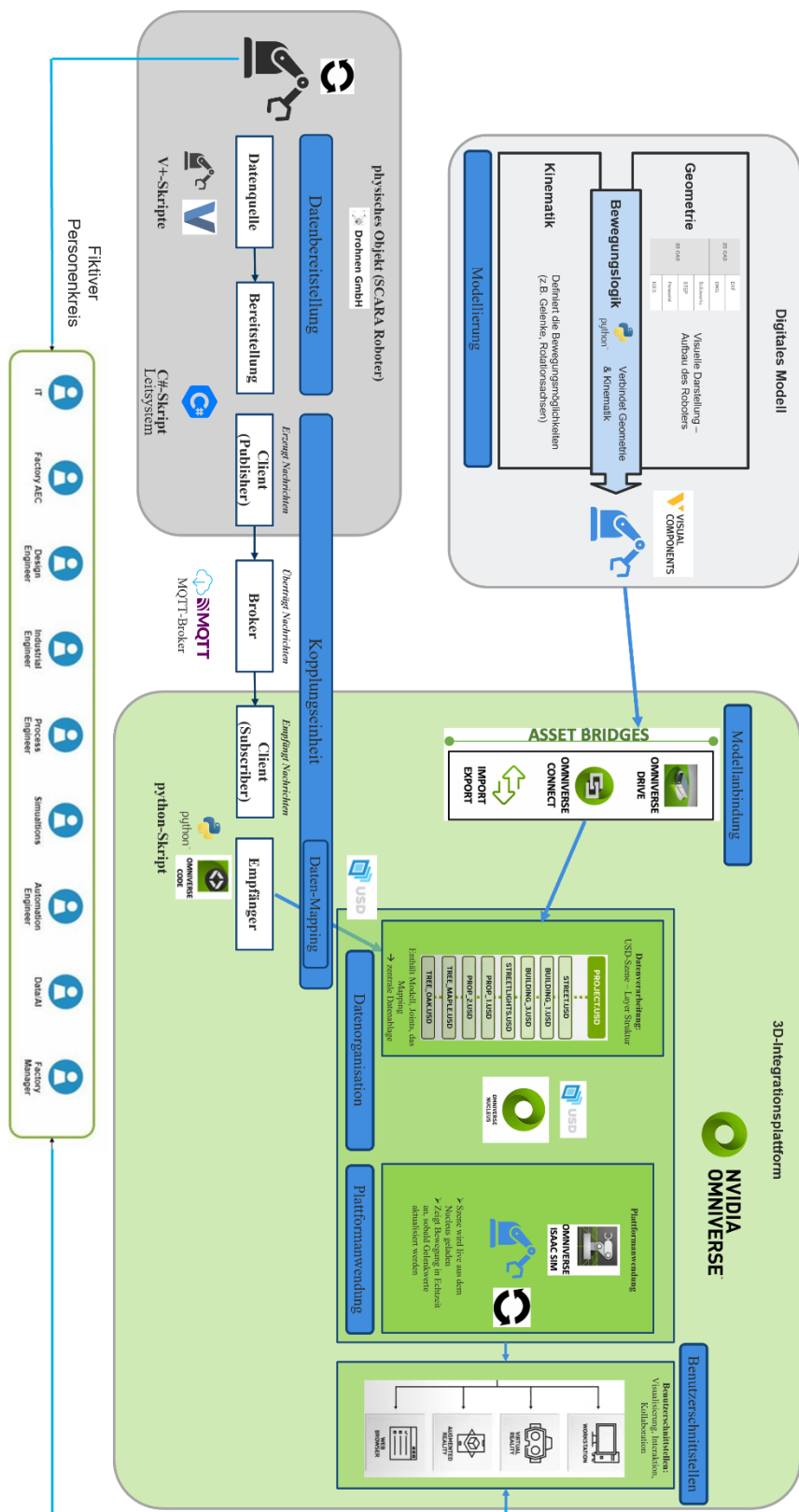


Abbildung 15: Zielsystem
Quelle: Eigene Darstellung

Im Übergang von Abbildung 15 zu Abbildung 16 zeigen sich mehrere zentrale Anpassungen: Die Kopplungseinheit verbindet nun direkt das physische Objekt mit dem digitalen Modell innerhalb von Visual Components. Die Datenorganisation erfolgt lokal auf dem Rechner, auf dem auch die Software ausgeführt wird. Als Benutzerschnittstelle dient ausschließlich die Softwareumgebung von Visual Components, eine separate Plattformanwendung entfällt.

5.3 Anwendung der Vorgehensmodells: Ausarbeitungsphase

Im Anschluss an die Zielsystementwicklung folgt die Ausarbeitungsphase, die sich, wie in Kapitel 4.4.2 beschrieben, in die Unterphasen Datenbereitstellung, Modellierung, Datenintegration und -verarbeitung sowie Einrichtung von Plattformanwendungen und Benutzerschnittstellen gliedert. Diese Phase wird gemäß den in Kapitel 5.1 dargestellten Rahmenbedingungen angepasst. Dabei erfolgt die detaillierte Ausarbeitung des in Abbildung 16 dargestellten Zielsystems.

Datenbereitstellung: In dieser Unterphase geht es darum, die für die Echtzeitvisualisierung relevanten Daten direkt an der Datenquelle, dem SCARA-Roboter zu erfassen, zu strukturieren und für den Zugriff durch die Kopplungseinheit vorzubereiten.

Wie in Kapitel 3.2.2 beschrieben, werden für eine funktionsfähige und realitätsnahe Echtzeitvisualisierung folgende Daten benötigt: die Startpositionen der vier Gelenke ($\$joint1$ bis $\$joint4$), die Zielpositionen der vier Gelenke ($\$dest1$ bis $\$dest4$), die sieben Rampenzeiten ($\$state1$ bis $\$state7$) sowie deren Summenwert ($\$state8$). Diese werden zusammenfassend als Daten X bezeichnet. Die Erfassung dieser Daten X erfolgt in einem festen Intervall von 100 Millisekunden durch das auf dem IPC Application Controller ausgeführte Skript „twin.data()“ (vgl. Kapitel 3.1.2). Dieses Skript liest die relevanten Werte aus dem Roboter aus, strukturiert sie und speichert sie in einer Variablen. Die Übertragung an das Leitsystem erfolgt über das Skript „tcp.exchange()“, dass die Inhalte dieser Variablen über das TCP/IP-Protokoll sendet. Wie in Kapitel 3.1.2 beschrieben, stehen die Daten X nach der Übertragung im Leitsystem zur weiteren Verarbeitung durch die Kopplungseinheit zur Verfügung. In der Drohnen GmbH basiert das benutzerdefinierte Leitsystem auf der Programmiersprache C# und wird auf einem lokalen Server ausgeführt.

Modellierung: Wie in Kapitel 3.2.1 beschrieben, basiert die Modellbildung in dieser Arbeit auf einem physikalischen Modellierungsansatz, bei dem reale Eigenschaften des Systems explizit abgebildet werden. Das digitale Modell wird unter Verwendung importierter CAD-Geometrien, definierter kinematischer Strukturen sowie einer bewegungsbasierten Logik aufgebaut. Der Fokus liegt dabei auf einem White-Box-Modell, das eine präzise und nachvollziehbare Repräsentation des SCARA-Roboters ermöglicht.

Die Modellbildung gliedert sich, wie in Kapitel 3.2.2 erläutert, in drei aufeinander aufbauende Phasen: Systemanalyse, Formalisierung und Implementierung. Die Informationen zur Systemanalyse stammen aus Kapitel 3.1.1, das den SCARA-Roboter i4-550L von Omron ausführlich beschreibt. In der Phase der Formalisierung erfolgt der Import der CAD-Daten sowie die Definition der Kinematik auf Basis der Herstellerangaben. Nach Abschluss dieser Phase liegt das vollständige digitale Modell mit Komponentenstruktur vor (vgl. Abbildung 17).

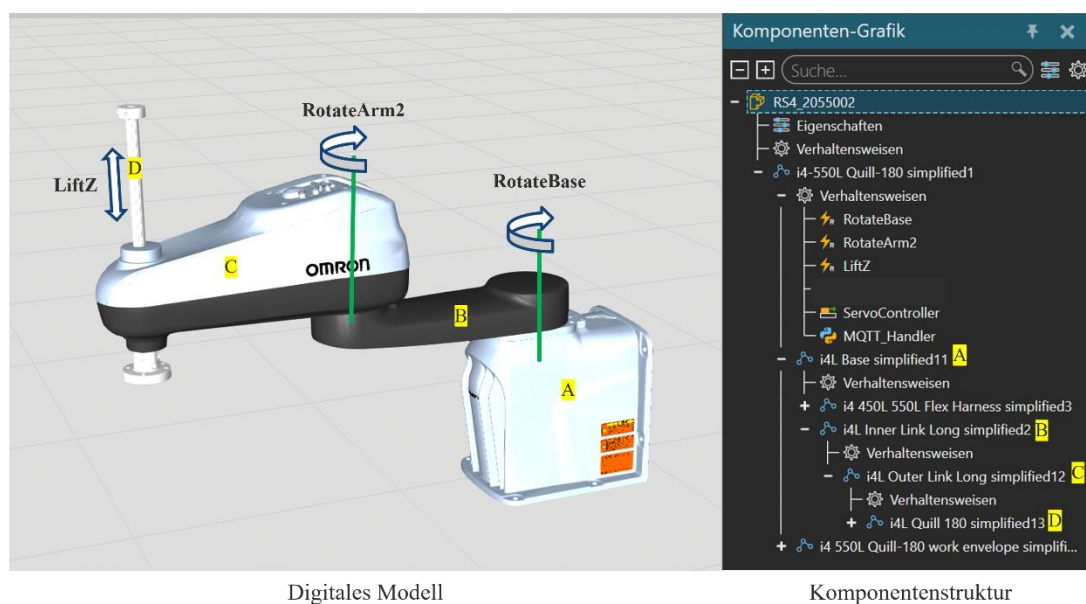


Abbildung 17: Visual Components - Digitales Modell und Komponentenstruktur

Quelle: Eigene Darstellung unter Verwendung der Software Visual Components

Die abschließende Implementierung betrifft die Bewegungslogik, durch die das Modell funktionsfähig wird. Die Bewegungsplanung erfolgt über das Python-Skript „MQTT_Handler“ (vgl. Abbildung 17). Es realisiert die in Kapitel 3.2.2 beschriebene siebenphasige S-Kurven-Interpolation. Die berechneten Werte aktualisieren in Echtzeit die Value-Signale „RotateBase“, „RotateArm2“ und „LiftZ“, die durch den Controller „ServoController“ ausgeführt werden (vgl. Abbildung 17). Die Bewegungsart ergibt sich direkt aus der zuvor definierten Kinematikstruktur.

Datenintegration und -verarbeitung: Auf Grundlage der in Kapitel 5.1 beschriebenen Rahmenbedingungen beschränkt sich diese Unterphase auf die Umsetzung der Kopplungseinheit sowie des anschließenden Daten-Mappings.

Nach der Datenbereitstellung erfolgt die Einbindung der Daten X in das digitale Modell über eine Kopplungseinheit. Ziel ist es, die Daten X aus dem physischen Objekt über ein MQTT-basiertes Kommunikationssystem an die Softwareumgebung Visual Components zu übertragen und dort in die Bewegungslogik zu integrieren (vgl. Abbildung 18).

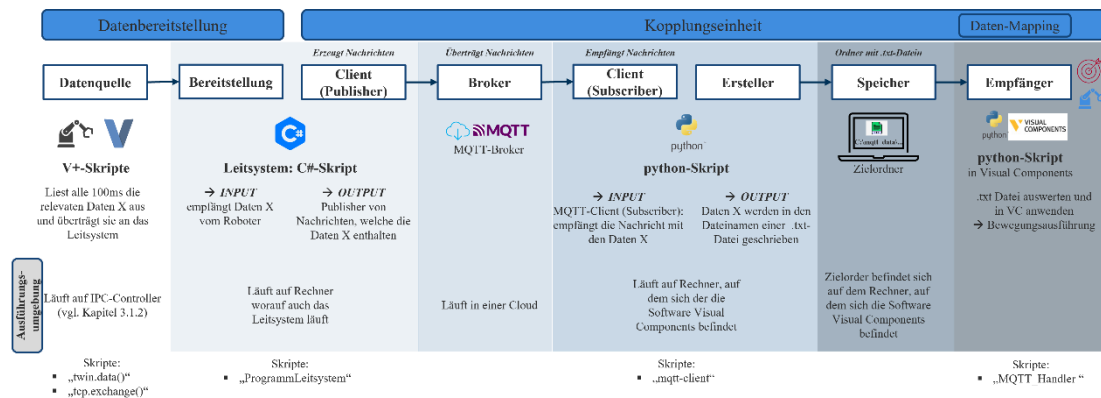


Abbildung 18: unidirektionaler Datenfluss

Quelle: Eigene Darstellung

Die Übertragung erfolgt in mehreren Schritten: Zunächst sendet der MQTT-Client (Publisher), der Teil des C#-Skripts des Leitsystems ist, die strukturierten Daten X als MQTT-Nachricht an einen MQTT-Broker (vgl. Kapitel 3.3.2). Ein auf dem Zielrechner laufendes Python-Skript („mqtt-client“) fungiert als MQTT-Client (Subscriber) und empfängt die Nachrichten. Die empfangenen Daten X werden zeilenweise in eine .txt-Datei geschrieben und in einem festgelegten Zielverzeichnis gespeichert. Jede Datei erhält zudem einen Zeitstempel im Dateinamen, sodass eindeutig erkennbar ist, welche Datei aktuell ist. Der Zielordner befindet sich auf dem Rechner, auf dem auch Visual Components installiert ist. Das Python-Skript „MQTT_Handler“ innerhalb von Visual Components (vgl. Abbildung 17) liest die jeweils aktuellste Datei aus, speichert die enthaltenen Daten X und löscht die Datei anschließend. Im Anschluss erfolgt das Daten-Mapping: Die Werte aus Daten X werden entsprechenden Variablen im Skript zugeordnet. Diese Variablen dienen als Eingangsgrößen für die S-Kurven-Interpolation, deren Ergebnis die Value-Signale im Modell in Echtzeit aktualisiert. So entsteht eine kontinuierliche Bewegungsausführung auf Basis realer Roboterbewegungen. Dieser Prozess wiederholt sich bei jeder neu eingehenden Nachricht.

Wie in Abbildung 18 dargestellt, zeigt dieser unidirektionale Datenfluss die aufeinander abgestimmten Schritte der Datenerfassung, der Kopplungseinheit und des Daten-Mappings. Dieses Zusammenspiel ermöglicht die Echtzeitvisualisierung der Bewegung des SCARA-Roboters im digitalen Modell (vgl. Abbildung 19).

Einrichtung von Plattformanwendung und Benutzerschnittstellen: In diesem Anwendungsfall erfolgt die Umsetzung dieser Unterphase in stark vereinfachter Form. Als Benutzerschnittstelle dient ausschließlich die Softwareumgebung von Visual Components, auch eine spezifische Plattformanwendung im Sinne einer dedizierten Steuer- oder Visualisierungsoberfläche entfällt. Das digitale Modell wird in den Live-Modus gesetzt und empfängt die Daten X, welche über die Kopplungseinheit übertragen werden. Abbildung 19 zeigt das digitale Modell in Visual Components in Echtzeitabgleich mit dem physischen SCARA-Roboter.

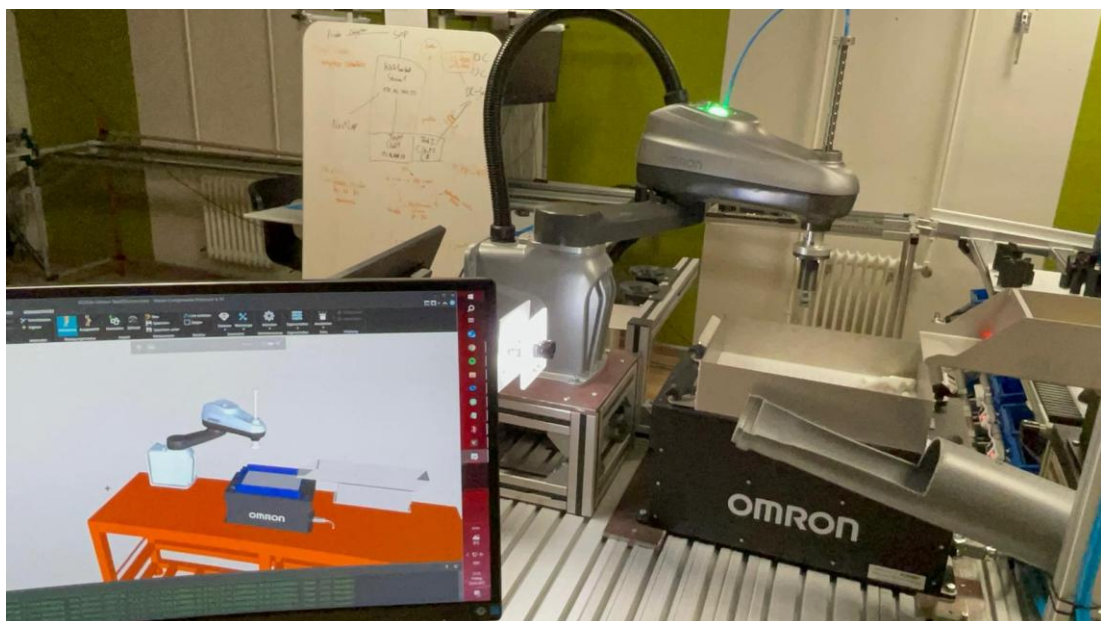


Abbildung 19: Physisches Objekt und digitales Modell im Abgleich

Quelle: Eigene Darstellung unter Verwendung der Software Visual Components

5.4 Kritische Reflexion

Das Ziel, einen digitalen Produktzwilling in Form eines Datenzwillings für den SCARA-Roboter Omron i4-550L zu erstellen, wurde erfolgreich umgesetzt. Die Bewegungen des physischen Roboters konnten in Echtzeit visuell im digitalen Modell abgebildet werden.

Dennoch zeigten sich bei der technischen Umsetzung gewisse Einschränkungen. Die Visualisierung weist minimale „Ruckler“ auf, wodurch vereinzelt Sprünge im Bewegungsablauf sichtbar sind (siehe Video, welches im Anhang bereitgestellt ist). Ursache dafür ist, dass die Bewegungsbahn in diskrete Einzelpunkte zerlegt wird, um

die Bewegung digital darzustellen. Jeder dieser Punkte beschreibt eine Soll-Position eines Gelenks zu einem bestimmten Zeitpunkt. Je enger die zeitlichen Abstände zwischen den Punkten, desto flüssiger und realistischer erscheint die Animation. Die Berechnung dieser Bewegungspositionen erfolgt innerhalb des Python-Skripts „MQTT_Handler“ in Visual Components (vgl. Abbildung 17), das in einer Schleifenlogik zyklisch neue Sollwerte berechnet. Dabei gilt: Je häufiger die Schleife pro Sekunde ausgeführt wird, desto höher die Bewegungstreue. In der Praxis ließ sich jedoch ein Takt von 0,01 s technisch nicht stabil realisieren, weshalb ein Intervall von 0,05–0,075 s gewählt werden musste. Ein weiterer begrenzender Faktor war die fehlende native MQTT-Unterstützung in Visual Components. Stattdessen wurde ein Behelfslösung über .txt-Dateien implementiert, die regelmäßig geschrieben und anschließend ausgelesen werden. Dadurch entsteht eine zusätzliche zeitliche Verzögerung bei der Datenverarbeitung. Zudem bringt der Einsatz eines kostenlosen MQTT-Brokers Einschränkungen hinsichtlich der verfügbaren Datenrate mit sich. Nicht zuletzt erfordert die gleichzeitige Datenverarbeitung und 3D-Visualisierung eine hohe Rechenleistung, um eine durchgehend flüssige Darstellung zu gewährleisten. Alle diese Aspekte beeinflussen die Qualität der Bewegungsausführung im digitalen Modell merklich und zeigen technische Grenzen der aktuellen Lösung auf.

Trotz dieser technischen Herausforderungen hat sich gezeigt, dass das zugrunde liegende Vorgehensmodell einen stabilen Orientierungsrahmen bietet, um solche komplexen Umsetzungsprozesse methodisch zu strukturieren und zielgerichtet voranzutreiben. Besonders geeignet ist es für Personen mit Vorkenntnissen in Simulation, Robotik und IIoT-Kommunikation. Es hilft dabei, anspruchsvolle Aufgaben in einzelne Schritte aufzuteilen und gezielt zu bearbeiten. Gleichzeitig wird deutlich, dass das Modell nicht alle technischen Details vorgibt. Für die Umsetzung braucht es daher entweder Fachwissen aus verschiedenen Bereichen oder ein gut aufgestelltes interdisziplinäres Team.

6. Fazit und Ausblick

Im Rahmen dieser Arbeit wurde ein strukturiertes Vorgehensmodell zur Entwicklung eines digitalen Produktzwillings im industriellen Metaverse entwickelt. Dieses methodische Rahmenwerk berücksichtigt sowohl die Anforderungen digitaler Zwillinge als auch die spezifischen Eigenschaften des industriellen Metaverse. Im Mittelpunkt stehen eine qualitativ hochwertige Datengrundlage, ein stabiler Datenfluss und die realitätsgetreue Abbildung physischer Objekte im digitalen Modell. Zentrale Anforderungen wie Interoperabilität, Echtzeitfähigkeit und Integrationsfähigkeit werden systematisch berücksichtigt, um eine durchgängige Kommunikation und Zusammenarbeit zwischen verschiedenen Systemen und Plattformen zu ermöglichen. Die Validierung des Vorgehensmodells erfolgte exemplarisch anhand eines Anwendungsfalls bei der Drohnen GmbH. Aufgrund lizenzrechtlicher Einschränkungen konnte die vollständige Integration in eine Metaverse-Plattform nicht praktisch realisiert, jedoch konzeptionell ausgearbeitet werden. In der praktischen Umsetzung gelang die Entwicklung eines funktionalen Datenzwillings (als erste Entwicklungsstufe des digitalen Produktzwillings) des Omron i4L SCARA-Roboters. Dabei wurden die Phasen der Modellbildung, Datenerfassung, Datenbereitstellung sowie die Echtzeitübertragung und -visualisierung erfolgreich umgesetzt.

Die Anwendung zeigt, dass das entwickelte Vorgehensmodell in seiner derzeitigen Ausprägung die Erstellung eines funktionalen Datenzwillings ermöglicht. Perspektivisch eröffnet es das Potenzial, nicht nur einzelne Produkte wie den SCARA-Roboter abzubilden, sondern auch komplexere Systeme, etwa eine vollständige Pick-and-Place-Zelle (vgl. Abbildung 7) oder ganze Prozesse wie komplette Montagelinien digital zu repräsentieren (vgl. Kapitel 2.1.2). Dabei ist zu beachten, dass digitale Zwillinge verschiedene funktionale Entwicklungsstufen durchlaufen können (vgl. Kapitel 2.2.2). In dieser Arbeit wurde mit dem Datenzwilling die erste Stufe realisiert. Zukünftige Erweiterungen könnten die Entwicklung eines Simulationszwillings ermöglichen, der auf Basis der erfassten Daten Analysen durchführt und Prognosen erstellt. Langfristig erscheint auch die Umsetzung eines Steuerungszwillings denkbar, der mithilfe Künstlicher Intelligenz autonome Entscheidungen trifft und aktiv in das physische System eingreift.

Das vorgestellte Vorgehensmodell bietet eine fundierte Grundlage, die auf den dargestellten Aspekten weiterentwickelt werden kann. Von der datengetriebenen Visualisierung einzelner Produktionsmaschinen bis hin zur autonomen, selbststeuernden Fabrik.

Literaturverzeichnis:

Angerer, Horst u. a. (2023): Industrial Metaverse: Use Cases, Mehrwerte und Potenziale für den Wirtschaftsstandort Deutschland. Bitkom e. V. URL: <https://www.bitkom.org/sites/main/files/2023-09/bitkom-leitfaden-industrial-metaverse.pdf> (01.08.2024).

Appelfeller, Wieland/Feldmann, Carsten (2023): Die digitale Transformation des Unternehmens: systematischer Leitfaden mit zehn Elementen zur Strukturierung und Reifegradmessung. 2. Auflage. Berlin [Heidelberg]: Springer Gabler.

Ariansyah, Dedy u. a. (2020): Digital Twin Development: A Step by Step Guideline. Rochester, NY: Social Science Research Network. DOI: 10.2139/ssrn.3717726.

Babel, Wolfgang (2023): Internet of Things und Industrie 4.0. 1st ed. Auflage. Wiesbaden: Springer Vieweg. in Springer Fachmedien Wiesbaden GmbH.

Babel, Wolfgang (2024): Systemintegration in Industrie 4.0 und IoT: Vom Ethernet bis hin zum Internet und OPC UA. Wiesbaden: Springer Vieweg.

Barbosa, William S. u. a. (2020): Industry 4.0: examples of the use of the robotic arm for digital manufacturing processes. In: International Journal on Interactive Design and Manufacturing (IJIDeM), Jg. 14 (4), S. 1569–1575. DOI: 10.1007/s12008-020-00714-4.

Buchholz, Peter (2006): Modellierung und Simulation diskreter und kontinuierlicher Systeme. Universität Dortmund. URL: https://ls4-www.cs.tu-dortmund.de/Lehre/06-42279/Folien/MS_1.pdf (12.04.2025).

Bundesverband der Deutschen Industrie e. V. (2023): Das industrielle Metaverse – Chancen für die Industrie. URL: <https://bdi.eu/artikel/news/das-industrielle-metaverse-chancen-fuer-die-industrie> (07.08.2024).

Caulfield, Brian (2021): What is the Metaverse? URL: <https://34.214.249.23.nip.io/blog/what-is-the-metaverse/> (10.02.2025).

Chapman, Pete u. a. (1999): CRISP-DM 1.0: Step-by-step Data Mining Guide. CRISP-DM Consortium. URL: <https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf> (01.03.2025).

Chen, Guangfeng u. a. (2012): SCARA Robot Control System Design and Trajectory Planning: A Case Study. Berlin, Heidelberg: Springer. DOI: 10.1007/978-3-642-27951-5_26.

Crespi, Noel/Drobot, Adam T./Minverva, Roberto (2023): The Digital Twin: What and Why? In: The digital twin. Cham, Switzerland: Springer International Publishing AG.

Ehmer, Mohd/Khan, Farmeena (2012): A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. In: International Journal of Advanced Computer Science and Applications, Jg. 3. DOI: 10.14569/IJACSA.2012.030603.

EMQX Team (2025): Mastering MQTT: The Ultimate Beginner's Guide for 2025. URL: <https://www.emqx.com/en/blog/the-easiest-guide-to-getting-started-with-mqtt> (05.05.2025).

Fang, Yi u. a. (2019): Smooth and time-optimal S-curve trajectory planning for automated robots and machines. In: Mechanism and Machine Theory, Jg. 137, S. 127–153. DOI: 10.1016/j.mechmachtheory.2019.03.019.

Follath, Alexander/Bross, Florian/Galka, Stefan (2022): Vorgehensmodell zur Erstellung Digitaler Zwillinge für Produktion und Logistik. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb, Jg. 117 (10), S. 691–696. DOI: 10.1515/zwf-2022-1127.

Grieves, Michael (2015): Digital Twin: Manufacturing Excellence through Virtual Factory Replication. URL: https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication (20.03.2024).

Gronau, Norbert (2018): Industrial Internet of Things - Protokolle und Standards. Universität Potsdam. URL: [https://subcom.wi.uni-potsdam.de/homepage/lehrewi.nsf/0/14d1a569b64f80f7c12583dd00516f8d/\\$FILE/06_Protokolle_und_Standards.pdf](https://subcom.wi.uni-potsdam.de/homepage/lehrewi.nsf/0/14d1a569b64f80f7c12583dd00516f8d/$FILE/06_Protokolle_und_Standards.pdf) (20.04.2025).

Gutenschwager, Kai u. a. (2017): Simulation in Produktion und Logistik: Grundlagen und Anwendungen. Berlin; [Heidelberg]: Springer Vieweg.

Hochschule Neu-Ulm (o. J.): Technologie-Transfer-Zentrum Leipheim - HNU. URL: <https://www.hnu.de/forschung/forschungs-und-transfereinrichtungen/technologie-transfer-zentrum-leipheim> (12.05.2025).

Horne, Bill/Jamshidi, M./Vadiee, Nader (1990): Neural networks in robotics: A survey. In: Journal of Intelligent and Robotic Systems, Jg. 3 (1), S. 51–66. DOI: 10.1007/BF00368972.

Huber, Steffen u. a. (2019): DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. In: Procedia CIRP, Jg. 79, S. 403–408. DOI: 10.1016/j.procir.2019.02.106.

Jakoby, Walter (2021): Projektmanagement für Ingenieure: ein praxisnahes Lehrbuch für den systematischen Projekterfolg. 5., überarbeitete und aktualisierte Auflage. Auflage. Wiesbaden [Heidelberg]: Springer Vieweg.

Jenny, Bruno (2014): Projektmanagement: das Wissen für den Profi. 3., vollständig überarb. und aktual. Aufl. Auflage. Zürich: vdf, Hochschulverl. an der ETH Zürich.

Kaur, Maninder J./Mishra, Ved P./Maheshwari, Piyush (2020): The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action. In: Digital twin technologies and smart cities. Cham: Springer, S. 212.

König, Dimitri (2024): Trajektorien Daten. URL: <https://scisimple.com/de/keywords/trajektorien-daten--k-kegowp> (04.05.2025).

Kreutzer, Ralf T./Klose, Sonja (2023): Metaverse kompakt: Begriffe, Konzepte, Handlungsoptionen. Wiesbaden [Heidelberg]: Springer Gabler.

Kritzinger, Werner u. a. (2018): Digital Twin in manufacturing: A categorical literature review and classification. In: IFAC-PapersOnLine, Jg. 51 (11), S. 1016–1022. DOI: 10.1016/j.ifacol.2018.08.474.

Langlotz, Pascal u. a. (2020): Modelle als Grundlage für den Digitalen Zwilling: Fusion von physikalischen und datengetriebenen Modellen. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb, Jg. 115 (5), S. 340–343. DOI: 10.3139/104.112281.

Lindemann, Udo (2009): Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden. 3., korrigierte Aufl. Auflage. Dordrecht Heidelberg: Springer.

Maier, Prof Dr Günter W. (o. J.): Definition: Synergie. URL: <https://wirtschaftslexikon.gabler.de/definition/synergie-47512> (22.04.2025).

Matysczok, Carsten/Hohlweg, Nils (2021): Digitaler Zwilling. UNITY Consulting & Innovation. URL: <https://www.unity-consulting.com/de/news/opportunity-digitaler-zwilling/> (10.02.2024).

Mockenhaupt, Andreas/Rüttgers, Jürgen (2021): Digitalisierung und Künstliche Intelligenz in der Produktion: Grundlagen und Anwendung. 1. Auflage. Wiesbaden [Heidelberg]: Springer Vieweg. DOI: 10.1007/978-3-658-32773-6.

NVIDIA (2023): Architecture - Omniverse Nucleus. URL: <https://docs.omniverse.nvidia.com/nucleus/latest/architecture.html> (06.05.2025).

NVIDIA (2025a): Collaborating and Sharing - Omniverse Digital Twins. URL: <https://docs.omniverse.nvidia.com/digital-twins/latest/building-full-fidelity-viz/collaborating-sharing.html> (07.05.2025).

NVIDIA (2025b): Data IoT - Reference architecture diagrams for Omniverse. URL: <https://docs.omniverse.nvidia.com/arch-diagrams/latest/ref-arch-diagrams/factory-dt-diagrams/data.html> (06.05.2025).

NVIDIA (2025c): Delivery - Reference architecture diagrams for Omniverse. URL: <https://docs.omniverse.nvidia.com/arch-diagrams/latest/ref-arch-diagrams/factory-dt-diagrams/delivery.html> (06.05.2025).

NVIDIA (2025d): Factory Digital Twin Reference Architecture - Reference architecture diagrams for Omniverse. URL: <https://docs.omniverse.nvidia.com/arch-diagrams/latest/ref-arch-diagrams/factory-dt-diagram.html> (07.05.2025).

NVIDIA (2025e): Geometry Creation - Reference architecture diagrams for Omniverse. URL: <https://docs.omniverse.nvidia.com/arch-diagrams/latest/ref-arch-diagrams/factory-dt-diagrams/geometry-creation.html> (06.05.2025).

NVIDIA (2025f): Omniverse Platform - Reference architecture diagrams for Omniverse. URL: <https://docs.omniverse.nvidia.com/arch-diagrams/latest/ref-arch-diagrams/factory-dt-diagrams/ov-platform.html> (06.05.2025).

NVIDIA (2025g): OpenUSD Overview - Omniverse USD. URL: <https://docs.omniverse.nvidia.com/usd/latest/index.html> (06.05.2025).

NVIDIA (2025h): Platform Overview - Omniverse Developer Overview. URL: <https://docs.omniverse.nvidia.com/dev-overview/latest/platform-overview.html> (06.05.2025).

NVIDIA (2025i): USD - Omniverse Digital Twins. URL: https://docs.omniverse.nvidia.com/digital-twins/latest/building-full-fidelity-viz/usd.html?utm_source=chatgpt.com (06.05.2025).

NVIDIA (2025j): USD Connections Overview - Omniverse Connect. URL: <https://docs.omniverse.nvidia.com/connect/latest/overview.html> (06.05.2025).

Omron (2020): Automation Control Environment (ACE) Version 4 User's Manual. Omron Robotics and Safety Technologies, Inc. URL: https://files.omron.eu/downloads/latest/manual/en/i633_ace_4.0_users_manual_en.pdf?v=2 (01.03.2025).

Omron (2023a): i4L Robots User's Manual. Pleasanton, CA, USA: Omron Robotics and Safety Technologies, Inc. URL: https://files.omron.eu/downloads/latest/manual/en/i658_i4l_series_users_manual_en.pdf?v=2 (01.03.2025).

Omron (2023b): iPF-Series Flexible Part Feeder User's Manual. Omron Robotics and Safety Technologies, Inc. URL: https://files.omron.eu/downloads/latest/manual/en/i684_flexible_part_feeder_users_manual_en.pdf?v=1 (01.03.2025).

Pixar (2021): Introduction to USD — Universal Scene Description 25.05 documentation. URL: <https://openusd.org/release/intro.html> (06.05.2025).

Rabe, Markus/Spieckermann, Sven/Wenzel, Sigrid (2008): Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken. 1. Aufl. Auflage. Berlin Heidelberg: Springer.

Shariatee, Morteza u. a. (2014): Design of an economical SCARA robot for industrial applications. DOI: 10.1109/ICRoM.2014.6990957.

- Siemens (o. J.): Factsheet: Industrial Metaverse. URL: <https://assets.new.siemens.com/siemens/assets/api/uuid:7c4be71d-72ea-4958-aef4-0915a42f37f3/Factsheet-Was-ist-das-Industrial-Metaverse.pdf> (01.08.2024).
- Stark, Reiner/Kind, Simon/Neumeyer, Sebastian (2017): Innovations in Digital Modelling for Next Generation Manufacturing System Design. In: CIRP Annals - Manufacturing Technology, Jg. 66 (1), S. 169–172.
- Stiebitz, Mattis (2022): Potenziale des Digitalen Zwillings im Produktlebenszyklus. In: Smart Services. 1. Auflage. Wiesbaden [Heidelberg]: Springer Fachmedien, S. 571.
- Stockhausen, Joline/Lambertz, Lamar (2024): Digitaler Zwilling - Aufbau und effektives Datenmanagement. KPMG AG Wirtschaftsprüfungsgesellschaft. URL: <https://hub.kpmg.de/de/whitepaper-digitaler-zwilling> (15.08.2025).
- Tao, Fei/Zhang, Meng/Nee, Andrew Yeh Chris (2019): Digital twin driven smart manufacturing. London: Academic press.
- Timinger, Holger (2017): Modernes Projektmanagement: mit traditionellem, agilem und hybridem Vorgehen zum Erfolg. 1. Auflage. Auflage. Weinheim: Wiley.
- Tu, Xinyi u. a. (2024): Architecture for data-centric and semantic-enhanced industrial metaverse: Bridging physical factories and virtual landscape. In: Journal of Manufacturing Systems, Jg. 74, S. 965–979. DOI: 10.1016/j.jmsy.2024.05.016.
- Ulbrich, Peter (2017): Echtzeitsysteme Einleitung. Lehrstuhl für Verteilte Systeme und Betriebssysteme. URL: https://www4.cs.fau.de/Lehre/WS17/V_EZS/Skript/02-Einleitung_a4.pdf (15.04.2025).
- Ullrich, Marco u. a. (2024): Eine Referenzarchitektur für das Industrial Metaverse. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb, Jg. 119 (3), S. 186–191. DOI: 10.1515/zwf-2024-1033.
- Visual Components (2018): Python Robotics - Programming a Robot with Python. URL: <https://academy.visualcomponents.com/courses/python-robotics-programming-a-robot-with-python/> (04.05.2025).
- Visual Components (o. J.): Roboter Programmierung. URL: <https://www.visualcomponents.com/de/anwendungsfaelle/roboter-programmierung/> (04.05.2025).
- Waxer, Cindy (2023): Spielfeld der Zukunft: Das industrielle Metaverse. MIT Technology Review Insights. URL: <https://www.siemens.com/de/de/unternehmen/insights/industrial-metaverse-spielfeld-der-zukunft.html> (01.08.2024).

Wiemer, Hajo/Drowatzky, Lucas/Ihlenfeldt, Steffen (2019): Data Mining Methodology for Engineering Applications (DMME)—A Holistic Extension to the CRISP-DM Model. In: Applied Sciences, Jg. 9 (12), S. 2407. DOI: 10.3390/app9122407.

Zauner, Martin/Schrempf, Andreas (2009): Modellbildung und Simulation. In: Zauner, Martin/Schrempf, Andreas (Hrsg.): Informatik in der Medizintechnik: Grundlagen — Software — Computergestützte Systeme. Vienna: Springer, S. 223–247. DOI: 10.1007/978-3-211-89189-6_8.

Zheng, Yu/Yang, Sen/Cheng, Huanchong (2019): An application framework of digital twin and its case study. In: Journal of Ambient Intelligence and Humanized Computing, Jg. 10 (3), S. 1141–1153.

Zheng, Zhiming u. a. (2022): Industrial Metaverse: Connotation, Features, Technologies, Applications and Challenges. In: Methods and Applications for Modeling and Simulation of Complex Systems: 21st Asia Simulation Conference, AsiaSim 2022, Changsha, China, December 9-11, 2022, Proceedings, Part I. 1. Auflage. Singapore: Springer, S. 239–263.

Abbildungsverzeichnis:

Abbildung 1: Modell des digitalen Produktzwillings.....	7
Abbildung 2: Referenzarchitektur eines industriellen Metaverse.....	8
Abbildung 3: Möglichkeiten der Integration von digitalen Modellen in die 3D- Integrationsplattform.....	9
Abbildung 4: Unterscheidung digitaler Produktzwillings anhand der Art ihrer Anwendung	11
Abbildung 5: Systemstruktur - Digitaler Produktzwillings im industriellen Metaverse	11
Abbildung 6: Mechanischer Aufbau und Freiheitsgrade	14
Abbildung 7: vereinfachter Aufbau einer Pick-and-Place-Zelle	16
Abbildung 8: Bewegungsverlauf einer S-Kurve	19
Abbildung 9: Technologiezuordnung im OSI-Modell zur IIoT-basierten Datenübertragung.....	23
Abbildung 10: Publisher/Subscriber Modell.....	23
Abbildung 11: Vorgehensmodell 1 nach Follath Bross und Galka	32
Abbildung 12: Vorgehensmodell 2 nach Wiemer, Drowatzky und Ihlenfeldt	33
Abbildung 13: Aufbau des Vorgehensmodells	37
Abbildung 14: Vorgehensmodell zur Erstellung eines digitalen Produktzwillings im industriellen Metaverse	43
Abbildung 15: Zielsystem.....	46
Abbildung 16: Zielsystem (angepasst an die Rahmenbedingungen)	47
Abbildung 17: Visual Components - Digitales Modell und Komponentenstruktur...	49
Abbildung 18: unidirektionaler Datenfluss	50
Abbildung 19: Physisches Objekt und digitales Modell im Abgleich	51