

**Forschungsbericht
Institut für Automatisierungstechnik
und Softwaresysteme**

Hrsg.: Prof. Dr.-Ing. Dr. h. c. Michael Weyrich

Dominik Braun

**Automatisierte Erstellung
der Relationen eines Digitalen
Zwillings für Bestandssysteme
in der Produktion**

Band 3/2025

Universität Stuttgart

Automatisierte Erstellung der Relationen eines Digitalen Zwillings für Bestandssysteme in der Produktion

Von der Graduate School of Excellence
advanced Manufacturing Engineering der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von
Dominik Braun
aus Heilbronn

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. Michael Weyrich
Mitberichter:	Prof. Dr. Hans-Georg Kemper
Tag der Einreichung:	05.04.2024
Tag der mündlichen Prüfung:	28.11.2024

Institut für Automatisierungstechnik und Softwaresysteme
der Universität Stuttgart

2024

IAS-Forschungsberichte

Band 3/2025

Dominik Braun

**Automatisierte Erstellung der Relationen
eines Digitalen Zwillings für
Bestandssysteme in der Produktion**

D 93 (Diss. Universität Stuttgart)

Shaker Verlag
Düren 2025

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Stuttgart, Univ., Diss., 2024

Copyright Shaker Verlag 2025

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

Print-ISBN 978-3-8440-9774-0
PDF-ISBN 978-3-8440-9869-3
ISSN 1610-4781
eISSN 2944-8417
<https://doi.org/10.2370/9783844098693>

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Telefon: 02421 / 99 0 11 - 0 • Telefax: 02421 / 99 0 11 - 9
Internet: www.shaker.de • E-Mail: info@shaker.de

Die vorliegende Arbeit entstand während meiner Tätigkeit als Doktorand der Graduate School of Excellence advanced Manufacturing Engineering (GSaME) der Universität Stuttgart am Institut für Automatisierungstechnik und Softwaresysteme (IAS) und einem Unternehmen der Automatisierungstechnik.

Mein besonderer Dank gilt meinem Doktorvater und Leiter des Instituts, Herrn Prof. Dr.-Ing. Dr. h. c. Michael Weyrich, für die Übernahme des Hauptberichts, die vielen konstruktiven Diskussionen und wertvollen Hinweise und Anregungen bei der Erstellung dieser Arbeit.

Herrn Prof. Dr. Hans-Georg Kemper danke ich für das Interesse an meiner Arbeit, die Mitarbeit im Thesis-Committee, die freundlichen Diskussionen und die Übernahme des Mitberichts.

Darüber hinaus möchte ich mich bei Herrn Dr.-Ing. Wolfgang Schlögl maßgeblich bedanken, der mich mit der Aufgabenstellung betraute, mich kontinuierlich unterstützte und mir Türen öffnete, um mein Thema voranzubringen. Ebenso danke ich Uwe Scheiblhuber und allen anderen Personen in diesem Umfeld, die mir als wichtige Diskussionspartner und Unterstützer zur Seite gestanden sind.

Mein Dank gilt außerdem dem wissenschaftlichen Umfeld der GSaME für die vielen Impulse abseits meines Forschungsthemas und die ermöglichten Fortbildungen. Stellvertretend für alle Personen möchte ich mich hier bei Herrn Hans-Friedrich Jacobi bedanken, der mich vom Beginn meiner Promotion begleitet hat.

Ein besonderer Dank gilt Herrn Dr.-Ing. Nasser Jazdi für seine fortwährende Unterstützung, die vielen wertvollen Gespräche und Ratschläge während der Arbeit. Des Weiteren gilt mein herzlichster Dank allen Kolleginnen und Kollegen am IAS für die gute und freundschaftliche Zusammenarbeit, die vielen Diskussionen und bedingungslose Unterstützung. Ebenfalls möchte ich allen Studierenden danken, die im Rahmen ihrer Master-, Forschungs- und Bachelorarbeiten einen Beitrag zum Erfolg dieser Arbeit geleistet haben.

Abschließend möchte ich meiner Familie und meinen Freunden danken, für ihre geduldige Unterstützung und die kontinuierliche Motivation bis zur Fertigstellung dieser Arbeit.

Stuttgart, im Dezember 2024

Dominik Braun

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
Tabellenverzeichnis	v
Abkürzungsverzeichnis	vi
Begriffsverzeichnis	viii
Zusammenfassung	x
Abstract	xi
1 Einleitung	1
1.1 Ausgangssituation in der diskreten Produktion	1
1.2 Industrielle Problemstellung.....	3
1.3 Herausforderungen bei der Erstellung eines Digitalen Zwillings für Brownfield- Produktionssysteme	4
1.4 Zielsetzung dieser Forschung	6
1.5 Anforderungen an die Konzeption.....	7
1.6 Aufbau der Arbeit.....	9
2 Grundlagen	11
2.1 Ausgangssituation: Automatisierte Produktionssysteme.....	11
2.1.1 Produktionssysteme	13
2.1.2 Automatisierungssysteme	13
2.1.3 Digitaler Entwicklungsprozess automatisierter Produktionssysteme – MBSE.....	15
2.1.4 Bestehende Produktionssysteme – Brownfield-Entwicklung.....	16
2.2 Zielvorstellung: Digitaler Zwilling.....	17
2.2.1 Entstehung des Konzepts „Digitaler Zwilling“	17
2.2.2 Definition, Bestandteile und Architektur eines Digitalen Zwillings	18
2.2.3 Relationen des Digitalen Zwillings.....	22
2.2.4 Anwendungsfälle des Digitalen Zwillings und deren Mehrwerte sowie Herausforderungen.....	23
2.3 Zusammenfassung der Ausgangssituation und der Zielvorstellung	24
3 Stand der Wissenschaft und Technik	25
3.1 Ansätze zur Erfassung und Modellierung bestehender Produktionssysteme	26
3.1.1 Optische Verfahren	26
3.1.2 Dokumentenbasierte Verfahren	27
3.1.3 Netzwerkbasierte Verfahren	27
3.1.4 SPS-Codebasierte Verfahren	28
3.1.5 Sonstige Ansätze ohne eindeutige Informationsquelle.....	29

3.2	Ansätze zur Verknüpfung von Modellen verschiedener Domänen.....	30
3.2.1	Ansätze zur Erstellung der Relationen eines Digitalen Zwillings.....	30
3.2.2	Formate zur Abbildung der Relationen eines Digitalen Zwillings.....	30
3.3	Zusammenfassung und Bewertung der bestehenden Ansätze.....	32
3.4	Aufzeigen der wissenschaftlichen Forschungslücke.....	36
4	Methodik zur automatisierten Erstellung der Relationen eines Digitalen Zwillings....	37
4.1	Hinführung und grundlegende Entscheidungen zur Erstellung der Methodik.....	37
4.2	Überblick über die Methodenbausteine.....	41
4.3	Regelbasierte Analyseverfahren der SPS-Codestruktur zur Extraktion funktionaler Relationen.....	43
4.3.1	Import des nativen SPS-Codes als Informationsquelle in ein herstellerunabhängiges Informationsmodell.....	45
4.3.2	Anreicherungs- und Analyseregeln der Analyseverfahren.....	47
4.3.3	Zusammenfassung der Analyseergebnisse.....	52
4.4	Datenbasierte Analyseverfahren der Positionsdaten zur Identifikation räumlicher Relationen.....	53
4.4.1	Erfassung und Speicherung der Positionsdaten als Informationsquelle.....	53
4.4.2	Analyse der Positionen zur Wissensextraktion.....	56
4.4.3	Zusammenfassung der Analyseergebnisse.....	59
4.5	Korrelationsbasierte Analyseverfahren von Betriebs- und Positionsdaten zur Berechnung der Peripheriepositionen.....	59
4.5.1	Erfassung und Speicherung der Betriebsdaten als Informationsquelle.....	60
4.5.2	Analyseverfahren.....	61
4.5.3	Zusammenfassung der Analyseergebnisse.....	64
4.6	Methodensynchronisation: Wissensmanagement.....	64
4.6.1	Wissenszusammenführung.....	65
4.6.2	Wissensspeicherung.....	66
4.6.3	Semantische Anreicherung mit Ontologien.....	68
4.6.4	Identifikation repetitiver Strukturen und Kennzeichnung als Template.....	70
4.7	Überführung in einen Digitalen Zwilling.....	73
5	Realisierung der entwickelten Methodik als prototypisches Software- Assistenzsystem.....	75
5.1	Gesamtüberblick über die Softwaremodule.....	75
5.2	Softwaremodul für die Analyse des SPS-Codes.....	78
5.2.1	Adapter zum Import des nativen SPS-Codes.....	78
5.2.2	Implementierung der regelbasierten Strukturanalyse.....	82
5.3	Analysemodul für Positions- und E/A-Daten.....	84
5.3.1	Zeitreihendatenbank zur Speicherung der Daten.....	84
5.3.2	Implementierung der datengetriebenen Analysen.....	86
5.4	Modul zur Wissensspeicherung und Extraktion in einem Graphen.....	87
5.4.1	Wissensspeicherung in einer Graphdatenbank.....	87
5.4.2	Generierung von Templates durch die Identifikation von repetitiven Strukturen.....	89

5.5	Exportmodul aus dem Wissensgraphen.....	92
6	Evaluierung.....	94
6.1	Evaluierungskonzept.....	94
6.2	Erfüllung der Anforderungen durch die Methodik.....	95
6.3	Evaluierung der Mehrwerte der Methodik	97
6.3.1	Brownfield-Produktionssystem zur Demonstration und Evaluierung der Methodik.....	97
6.3.2	Erstellung des Digitalen Zwillings für das intelligente Lager mithilfe der entwickelten Methodik	101
6.3.3	Erfassung des Ist-Zustands der manuellen Erstellung des Digitalen Zwillings durch eine Expertenbefragung.....	102
6.3.4	Diskussion und Bewertung der Evaluierungsergebnisse	105
6.4	Übertragbarkeit der Methodik auf andere Brownfield-Produktionssysteme.....	107
6.4.1	Beispielhafter Übertrag auf das intelligente Lager	107
6.4.2	Randbedingungen der Methodik.....	108
6.4.3	Ableich mit den Eigenschaften von Brownfield-Produktionssystemen	110
6.4.4	Bewertung der Übertragbarkeit	112
7	Schlussbetrachtung	114
7.1	Zusammenfassung der Ergebnisse.....	114
7.2	Ausblick.....	116
	Literaturverzeichnis.....	118
	Anhang	130

Abbildungsverzeichnis

Abbildung 1.1: Aufbau der Arbeit	10
Abbildung 2.1: Schematische Darstellung eines automatisierten Produktionssystems und dessen Ein- und Ausgangsgrößen nach [49].....	12
Abbildung 2.2: Aufteilung der Steueraufgaben in der Automatisierungspyramide nach [53]	14
Abbildung 2.3: Komponenten des Digitalen Zwillings nach [4]	20
Abbildung 2.4: Architektur eines (intelligenten) Digitalen Zwillings nach [4].....	21
Abbildung 2.5: Relationsarten am Beispiel des intelligenten Lagers	22
Abbildung 4.1: Das Konzept – Überblick über die Methodik (blauhinterlegte Bereiche)	41
Abbildung 4.2: Importschritte	47
Abbildung 4.3: Klassendiagramm des herstellerunabhängigen Informationsmodells des SPS-Codes	48
Abbildung 4.4: Generalisierter Aufbau eines SPS-Codes.....	50
Abbildung 4.5: Beispiel eines Analyseergebnisses der SPS-Code-Analyse.....	52
Abbildung 4.6: Schemata der Positionserfassung und Speicherung	55
Abbildung 4.7 oben: unterschiedliche Gruppierung der Positionen durch zwei Klassen von Algorithmen; unten: die erwünschte Gruppierung mit Bewegungssequenzen...	57
Abbildung 4.8: Analyseschritte zur Bestimmung der Position der Peripherie	61
Abbildung 4.9: Grundstruktur des extrahierten Wissens mit der ersten Analysemethode	65
Abbildung 4.10: Hybride, hierarchische Wissensspeicherung.....	69
Abbildung 4.11: ODP mechatronischer Systeme basierend auf der VDI 2206/3682 in Protegé.	69
Abbildung 4.12: Identifikation repetitiver Strukturen und Kennzeichnung als Templates	71
Abbildung 5.1: Überblick über die Softwaremodule des Assistenzsystems.....	75
Abbildung 5.2: Graphische Benutzeroberfläche des Assistenzsystems.....	76
Abbildung 5.3: Submodule der SPS-Code-Analyse (Dunkelblau) und der einmaligen Entwicklungsschritte (Hellblau)	80
Abbildung 5.4: Beispiel für die Assoziationsergänzung (hellblau)	81
Abbildung 5.5: UML-Diagramm der Klassen mit Bezug zur Analyse des SPS-Code-Modells .	83
Abbildung 5.6: Visualisierung der Ergebnisse der SPS-Code-Analyse.....	84
Abbildung 5.7: Submodule des EA/RTLS-Analyse-Moduls.....	86
Abbildung 5.8: Wissensgraph für das intelligente Lager in Neo4j Desktop.....	88
Abbildung 5.9: Submodule der Template-Identifikation	90
Abbildung 5.10: Manuell sortierter Extrakt des Wissensgraphen	91
Abbildung 5.11: Export des Wissens als AML-Datei in „pretty print“ Strukturierung.....	92
Abbildung 5.12: Resultierender Digitaler Zwilling im Automation Designer mit manuell ergänztem CAD-Modell	93
Abbildung 6.1: Die drei Aspekte zur Evaluierung der entwickelten Methodik.....	94
Abbildung 6.2: Blick auf das intelligente Lager mit dem teilweise verdeckten mobilen Roboter, der Laserbearbeitungsmaschine und der übergeordneten Steuerung im Hintergrund.....	98
Abbildung 6.3: CAD-Modell der FlexCell mit nachträglicher Beschriftung	99
Abbildung 6.4: Ergebnis der Erhebung für die Untersuchungsdauer des intelligenten Lagers .	103
Abbildung 6.5: Vergleich des Mean (grau) und Median (gelb) im Vergleich zu den Expertenantworten (blau) für die Fragen zwei bis fünf.....	104
Abbildung 6.6: Manuelle Schritte des Ist-Ablaufs, deren Dauer und Herausforderungen	105
Abbildung 6.7: Gegenüberstellung der Dauer des manuellen und des automatisierten Prozesses zur Erstellung der Relationen des Digitalen Zwillings für das intelligente Lager	106

Tabellenverzeichnis

Tabelle 1: Vergleich der bestehenden Ansätze in der Wissenschaft und Technik	32
Tabelle 2: Vergleich von Klassifikationsverfahren für Zeitreihen	58
Tabelle 3: Komponenten des intelligenten Lagers	100
Tabelle 4: Komponenten des Indoor-Lokalisationssystems	101

Abkürzungsverzeichnis

API	A pplication P rogramming I nterface
ABox	A ssertional B ox
ARENA2036	A ctive R esearch E nvironment for the Next Generation of A utomobiles 2036
AML	A utomation M arkup L anguage
CAD	C omputer- A ided D esign
CAEX	C omputer- A ided E ngineering E xchange
CAX	C omputer- A ided x
COLLADA	C ollaborative D esign A ctivity
CPS	C yber- p hysisches S ystem
CPPS	C yber- p hysisches P roduktionssystem
CPU	C entral P rocessing U nit
CSV	C omma-separated V alues
DI	D igital I nterface
DIKW	D ata I nformation K nowledge W isdom
DIN	D eutsches I nstitut für N ormung
DO	D igitale O utput
DTW	D ynamic T ime W arping
DZ	D igitaler Z willing
E/A	E in-/ A usgabe S chnittstelle
ECAD	E lectronic and E lectrical C omputer- A ided D esign
EMF	E clipse M odeling F ramework
FUP	F unktionsplan
GUI	G raphical U ser I nterface
GSpan	g raph-based S ubstructure p attern mining
HMI	H uman- M achine I nterface
IAS	I nstitut für A utomatisierungstechnik und S oftwaresysteme
IEC	I nternational E lectrotechnical C ommission
iLager	i ntelligentes L ager

IPC	I ndustrial P ersonal C omputer
kNN	K -Nearest-Neighbor
LPG	L abeled P roperty G raph
MPS	M odulares P roduktionssystem
ODP	O ntology D esign P attern
OPC UA	O pen C onnectivity U nified A rchitecture
OWL	W eb O ntology L anguage
PLM	P roduct- L ifecycle- M anagement
PROFIBUS	P rocess F ield B us
Profinet	P rocess F ield N etwork
RDF	R esource D escription F ramework
RDFS	R esource D escription F ramework S chema
RFID	R adio-frequency I dentification
RTLS	R eal- T ime L ocating S ystem
SPARQL	S PARQL P rotocol A nd R DF Q uery L anguage
SPS	S peicherprogrammierbare S teuerung
SQL	S tructured Q uery L anguage
STEP	S tandard for the E xchange of P roduct model data
TBox	T erminology B ox
UWB	U ltra-wideband
VDI	V erein D eutscher I ngenieure
VPN	V irtual P rivate N etwork
W3C	W orld W ide W eb C onsortium
WLAN	W ireless L ocal A rea N etwork
XML	E xtensible M arkup L anguage
XSD	X ML S chema D efinition
3D	3 - D imensional
4GD	4 th G eneration D esign

Begriffsverzeichnis

Aktor: Ein Gerät oder Schaltkreis, das ein Event oder Prozess anstößt basierend auf einem Steuersignal. [1]

Architektur: „Kombination von Elementen eines Modells aufbauend auf Prinzipien und Regeln zum Zweck seiner Konstruktion, Weiterentwicklung und Nutzung.“ [2]

Asset: „Entität, die einen wahrgenommenen oder tatsächlichen Wert für eine Organisation hat [...]“ [3] Es kann ein einzelnes Bauteil oder eine ganze Produktionsanlage bezeichnen.

Betriebsdaten: Als Betriebsdaten werden im Betrieb des Produktionssystems erzeugte Aktor- und Sensorwerte bezeichnet.

Brownfield: Der Begriff „Brownfield“ wird für bestehende Produktionssysteme verwendet, die aktiv genutzt werden. Durch den bestehenden Betrieb ergeben sich Einschränkungen für die Untersuchung und Modifikation dieses Systems.

Cyber-physisches System: „System, das reale (physische) Objekte und Prozesse verknüpft mit informationsverarbeitenden (virtuellen) Objekten und Prozessen über offene, teilweise globale und jederzeit miteinander verbundene Informationsnetze.“ [3]

Cyber-physisches Produktionssystem: Produktionssystem aus mehreren CPS.

Digitaler Zwilling: Der Digitale Zwilling ist das virtuelle Abbild eines physischen Assets und der „cyber“-Anteil eines cyber-physischen Systems. Anhand von Modellen und Relationen werden die statischen und dynamischen Eigenschaften des Assets nachgebildet. Darüber hinaus muss eine Datenerfassung, eine Synchronisierung und die Simulierbarkeit gegeben sein. [4]

Mechatronische Komponente: Ein Asset, das mechanische, elektrische und steuerungstechnische Anteile umfasst. [5]

Metamodell: „Das Metamodell beschreibt die Struktur eines Modells und beschreibt so den Modelltyp. Es definiert die verfügbaren Modellelemente und deren mögliche Verknüpfungen. Das Meta-Modell ist vergleichbar mit einem Klassendiagramm, das die Struktur eines zugehörigen Objektdiagramms festlegt.“ [6]

Modell: Ein Modell ist eine „vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen [...], [7]. Dabei muss ein Modell nicht das komplette System abbilden sondern muss lediglich eine „Schlüssige, ausreichend detaillierte Abstraktion von Aspekten in einem Anwendungsbereich“ [3] sein.

Peripherie: Als Peripherie oder auch Komponente des Produktionssystems werden die Aktoren und Sensoren eines Systems zusammengefasst. Sie werden durch die Tags in einer SPS in der Steuersoftware repräsentiert.

Produktionssystem: Als Produktionssystem werden die technischen Anlagen und Maschinen zur Herstellung eines materiellen Produkts verstanden. Das Produktionssystem führt, gesteuert durch das Automatisierungssystem, den technischen Herstellungsprozess aus.

Prozess: Die Durchführung von mindestens einer Aufgabe durch einen Menschen oder eine Maschine unter der Verwendung von Ressourcen wie Energie oder Material mit dem Ziel, eine Eingangsgröße in eine geplante Ausgangsgröße zu überführen [8].

Rekonfiguration: „Die Rekonfiguration umfasst die technische Sicht des Prozesses der Veränderung eines bereits entwickelten und operativ eingesetzten Systems, um es an neue Anforderungen anzupassen, Funktionalität zu erweitern, Fehler zu beseitigen oder die Qualitätseigenschaften zu verbessern.“ [9]

Semantik: „Die Semantik befasst sich mit der Beziehung zwischen Zeichen und ihrer Bedeutung. Zeichen können grafischer Natur (z. B. Verkehrszeichen) oder alphanumerische Einheiten (z. B. Wörter) sein, die Dinge benennen.“ [10]

Sensor: Sensoren konvertieren eine nicht elektrische Eingangsgröße in ein analoges oder digitales elektrisches Signal. [1]

Speicherprogrammierbare Steuerung (SPS): Eine Steuereinheit für industrielle Anwendungen, wie die Steuerung von Maschinen. Für die Erfüllung dieser Aufgabe kommuniziert die SPS mit anderen Geräten, wie zum Beispiel Sensoren und Aktoren. [1] Die SPS liest zyklisch die Sensorwerte ein und bestimmt anhand des SPS-Programms die Ausgangswerte.

Wissensmanagement: Als Wissensmanagement wird die Summe der Aufgaben, Wissen aus verschiedenen Quellen zusammenzuführen, zu speichern, anzureichern und zu ergänzen, bezeichnet.

Zusammenfassung

Die heutigen Produktionssysteme werden zunehmend digital vernetzt, um ihre Flexibilität zu erhöhen und aktuellen Trends, wie zum Beispiel der Produktion kundenindividueller Produkte zu geringen Preisen, nachzukommen. Unterstützt durch Konzepte, wie die modellbasierte Systementwicklung, entstehen cyber-physische Produktionssysteme mit einem Digitalen Zwilling als virtuelle Repräsentation. Digitale Zwillinge eröffnen großes Potential zur Kosteneinsparung indem etwa Vorhersagen getroffen, Optionen untersucht, Optimierungen getestet oder Fehler nachvollzogen werden können, ohne das reale Produktionssystem zu beeinträchtigen. In der Industrie existieren zahlreiche aktiv verwendete Produktionssysteme, sogenannte Brownfield-Produktionssysteme, die über keinen Digitalen Zwilling verfügen, da sie vor längerer Zeit entwickelt und realisiert wurden. In diese Brownfield-Produktionssysteme wurden bereits hohe Investitionen getätigt, weshalb sie aus ökonomischen Gründen erhalten und nachhaltig um einen Digitalen Zwilling ergänzt werden sollten. Aktuell wird der Digitale Zwilling für diese Bestandssysteme überwiegend manuell erstellt. Dies ist zeitaufwendig, erfordert kostspielige Experten aus den Fachbereichen der Mechanik, Elektrik und Software und ist anfällig für Flüchtigkeitsfehler. Die Industrie verfügt bisher über keine Methodik zur automatisierten Erstellung eines Digitalen Zwillings für Bestandssysteme, um die benötigte Zeit, die Kosten und resultierende Fehler des manuellen Prozesses zu reduzieren. In der Wissenschaft gibt es bereits Ansätze, die einzelne Modelle erstellen. Für Digitale Zwillinge fehlen diesen Ansätzen die Berücksichtigung der Relationen zwischen den einzelnen Modellen eines Digitalen Zwillings.

In der vorliegenden Arbeit wird daher eine Methodik entwickelt, die eine automatisierte Erstellung dieser Relationen für ein bestehendes Produktionssystem als Basis für einen Digitalen Zwilling ermöglicht. Dabei werden Relationen zwischen Modellen verschiedener Fachbereiche, zwischen Vorlagentemplates und deren Instanzen, sowie Eltern-Kind-Relationen zur Erstellung der klassischen CAEX-Baumstruktur berücksichtigt. Dazu werden der SPS-Code, Betriebsdaten und Positionsdaten als Quellen herangezogen, die ohne eine Einschränkung des produktiven Betriebs analysiert werden können. Die gewonnenen Informationen aus den einzelnen Analysen werden anschließend verknüpft und in einem Wissensgraphen zusammengeführt. Durch eine Analyse des Graphen nach häufigen Teilgraphen werden sich wiederholende Strukturen identifiziert und damit das Wissen angereichert. Dieser computergestützte Schritt reduziert die Anzahl der Fehler im Vergleich zur manuellen Strukturierung des Wissens und erhöht damit die Wiederverwendbarkeit auch für weitere Projekte. Die semantische Anreicherung des Wissens mit Ontologien ermöglicht die automatisierte Integration weiterer Informationen, die maschinelle Verarbeitung und erhöht die Eindeutigkeit des Wissenskontextes. Die Evaluierung erfolgt anhand des Vergleichs des automatisierten Vorgehens der entwickelten Methodik am intelligenten Lager in der ARENA2036 und dem manuellen Vorgehen von Fachexperten. Zusätzlich zu dieser Evaluierung der Mehrwerte wird die Übertragbarkeit anhand eines beispielhaften Übertrags sowie einer Diskussion der Randbedingungen der Methodik gezeigt.

Abstract

Current production systems are increasingly digitally connected to improve their flexibility and meet current trends, such as the production of customized products at low prices. Supported by concepts such as model-based system engineering, cyber-physical production systems are emerging with a Digital Twin as a virtual representation. Digital Twins unlock great potential for cost savings by making predictions, investigating options, testing optimizations, or reproducing errors without affecting the real production systems. In industry, there are many actively used production systems, so-called brownfield systems, which do not have a Digital Twin because they were developed and implemented a long time ago. High investments have already been made in these brownfield systems, which is why they should be maintained and sustainably enhanced with a Digital Twin for economic reasons. Currently, Digital Twins for existing systems are mainly created manually. This manual process is time-consuming, requires expensive experts from the mechanical, electrical, and software domains and is prone to oversights. The industry currently has no methodology for the automated creation of a Digital Twin for existing systems to reduce the time, costs, and resulting errors caused by the manual process. In research, there are already approaches that create individual models. However, these approaches do not consider the relations between the individual models of a Digital Twin.

This work therefore develops a methodology that enables the automated creation of these relations for an existing production system as the basis for a Digital Twin. Relationships between models from different domains, between templates and their instances, as well as parent-child relationships for creating the classic CAEX tree structure are considered. For this purpose, the PLC code, operational data, and position data are used as sources that can be analyzed without restricting active production. The information obtained from the individual analyses is then linked and merged into a knowledge graph. By analyzing the graph for frequent subgraphs, repetitive structures are identified, and the knowledge is subsequently enriched. This computer-aided step reduces the number of errors compared to manual structuring of the knowledge and thus increases reusability for further projects. Semantic enrichment of the knowledge with ontologies enables the automated integration of further information, machine processing, and increases the clarity of the knowledge context. The evaluation is based on a comparison of the automated methodology developed for the intelligent warehouse in ARENA2036 and the manual procedure used by domain experts. In addition to this evaluation of the added value, the applicability is shown through an exemplary transfer and a discussion of the constraints of the methodology.

1 Einleitung

Die sich wandelnde Erwartungshaltung an die produzierende Industrie sowie der technologische Fortschritt führen dazu, dass bestehende Produktionssysteme weiterentwickelt werden. Dabei ist die steigende Digitalisierung und Vernetzung von (Teil-)Systemen der Produktionskette sowie die Erweiterung von Bestandsystemen um einen Digitalen Zwillingen zu Cyber-physischen Produktionssystemen ein zentrales Konzept. In diesem Kapitel wird der Frage nachgegangen, warum für Produktionssysteme diskreter Produkte ein Digitaler Zwilling relevant ist und welche Vorteile sich daraus ergeben. Darauf aufbauend wird der gegenwärtig manuelle Erstellungsprozess von Digitalen Zwillingen bestehender Produktionssysteme und die damit verbundenen Schwierigkeiten der Industrie aufgezeigt. Anschließend werden die Herausforderungen formuliert, mit der ein Konzept zur nachträglichen Erstellung eines Digitalen Zwillings für bestehende Produktionssysteme konfrontiert ist. Daraus wird das Forschungsziel herausgearbeitet und die Anforderungen an die hiesige Forschung abgeleitet. Der letzte Teil dieses Kapitels skizziert die Struktur der Vorgehensweise dieser Arbeit.

1.1 Ausgangssituation in der diskreten Produktion

Die produzierende Industrie befindet sich in einem Umbruch, der zu einer kontinuierlichen Anpassung und Erweiterung der bestehenden Produktionssysteme führt. Dem Konzept des Digitalen Zwillings wird dabei ein großes Potential zugesprochen, wie etwa die Flexibilität und Fähigkeiten eines Produktionssystems zu erweitern sowie den Entwurf neuer Produkte und deren Herstellungsprozess entlang der gesamten Produktionskette unter diesen neuen Bedingungen zu unterstützen [11].

Der Wandel in der Industrie wird durch den technologischen Fortschritt (Technologiedruck) und die sich ändernden Rahmenbedingungen an die Produktion (Bedarfssog) eingeleitet [12, 13]. Der Änderungs- und Erweiterungsbedarf von bestehenden Produktionssystemen entsteht dabei unter anderem durch neue gesetzliche umweltbezogene Anforderungen und gesellschaftliche Erwartungen (geringerer Verbrauch, niedrigere Emissionen, Schutz der Menschen, etc.). Auch Unternehmen selbst stecken sich zunehmend neue Umweltziele [14]. Ein weiterer Sogfaktor, der beispielsweise durch die Coronapandemie verstärkt aufgetreten ist, sind fehlende Ressourcen durch Lieferverzögerungen und unterbrochene Lieferketten. Die fehlenden Zulieferteile oder Produktionsressourcen müssen zum Beispiel durch eine Anpassung des Produktes oder der Produktion kompensiert werden. Ebenso sind die Nachfrage nach individuelleren Produkten durch die Kunden [15] sowie die Entwicklung neuer Produkte durch die Firmen in stetig kürzeren Produktzyklen zur Steigerung der Verkaufszahlen [13, 16] Auslöser, die eine häufige und zügige Anpassung der Produktionssysteme erforderlich machen. Diese Aspekte sind äußere Einflüsse,

welche die Häufigkeit der Anforderungsänderungen an bestehende Produktionssysteme erhöhen oder zur Anpassung von Produkten selbst und infolgedessen zur Rekonfiguration der Produktionssysteme führen [17]. Beispielsweise lässt sich in der Automobilindustrie bereits erkennen, dass der Wettbewerb und die Anforderungsänderungen zu einer höheren Produktvielfalt innerhalb kürzerer Zeit geführt haben [13].

Neben den äußeren Einflüssen sind die inneren Einflussfaktoren ein Treiber zur Weiterentwicklung von Produktionssystemen [17, 18]. Zu diesen Push-Faktoren gehört der aktuelle technologische Fortschritt hinsichtlich der verstärkten Vernetzung, Digitalisierung und Kommunikation zwischen Produktionssystemen gegenüber bisherigen Systemen [19]. Diese Technologien sind grundlegend für Konzepte wie der Digitale Zwilling und Cyber-physische-Systeme, die unter dem Begriff Industrie 4.0 zusammengefasst werden können [20], und erleichtern deren Verbreitung. Die neuen Konzepte ermöglichen nicht zuletzt eine höhere Flexibilität beziehungsweise eine leichtere Rekonfiguration¹ von Produktionssystemen. Das wiederum befähigt Produktionssysteme den äußeren Einflüssen gerecht zu werden. Daher ist die Modernisierung und Erweiterung von bestehenden Produktionssystemen um einen Digitalen Zwilling relevant, um auf die variableren Anforderungen reagieren zu können.

Der Digitale Zwilling bildet mithilfe verschiedener Modelle und deren Beziehungen zueinander das Produktionssystem virtuell ab. Die digitale Abbildung von Produktionssystemen ermöglicht beispielsweise die Planung, eine simulative Generierung oder virtuelle Tests von Modifikationen, ohne das reale Produktionssystem zu verwenden. Dadurch können Rekonfigurationen eines bestehenden Produktionssystems parallel zum aktiven Produktionsbetrieb ermittelt und Modifikationen schneller durchgeführt werden. Hieraus resultierende konkrete, kostensparende Anwendungsfälle sind eine virtuelle Inbetriebnahme [21], Wartungsvorhersagen oder eine virtuelle Produktionsplanung [22]. Darüber hinaus sind zahlreiche weitere Anwendungen entlang des kompletten Anlagenlebenszyklus möglich [23, 24]. Der Vorteil der Nutzung des Digitalen Zwillings besteht im Wesentlichen darin, dass der Produktionsbetrieb nicht eingeschränkt werden muss und verschiedene Optionen virtuell erprobt werden können. Dadurch lassen sich der Planungsaufwand reduzieren, die Stillstandzeit für den Umbau verkürzen oder Ausfallzeiten durch Umbaufehler vermeiden. In der Realität kann so ein effektiver Eingriff zur Rekonfiguration des bestehenden Produktionssystems erfolgen. Der Digitale Zwilling, dessen Relationen das Ziel dieser Arbeit sind, wird in Kapitel 2.2 definiert und ausführlich betrachtet. Der Digitale Zwilling ermöglicht es also, auf Änderungen durch die inneren und äußeren Einflüsse bei gleichbleibendem Preis und Qualität mit existierenden Produktionssystemen zu reagieren [25].

¹ „Eine Rekonfiguration stellt die technische Sicht des Prozesses der Veränderung eines bereits entwickelten und operativ eingesetzten Systems dar, um es an neue Anforderungen anzupassen, Funktionalität zu erweitern, Fehler zu beseitigen oder die Qualitätseigenschaften zu verbessern“ [9]

1.2 Industrielle Problemstellung

Heutzutage werden neue Produktionssysteme mittels modellbasierter Entwicklungsmethoden entworfen und geplant. Dieser Ansatz führt vor allem in den frühen Entwicklungsphasen zu einer höheren Reife der Systementwicklung und insgesamt zu einer höheren Qualität als mit dem dokumentenbasierten Entwicklungsansatz [26]. Durch die modellbasierte Entwicklung können fachbereichs- und teilsystemübergreifende Abhängigkeiten früher berücksichtigt und Modellierungsfehler entdeckt werden. Dies ergibt sich durch die leichtere Verständlichkeit der visuellen Modelle für Menschen als dokumentenbasierte, textuelle Entwürfe und die Parallelisierung der Arbeiten aus den involvierten Fachbereichen [26]. Die entstehenden Modelle können für einen Digitalen Zwilling verwendet werden, die dann noch in eine entsprechende Gesamtarchitektur eingebracht werden müssen. Die manuelle Erstellung eines Digitalen Zwillings kann dabei bereits durch Software, wie beispielsweise durch den Siemens Automation Designer, unterstützt werden. Templates von Anlagenteilen als Bibliothekselemente assistieren bei der Erstellung und beim Skalieren der Modellierung für große Anlagen mit wiederholenden Anordnungen beziehungsweise für mehrere Systeme derselben Art. Die modellbasierte Entwicklungsmethode wird jedoch in der Regel zum Entwurf neuer Produktionssysteme eingesetzt.

Anders ist es bei bereits fertig entwickelten und aktiv genutzten Produktionssystemen, den sogenannte Brownfield-Produktionssystemen². Diese zeichnen sich dadurch aus, dass sie zu einem früheren Zeitpunkt entwickelt, gebaut und in Betrieb genommen wurden und sich seit einiger Zeit aktiv im Einsatz befinden. Meist sind Brownfield-Produktionssysteme in der Zwischenzeit bereits ein- oder mehrfach modifiziert worden, wovon oft keine oder nur eine unvollständige Dokumentation zur Verfügung steht. Für diese Brownfield-Produktionssysteme existieren aufgrund ehemals noch nicht vorhandener technischer Möglichkeiten auch keine digitalen Modelle oder gar ein Digitaler Zwilling [32]. Daher können diese Systeme bei der Anpassung an neue innere und äußere Einflussfaktoren durch derartige Konzepte nicht unterstützt werden. Diese Ausgangssituation wird in Kapitel 2.1 im Detail betrachtet.

Aktuell erfolgt die Erstellung eines Digitalen Zwillings von Brownfield-Produktionssystemen deshalb in einem manuellen, zeitaufwendigen und fehleranfälligen Prozess. Die nachträgliche Erstellung eines Digitalen Zwillings für ein Brownfield-Produktionssystem kann nicht effektiv durch den modellbasierten Entwicklungsprozess unterstützt werden, da im Vergleich zum Entwurf neuer Produktionssysteme keine Freiheitsgrade bestehen und das bestehende Produktionssystem exakt nachmodelliert werden muss. Außerdem sind in existierenden

² Der Begriff „Brownfield“ entstammt dem Baugewerbe und bezeichnet bereits bebauten Land, das neu verwendet werden soll und durch die vorherige Nutzung kontaminiert sein kann [27]. Im vorliegenden Kontext wird der Begriff „Brownfield“ für bestehende Produktionssysteme verwendet, die aktiv genutzt werden [28]. Durch das bestehende System ergeben sich Einschränkungen vergleichbar zum Baugewerbe [29, 30]. Der Vorgang der Anpassung des Brownfield (oder auch Brownfield-Produktionssystem) wird Brownfield-Entwicklung(-prozess) genannt [31].

Softwaretools zur Modellierung von Produktionssystemen noch keine Templates für die individuellen Brownfield-Produktionssysteme verfügbar.

Aus der manuellen Erstellung des Digitalen Zwillings ergeben sich negative Auswirkungen hinsichtlich der benötigten Zeit, den Kosten, der erzielten Qualität und der Funktionalität des Digitalen Zwillings. Die exakte Modellierung des Ist-Zustands des Produktionssystems erfordert umfangreiche händische Untersuchungen und Recherchen. Die komplexe Struktur der Produktionssysteme und die teils verdeckten Abhängigkeiten machen den Prozess fehleranfällig und begünstigen Flüchtigkeitsfehler, wodurch die Qualität der Modelle leidet. Da die benötigten Modelle aus verschiedenen Fachbereichen stammen, werden dementsprechend verschiedene Experten benötigt, woraus hohe Personalkosten entstehen. Eine Beschränkung auf Modelle einzelner Fachbereiche oder Anlagenteile sowie ein Verzicht auf die Modellierung von Relationen zwischen Modellen reduziert hingegen den Funktionsumfang des resultierenden Digitalen Zwillings.

Um die Vorteile eines Digitalen Zwillings nachhaltig nutzen zu können, muss der Erstellungsprozess des Digitalen Zwillings von Brownfield-Produktionssystemen systematisiert und (teil-) automatisiert werden. Aus der Betrachtung der bestehenden Ansätze in der Wissenschaft und Technik geht hervor, dass bereits Ansätze existieren, die sich auf die Erstellung alleinstehender Modelle eines Fachbereichs konzentrieren. Dabei werden die Relationen zwischen verschiedenen Modellen nicht oder nur selten berücksichtigt. Diese sind jedoch ein Kernbestandteil des Digitalen Zwillings und notwendig, um dessen vielfältige Anwendungsmöglichkeiten und Mehrwerte für Brownfield-Produktionssysteme nutzen zu können.

Wie sich zeigt, besteht in der Industrie eine Wissenslücke über den effektiven Prozess zur Erstellung des Digitalen Zwillings eines Brownfield-Produktionssystems mit der Berücksichtigung der Relationen zwischen den Modellen. Vor dem Hintergrund, das Defizit der Industrie zu adressieren, stellt sich die folgende Forschungsfrage:

Wie kann der aktuell manuelle, zeitaufwendige und fehleranfällige Erstellungsprozess Digitaler Zwillinge für Brownfield-Produktionssysteme verbessert und verkürzt sowie Relationen zwischen Modellen berücksichtigt werden, um von den Mehrwerten eines Digitalen Zwillings zu profitieren?

1.3 Herausforderungen bei der Erstellung eines Digitalen Zwillings für Brownfield-Produktionssysteme

Der nachträgliche Erstellungsprozess eines Digitalen Zwillings für Brownfield-Produktionssysteme läuft ähnlich zum Engineering eines realen, automatisierten Produktionssystems [33] ab und erfordert gleichermaßen fachbereichsübergreifende Expertise. Es

müssen Informationen aus den Fachgebieten Mechanik, Elektrik und Software einbezogen werden, um alle Facetten des Brownfield-Produktionssystems abbilden zu können. Dabei kann der Modellierungsvorgang wie im klassischen Entwicklungsprozess überwiegend sequentiell ablaufen [33, 34] oder, unterstützt durch Softwaretools und einem vereinheitlichten Datenaustausch, bis zu einem gewissen Grad parallelisiert werden [35, 36]. Der Unterschied zum Engineering neuer Systeme liegt darin, dass die Modellierung bei bestehenden Systemen nicht mehr primär an kundenspezifischen Systemanforderungen ausgerichtet wird, sondern das Brownfield-Produktionssystem widerspiegeln muss. Dazu ist die bestehende Anlage genaustens zu analysieren, die Funktionalitäten einzelner Komponenten sind zu identifizieren und nachzuvollziehen. Dazu müssen die folgenden vier Herausforderungen überwunden werden.

(H1) Das erforderliche hohe Maß an Expertenwissen, Anlagenverständnis und Zeit behindert die Erstellung heterogener Modelle und deren Relationen zueinander

Brownfield-Produktionssysteme werden unabhängig von dem gewählten Ansatz durch mehrere Fachexperten aus den verschiedenen Fachbereichen Mechanik, Elektrik und Software entworfen und in einem zeitintensiven Vorgang manuell modelliert und zusammengeführt [26, 37]. Die Qualität der erstellten Modelle und fachbereichsübergreifenden Modellrelationen sind aufgrund verschiedener Modellierungsoptionen von der Expertise, zeitlichen Verfügbarkeit und der Tagesform der Fachexperten abhängig. Sie sind daher nicht objektiv. Experten mit entsprechender Erfahrung sind außerdem nur begrenzt verfügbar und erzeugen hohe Personalkosten. Zusammengefasst entsteht die Herausforderung aus der fehlenden Verfügbarkeit von Expertenwissen, der erforderliche, hohe Zeitaufwand für die manuelle Modellierung des Systems und den resultierenden Personalkosten.

(H2) Fachbereichsübergreifende und fachbereichsinterne Relationen sind für Menschen schwer erfassbar

Produktionssysteme werden aufgrund ihrer Größe und Komplexität durch verschiedene Modelle abgebildet, die mittels Relationen in Beziehung gesetzt werden. Die Untersuchung einer Anlage durch Experten ist schon aufgrund der Größe eine schwierige und fehleranfällige Aufgabe [38]. Abgesehen davon müssen die fachbereichsübergreifenden sowie fachbereichsinternen Relationen zwischen Anlagenteilen und dadurch entstehende Abhängigkeiten entdeckt werden. Einige dieser Abhängigkeiten sind nicht visuell erkennbar, sondern verdeckte oder immaterielle Abhängigkeiten [39, 40], wie zum Beispiel die funktionale Abhängigkeit eines Förderbandes vom zugehörigen Motorstarter. Dies erfordert ein profundes Wissen über das bestehende Produktionssystem und die eingesetzten Komponenten. Die Schwierigkeit, alle Beziehungen und Elemente aufgrund deren Sichtbarkeit sowie der Größe und Verschachtelung eines Produktionssystems sicher zu erkennen und zu modellieren, stellen eine Herausforderung dar und führen zu einer Vernachlässigung der Beziehungen in der Modellierung bestehender Produktionssysteme.

(H3) Das Wissen über Brownfield-Produktionssysteme ist durch die fehlende Dokumentation und Modellierung deutlich eingeschränkt

Brownfield-Produktionssysteme, deren Inbetriebnahme schon eine gewisse Zeit zurückliegt, sind meistens nicht oder nur unvollständig dokumentiert [32], sei es aufgrund der unvollständigen Dokumentation von früheren Änderungen während der bisherigen Laufzeit der Anlage oder nicht mehr vorhandenen, ursprünglichen Aufzeichnungen [41]. Der Zeitdruck durch die Einnahmeausfälle infolge von Stillständen während der Durchführung von Rekonfigurationsmaßnahmen an einem Brownfield-Produktionssystem führt unter anderem zu weniger gut beziehungsweise nicht dokumentierten Änderungen [38, 42]. Das dabei verlorene Wissen fehlt im Nachhinein, liegt in nicht formalisierter Form vor und muss unter erheblichen Kosten erarbeitet werden. Abgesehen davon liegen Brownfield-Produktionssysteme meist hinter dem aktuellen technischen Stand zurück und besitzen keinen aktuellen Digitalen Zwilling oder keine digitale Modellierung [43]. Daher muss eine große Nacharbeit geleistet werden, um die Modelle und deren Beziehungen für einen Digitalen Zwilling nachträglich zu erstellen, da direkt am Brownfield-Produktionssystem gearbeitet werden muss.

(H4) Der Produktivbetrieb eines Brownfield-Produktionssystems schränkt die möglichen Untersuchungen am System stark ein

Im Unterschied zur modellbasierten Entwicklung wird bei der nachträglichen Erstellung digitaler Modelle und deren Relationen von einem Brownfield-Produktionssystem nicht nach (Kunden-) Anforderungen entwickelt, sondern das bestehende Produktionssystem detailliert nachmodelliert. Dies erfordert eine genaue Untersuchung der Anlage. In manchen Fällen muss dazu die Produktion eingestellt werden, damit ein Mensch an das Produktionssystem herantreten und die erforderlichen Teile untersuchen kann oder entsprechende Tests durchgeführt werden können, ohne dass die Safety-Einrichtungen auslösen. Diese Abschaltung eines Produktionssystems erzeugt Einnahmeausfälle [44] und hohe Kosten, die mit höchster Priorität vermieden werden sollen [32]. Es stellt eine Herausforderung dar, alle benötigten Untersuchungen am Produktionssystem durchzuführen, um dieses nachbilden zu können, ohne den Betrieb zu unterbrechen und hohe Kosten zu verursachen.

1.4 Zielsetzung dieser Forschung

Zur Lösung des vorgestellten industriellen Problems und den einhergehenden Herausforderungen formuliert diese Arbeit folgendes Ziel:

Die Entwicklung einer Methodik zur Erstellung der Relationen eines Digitalen Zwillings für Brownfield-Produktionssysteme für die diskrete Fertigung

Dazu soll durch diese Arbeit der überwiegend manuelle, zeit- und kostenaufwendige sowie fehleranfällige Prozess der Erstellung der Relationen eines Digitalen Zwilling für bestehende Produktionssysteme diskreter Produkte, sogenannte Brownfield-Produktionssysteme, unterstützt und (teil-) automatisiert werden. Die hierfür entwickelte Methodik soll damit einen Beitrag zur Digitalisierung von existierenden Produktionssystemen und deren Weiterentwicklung durch einen Digitalen Zwilling zu cyber-physischen Produktionssystemen leisten.

Die entwickelte Methodik kombiniert dazu verschiedene Methoden, die zur Erkennung und Extraktion von Relationen in der Anlage sowie dessen Zusammenführung und Anreicherung eingesetzt werden. Die knappen Ressourcen, wie entsprechende Experten, können dann auf die verbleibenden Aufgaben verteilt und für wertschöpfende Tätigkeiten eingesetzt werden. Im Detail lassen sich damit die folgenden zwei Teilziele formulieren: Als erstes Teilziel sollen Ingenieure bei der Untersuchung und digitalen Abbildung von bestehenden Produktionssystemen unterstützt werden. Dazu muss die Methodik Schritte zur automatisierten Identifikation von funktionalen und räumlichen Relationen beinhalten. Dadurch können die benötigte Zeit, die durch manuelle Tätigkeiten entstehende Kosten der Ingenieure und die menschlichen Fehler reduziert werden. Folglich lässt sich so die gewonnene Arbeitszeit für die kreativen Aufgaben der Weiterentwicklung und Rekonfiguration nutzen. Dabei ist es wichtig, dass der Vorgang ohne größere Einschränkungen des Produktionsbetriebes durchgeführt werden kann und die Methodik somit für Brownfield-Produktionssysteme einsetzbar ist. Als zweites Teilziel ist eine strukturierte Bereitstellung des benötigten Wissens zur Bildung der Relationen des Digitalen Zwilling notwendig. So kann das Wissen bedarfsgerecht mit den jeweiligen existierenden Informationen zum bestehenden Produktionssystem kombiniert und in das Zielsystem als Digitaler Zwilling eingebracht werden können.

1.5 Anforderungen an die Konzeption

Bei der Erreichung des im vorherigen Kapitel formulierten Ziels dieser Arbeit sind zur Bewältigung der vorhergehenden Herausforderungen passende, überwiegend funktionale Anforderungen zu identifizieren. Dazu werden aus den dargelegten Herausforderungen zur Erstellung der Relationen zwischen den Modellen des Digitalen Zwilling eines Brownfield-Produktionssystems resultierende Anforderungen abgeleitet. Die in dieser Arbeit entwickelte Methodik muss diese Anforderungen erfüllen, um die Herausforderungen zu adressieren und die zugrundeliegende Problemstellung zu lösen. Sie dienen der späteren Evaluierung der Methodik.

(A1) Automatisierte Erstellung zur Reduzierung von Fehlern und der benötigten Zeit

Die erste Anforderung an die Konzeption ist die Automatisierung des Erstellungsprozesses des Digitalen Zwilling, sodass der Vorgang beschleunigt sowie die Abhängigkeit von individuellem Wissen und die Anzahl von Flüchtigkeitsfehlern reduziert wird. Diese Anforderung adressiert die erste Herausforderung (H1). Durch einen automatisierten Ablauf wird die Abhängigkeit von

Experten verringert, woraus mehrere Vorteile resultieren. Zum einen wird für die Erstellung eines Digitalen Zwillings weniger Wissen vorausgesetzt und dadurch die Kosten für Experten reduziert. Ebenso kann dadurch potenziell früher mit der Erstellung begonnen werden, da nicht auf die Verfügbarkeit aller Experten gewartet werden muss. Die Erstellung des Digitalen Zwillings selbst wird durch die automatische Generierung zeitlich verkürzt und die Dauer vereinheitlicht, da sie nicht mehr von individuellen Erfahrungen und Kenntnissen abhängt. Abschließend werden durch eine automatisierte Durchführung menschliche Flüchtigkeitsfehler aufgrund von Zeitdruck, einer hohen Anzahl zu berücksichtigender Einflussfaktoren und limitierten Kenntnissen verhindert.

(A2) Erstellung der Relationen automatisierter, mechatronischer Systeme

Die zweite Anforderung an die Konzeption ist es, die Relationen zwischen den Modellen eines Digitalen Zwillings bei der automatisierten Erstellung zu berücksichtigen. Dazu müssen über die einzelnen Fachbereiche Informationen gesammelt, daraus Wissen über das Brownfield-Produktionssystem gewonnen, zusammengebracht und angereichert werden. Der Fokus dieser Anforderung liegt dabei auf den Relationen zwischen den Modellen verschiedener Fachbereiche (Inter-Domänenrelationen) und innerhalb eines Fachbereichs (Intra-Domänenrelationen). Die Modelle selbst sind nicht Kern dieser Anforderung. Hinsichtlich der Relationen werden Eltern-Kind-Relationen benötigt, um die typische Baumstruktur einer Stückliste (BOM) des Produktionssystems abzubilden, Klassen-Instanz-Relationen zur Abbildung mehrfach instanzierter Anordnungen als Templates, und Instanz-Instanz-Relationen zur Verknüpfung von Elementen unterschiedlicher Fachbereiche (z.B. ein Sensor über ein Signal mit einem Inputmodul der SPS). Die Berücksichtigung dieser Relationen ist bei der Erstellung des Digitalen Zwillings ein relevanter Bestandteil, um die Herausforderung zwei (H2) bewältigen zu können.

(A3) Erstellung der Relationen des Digitalen Zwillings ohne (semi-)formales Expertenwissen

Die Erstellung eines Digitalen Zwillings für ein Brownfield-Produktionssystem kann in der Regel nicht auf eine vollständige Wissensbasis aufbauen (H3) und muss daher ohne (semi-)formal dokumentiertes Wissen über das Produktionssystem durchgeführt werden. Entsprechende Quellen wie papierbasierte oder digitale Dokumentationen sowie digitale Modelle sind nicht standardmäßig als gegeben zu betrachten. Die Methodik muss deshalb Wissen aus anderen Quellen extrahieren, die dieses implizit enthalten. Es gilt zusammenfassend dieses Wissen zu extrahieren, zu formalisieren und anzureichern, sodass es als Grundlage für den Digitalen Zwilling zur Verfügung steht.

(A4) Erstellung des Digitalen Zwillings ohne Beeinträchtigung des produktiven Betriebs

Um Kosten infolge eines Anlagenstillstands zu vermeiden (H4), muss die Methodik auf Quellen aufsetzen, deren Verwendung keine Einschränkung des produktiven Betriebs hervorrufen. Dabei sind sowohl Abschaltungen aus Gründen der Sicherheit eines untersuchenden Ingenieurs als auch

anlässlich von Testläufen im Produktionssystem oder für die Einbringung von Fremdsystemen zur Untersuchung zu vermeiden. Die vierte Anforderung ist daher, dass mögliche Quellen entweder bereits vorhanden oder parallel zum eigentlichen Betrieb nutzbar sein müssen.

1.6 Aufbau der Arbeit

Diese Arbeit gliedert sich in sieben Hauptkapitel, deren Struktur in Abbildung 1.1 dargestellt ist. Zur Erreichung des Ziels dieser Arbeit wird nach der wissenschaftlichen Entwurfsforschung³ (DSR) von Hevner et al. in [45] vorgegangen. Die Definition des Problems, dessen Umgebung und die formulierten Herausforderungen sowie Anforderungen in diesem Kapitel sind Bestandteil der DSR-Relevance-Phase.

Nach dieser Einführung in die Ziele dieser Forschung werden als Grundlagen der Arbeit in Kapitel 2 automatisierte Brownfield-Produktionssysteme als Ausgangssituation und der Digitale Zwilling als Zielvorstellung behandelt. Anschließend wird der Stand der Wissenschaft und Technik hinsichtlich existierender Methoden zur Erstellung Digitaler Zwillinge beleuchtet und die wissenschaftliche Forschungslücke aufgezeigt. Diese beiden Kapitel bilden die Wissensgrundlage und den Ausgangspunkt für den Rigor-Cycle der DSR.

Kapitel 4 beschreibt die entwickelte Methodik zur Erstellung der Relationen zwischen den Modellen des Digitalen Zwilling bestehender Produktionssysteme zur Schließung der ermittelten Forschungslücke. Dabei werden die Bausteine der Methodik individuell betrachtet, beginnend bei den drei involvierten Informationsquellen, über die Wissensextraktion bis zur anschließenden Zusammenführung des generierten Wissens und dessen Anreicherung. Jeder dieser Bausteine stellt ein Artefakt nach der Definition des DSR dar. Es wird jeweils schrittweise ein Artefakt zur Analyse des SPS-Codes, der Betriebs- und Positionsdaten (Kapitel 4.3 - 4.5) entwickelt. Außerdem wird je ein Artefakt zur Zusammenführung und Speicherung des Wissens in einer Graphdatenbank (Kapitel 4.6.1 & 4.6.2), zur semantischen Anreicherung (Kapitel 4.6.3) sowie zur Analyse repetitiver Strukturen (Kapitel 4.6.4) herausgearbeitet.

Die Implementierung der Methodik als Software-Assistenzsystem wird in Kapitel 5 vorgestellt. Es wird zu Beginn ein Überblick über die Softwarearchitektur gegeben und anschließend die Module des Assistenzsystems einzeln im Detail betrachtet. Dabei werden die Artefakte hinsichtlich ihrer Umsetzung verfeinert, sodass sie gegen die Anforderungen aus dem Relevance Cycle evaluiert werden können. In dieser Arbeit wird auf die vollständige Abbildung der Iterationszyklen zugunsten einer stringenten Darstellung verzichtet und lediglich die erfolgreichen Artefakte erläutert.

Die Methodik und ihre Realisierung werden in Kapitel 6 nach drei Bewertungskriterien evaluiert. Zum einen wird die Erfüllung der Anforderungen und Erreichung der Zielsetzung nachgewiesen,

³ Besser bekannt unter dem englischen Begriff „Design Science Research“ (DSR)

zum anderen werden die Mehrwerte der Methodik evaluiert und darüber hinaus wird die Übertragbarkeit der Methodik diskutiert. Nach diesem Kapitel ist der Durchlauf des Relevance Cycle abgeschlossen.

Kapitel 7 fasst die wesentlichen Ergebnisse dieser Arbeit abschließend zusammen, bewertet diese und gibt einen Ausblick auf mögliche zukünftige Einsatz- sowie Erweiterungsmöglichkeiten.

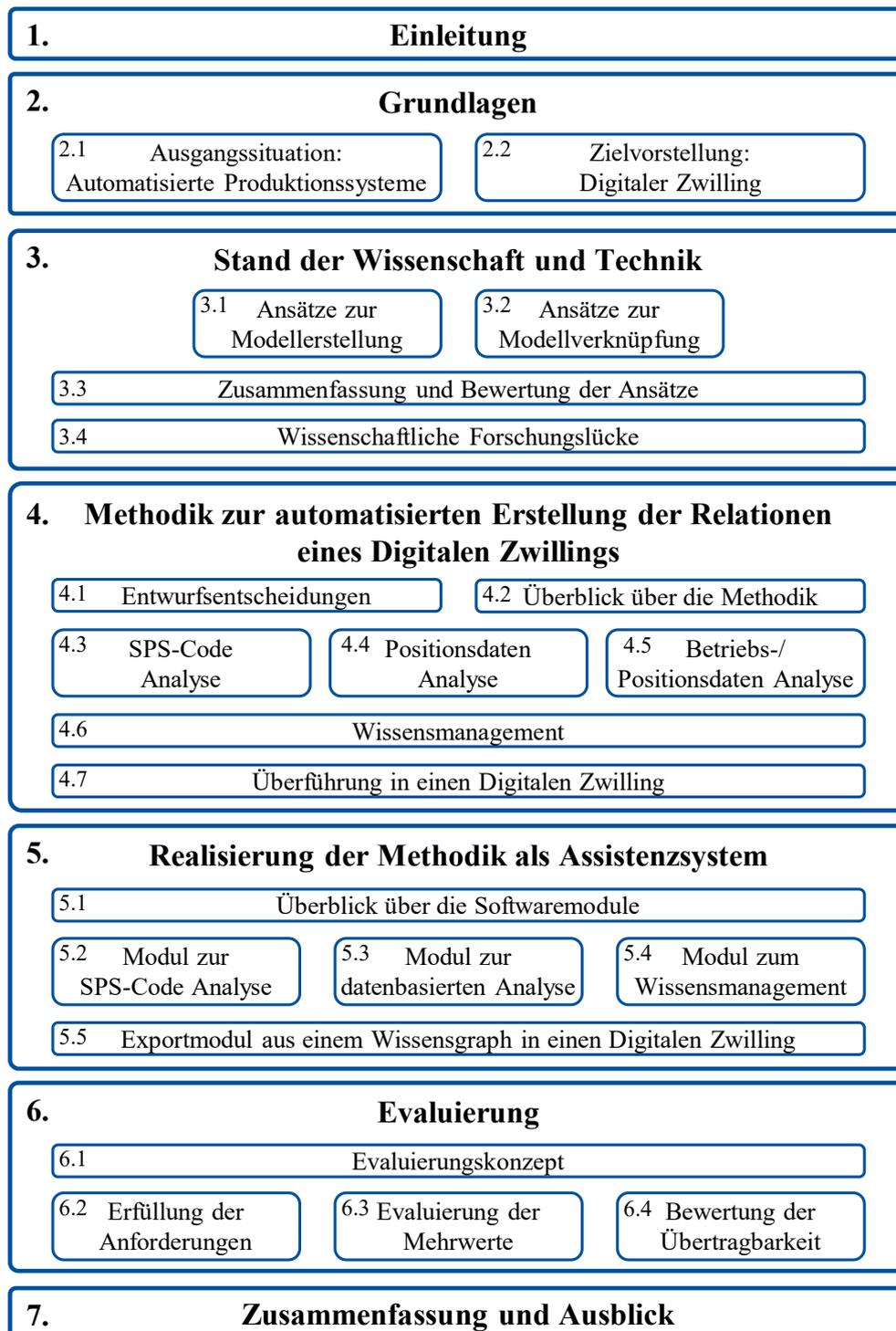


Abbildung 1.1: Aufbau der Arbeit

2 Grundlagen

Dieses Kapitel umfasst die relevanten, theoretischen Grundlagen dieser Arbeit und bildet die Basis für die weitere Untersuchung des aktuellen Standes der Wissenschaft und Technik und der Entwicklung der Methodik. Motiviert durch die ersten Kapitel zur Ausgangssituation, der industriellen Problemstellung und den beschriebenen Herausforderungen werden automatisierte Produktionssysteme zur Herstellung diskreter Produkte, deren grundsätzlicher Aufbau, die Besonderheiten und Herausforderungen betrachtet. Die Automatisierung dieser Systeme wird diskutiert und die existierenden Standards, Architekturen und Kommunikationsprotokolle eingeordnet. Dabei wird der Einfluss der Neuerungen im Zuge von Industrie 4.0 sowie deren Vorteile und Herausforderungen aufgezeigt. Darauf aufbauend wird der Entwicklungsprozess dieser komplexen, automatisierten Produktionssystemen beschrieben und die modellbasierte Systementwicklung als erfolgsversprechende Möglichkeit vorgestellt, die auch im späteren Bezug zum Digitalen Zwilling von diesen Produktionssystemen einen Vorteil eröffnet. Die Grenze dieses Entwicklungsansatzes für die Weiterentwicklung von bestehenden Systemen wird im Anschluss thematisiert und die Besonderheiten, die diese Systeme an die Erstellung eines Digitalen Zwillings stellen. In diesem Zuge werden auch die Mehrwerte eines Digitalen Zwillings für Bestandssysteme besprochen und welche weiteren Trends einen Einfluss auf die Weiterentwicklung von diesen Systemen haben. Der Digitale Zwilling als ein Fokus dieser Arbeit wird im Nachgang genauer betrachtet. Dabei werden dessen Entstehung und Ausprägungen betrachtet, eine Definition für diese Arbeit aus verschiedenen Veröffentlichungen abgeleitet, die vielfältigen Einsatzmöglichkeiten sowie daraus resultierende Mehrwerte zusammengefasst und häufige Herausforderungen benannt. In diesem Rahmen werden ebenfalls die im Digitalen Zwilling benötigten Relationen betrachtet und deren Bezug zum realen Produktionssystem. Abschließend werden diese Themen zusammengefasst und die wichtigsten Punkte für die weitere Arbeit hervorgehoben.

2.1 Ausgangssituation: Automatisierte Produktionssysteme

Im Laufe der letzten Jahre haben sich die Industrie und die industriellen Produktionssysteme immer weiterentwickelt. Begonnen mit der Mechanisierung Ende des 18. Jahrhunderts und der anschließenden Elektrifizierung im 19. Jahrhundert sind Produktionssysteme für die Massenproduktion entstanden. Der Einzug der Informations- und Kommunikationstechnik in den 70er Jahren haben dann zu den heutigen automatisierten, mechatronischen Systemen geführt [20]. Diese mechatronischen Systeme zeichnen sich durch eine höhere Genauigkeit und Performance auf Grund ihrer multi-disziplinären Zusammenarbeit aus [46]. Anfangs bezeichnete Mechatronik dabei nur die funktionale Erweiterung mechanischer Komponenten durch elektrische und elektronische Elemente [46]. Später bezeichnete dieser Begriff die Verbindung von Elementen aus dem Maschinenbau (Mechanik), der Elektrotechnik und der Informatik [47]. Derartige

Systeme können nur durch eine erfolgreiche Zusammenarbeit von Experten aus den verschiedenen Fachbereichen entwickelt werden [47]. Ein solches modernes automatisiertes Produktionssystem besteht aus dem Produktionssystem, das den technischen Prozess der Produktion durchführt, und einem Automatisierungssystem, das den Prozess überwacht und steuert [48]. Dieser Prozess wird durch Sensoren erfasst, die gesammelten Daten an das Automatisierungssystem übermittelt, das diese Daten interpretiert, die daraus erzeugten Informationen analysiert und entsprechende Steuerbefehle ausgibt. Über eine Vielzahl von Aktoren wirkt das Automatisierungssystem mit diesen Steuerbefehlen an den unterschiedlichen Stellen steuernd oder korrigierend auf den Prozess ein. Dadurch setzt der technische Prozess mit geringem manuellem Einfluss energetische und/oder materielle Eingangsgrößen in ein Ausgangsprodukt um (siehe Abbildung 2.1). Die Sensoren und Aktoren werden dabei im weiteren Verlauf als Peripherie oder allgemeiner als Komponente bezeichnet.

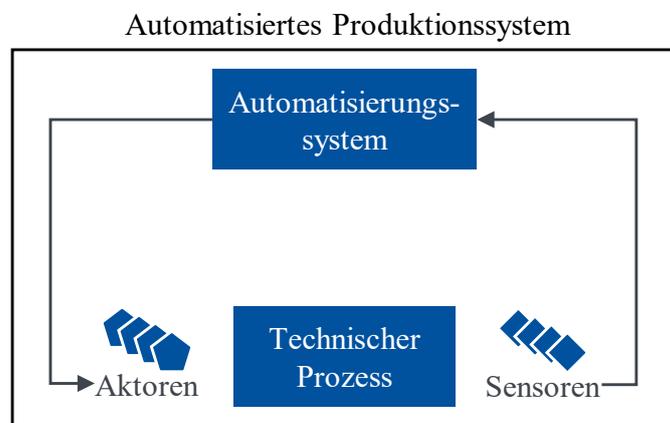


Abbildung 2.1: Schematische Darstellung eines automatisierten Produktionssystems und dessen Ein- und Ausgangsgrößen nach [49]

Die sogenannte vierte industrielle Revolution ist die aktuelle Weiterentwicklung dieser automatisierten Produktionssysteme und führt unter anderem zu verstärkter Kommunikation und vertikaler und horizontaler Vernetzung von Produktionssystemen. Sie wird nach der gleichnamigen Initiative der Bundesregierung Deutschland auch Industrie 4.0 genannt [20]. Die Industrie 4.0 sieht einen stärkeren Datenabgriff aus den Produktionssystemen beispielsweise über die vorhandene Automatisierungstechnik vor als bisher. Die erhöhte Kommunikation mit anderen Teilen der Produktionskette sowie die Sammlung und Analyse dieser Daten zur Schaffung von Mehrwerten, wie beispielsweise eine Fehlervorhersage, sind weitere Kernaspekte der Industrie 4.0. In diesem Rahmen wird von cyber-physischen Systemen (CPS) gesprochen oder auch von cyber-physischen Produktionssystemen (CPPS), wenn Produktionssystemen aus mehreren dieser CPS bestehen [50].

Im Folgenden wird kurz die Einteilung der Produktionssysteme zur Ausführung des technischen Prozesses angesprochen und für den weiteren Verlauf dieser Arbeit eingegrenzt, bevor das Automatisierungssystem und dessen Aspekte genauer beleuchtet werden. Dies wird ergänzt durch eine Betrachtung des Entwicklungsprozesses und den Eigenschaften von bestehenden Systemen.

2.1.1 Produktionssysteme

Als Produktionssystem wird in dieser Forschung das technische System zur Herstellung eines materiellen Produktes bezeichnet [18]. Die Produktionssysteme der produzierenden Industrie lassen sich den großen Industriezweigen, der Prozessindustrie oder der Fertigungsindustrie zuordnen. Die Prozessindustrie produziert in kontinuierlichen Prozessen einen stetigen Produktstrom, wie beispielsweise die Chemieindustrie. Diese großen, abgeschlossenen Systeme laufen mehrere Jahre bis Jahrzehnte unverändert, da bei dem Produkt kaum Änderungen auftreten und der technische Prozess deshalb gleichbleiben kann. Veränderungen der Anlagen finden meist als Erweiterungen oder parallele Aufbauten statt und werden beispielsweise zur Kapazitätserhöhung vorgenommen. [51]

Die Fertigungsindustrie hingegen ist geprägt durch die Produktion von einzelnen (diskreten) Produkten mit kurzen Lebenszyklen [51]. Die Produktionssysteme weisen eine Nutzungsdauer von wenigen Tagen bis zu mehreren Jahren auf [52]. Auf einem Produktionssystem werden dabei oft mehrere Produkte oder Produktversionen gefertigt. Die Anpassung zum Beispiel an neue Produkte erfolgt im Vergleich zur Prozessindustrie nicht durch eine Erweiterung, sondern durch die Rekonfiguration des bestehenden Produktionssystems. Die aktuellen Trends individueller Produkte und sinkenden Produktlebenszyklen erhöhen die Rekonfigurationshäufigkeit von Produktionssystemen für diskrete Produkte. Dadurch steigt die Häufigkeit von Anlagenstillständen und der Bedarf an Konzepten wie die des Digitalen Zwillinges.

Eine Methodik zur Unterstützung bei der digitalen Abbildung von Bestandsanlagen ist prinzipiell in beiden Industriezweigen hilfreich, jedoch wegen der unterschiedlichen Prozessarten und Komponenten nicht generell entwickelbar, sondern muss auf die Anlagentypen angepasst werden. Die weitere Arbeit konzentriert sich auf die Fertigungsindustrie und deren Produktionssysteme für diskrete Produkte aufgrund des höheren und weiter steigenden Bedarfs nach Unterstützung beispielsweise bei der Anlagen-Rekonfiguration für die Produktion individueller Produkte.

2.1.2 Automatisierungssysteme

Das Automatisierungssystem ist, wie bereits in Abbildung 2.1 dargestellt, über dem technischen Produktionsprozess angesiedelt und steuert diesen. Das Automatisierungssystem selbst ist ein komplexes System aus mehreren Komponenten. Klassischerweise wird es in die Automatisierungspyramide (siehe Abbildung 2.2) eingeteilt und dabei anhand der Nähe zum Prozess eingestuft. Der eigentliche Fertigungsprozess wird durch die Sensoren und Aktoren auf der Feldebene überwacht und manipuliert. Die Steuerungsaufgaben werden in mehrere Ebenen unterteilt und mit abnehmender Nähe zum Fertigungsprozess abstrakter. Die Steuerungsebene selbst weist überwiegend einen direkten Bezug zur funktionalen Ausgestaltung des Produktionssystems auf. Die darüber liegenden Ebenen sind auf die Koordination einzelner Steuerungen auf der Steuerungsebene, die Organisation des Materialflusses im Unternehmen und

auf die Datenaufbereitung für die Unternehmensführung spezialisiert. Die Abstraktion in höheren Ebenen ermöglicht es, den Überblick über die darunterliegenden, komplexen Prozesse zu behalten. Für eine digitale Nachbildung eines Produktionssystem sind allen voran die Feld- und Steuerungsebene bedeutsam, da diese Ebenen funktionale Elemente der Produktion beinhalten und letztere entsprechend beeinflussen. Die darüberliegenden Ebenen koordinieren das Zusammenspiel der Anlagenteile, den Materialfluss und die unternehmerische Kenngrößen auf einer größeren zeitlichen Auflösung, die keine direkte Aussage über die Funktionen der Anlage tragen. Aus diesem Grund beschäftigt sich die vorliegende Forschungsarbeit unter anderem mit den Elementen der Feld- und Steuerungsebene.

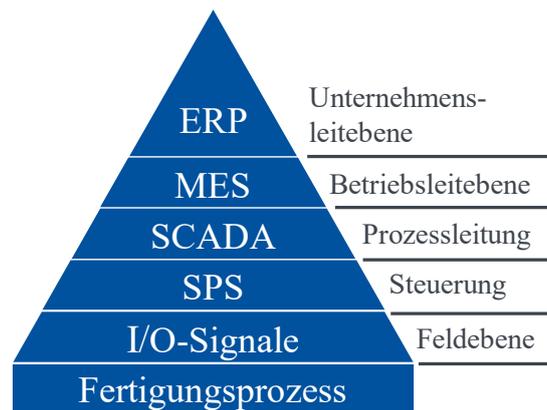


Abbildung 2.2: Aufteilung der Steueraufgaben in der Automatisierungspyramide nach [53]

Auf der Steuerungsebene werden verschiedene Rechnersysteme verbaut, die zur Verarbeitung der Messgrößen der Sensoren, der Abarbeitung des Steuerungsprogramms basierend auf diesen Messwerten und der anschließenden Ansteuerung der Aktorik verwendet werden. Abhängig von der Aufgabe und den Anforderungen kommen hier Speicherprogrammierbare Steuerungen (SPS), Soft-SPS, Industrie-PCs (IPC) oder auch Mikrocontroller zum Einsatz. Für die Steuerung des Produktionsprozesses sind Speicherprogrammierbare Steuerungen gängig, da diese durch eine feste Zykluszeit für die Ausführung des Steuerungsprogramms echtzeitfähig sind. Die SPS liest dabei zyklisch die neuen Sensorwerte ein, führt anschließend das Steuerprogramm aus und berechnet die neuen Stellwerte, die abschließend über die Ausgänge der SPS auf die Aktoren geschaltet werden. Durch den überwiegend modularen Aufbau der verfügbaren SPS können, abhängig von den verbauten Sensoren und Aktoren im Produktionssystem sowie den Steuerungsaufgaben, die benötigten Eingangs-, Ausgangs- und Kommunikationsmodule mit einem Prozessor kombiniert werden. Abgesehen von echten SPS werden zunehmend auch Soft-SPS verbaut, die keine dedizierte SPS-Hardware benötigen, sondern als Software auf einem normalen Computer oder einem IPC laufen. Diese IPC werden ebenfalls häufig für die Automatisierung eingesetzt beispielsweise für HMIs, andere Visualisierungen oder rechenintensivere Datenverarbeitungsaufgaben. Mikrocontroller hingegen sind in großen Produktionssystemen seltener im Einsatz und eher integriert in Subsystemen genutzt zur Vorverarbeitung eines Sensorsignals oder zur Anbindung eines einfachen Sensors an ein Kommunikationssystem. Insgesamt sind SPS die am weitesten verbreitete Steuerungstechnik

aufgrund der oben genannten Eigenschaften, aber auch wegen der Standardisierung über internationale Normen wie die EN 61131 oder EN 61499. Diese standardisieren die Programmiersprachen, Kommunikation, Formate, Funktionsbausteine und vieles mehr. [54–56] Aufgrund der weiten Verbreitung sowie der hohen Standardisierung der Steuerungsprogramme hinsichtlich der verwendeten Sprachen und Strukturen sind SPS als Informationsquelle über die funktionale Struktur von Brownfield-Produktionssystemen relevant und werden in der weiteren Forschung herangezogen.

Mit der aktuellen Entwicklung von vernetzten, selbständigen Produktionseinheiten und der Ausbreitung des Internets der Dinge wird erwartet, dass sich diese starre Automatisierungspyramide aufweicht und sich zu einer vernetzten, dezentralen Struktur entwickelt [57–59]. Dabei bleiben die einzelnen Steuerungen der feldnahen Ebenen erhalten und diese Forschungsarbeit, basierend auf der Feld- und Steuerungsebene, ist ebenfalls auf diese Struktur anwendbar. Die einzelnen Systeme bekommen zusätzliche Aufgaben und werden stärker vernetzt. Hierfür können bereits etablierte Kommunikationstechnologien wie Bussysteme (Profibus), Ethernet-basierte Netzwerke (ProfiNet, EtherCAT) oder neue Protokolle wie OPC UA, MQTT direkt mit den SPS eingesetzt werden. Der Vorteil einer solchen vernetzten, flacheren Struktur ist, dass die Anlagenteile modularer, autonomer und dadurch leichter austauschbar sind. Hierfür ist es jedoch notwendig, dass jeder Anlagenteil eine digitale Repräsentanz hat und mit anderen Anlagenteilen kommunizieren und verhandeln kann, um die komplette Herstellung eines Produktes zu ermöglichen.

Nachfolgende wird die systematische Erstellung der komplexen, mechatronischen Systeme betrachtet und die dabei entstehenden Artefakte.

2.1.3 Digitaler Entwicklungsprozess automatisierter Produktionssysteme – MBSE

Moderne automatisierte mechatronische Produktionssysteme entstehen durch das Zusammenwirken von Ingenieuren aus den verschiedenen Fachgebieten Mechanik, Elektrik und Software, wobei die Software einen zunehmenden Einfluss auf die Funktion des Produktionssystems gewinnt [60]. Das stellt eine grundlegende Herausforderung bei der Entwicklung der mechatronischen Systeme dar, da die Ingenieure ihre unterschiedlichen Perspektiven auf das System zusammenführen und konsistent halten müssen [61]. Das Konzept der modellbasierten Systementwicklung (MBSE) bietet hier Unterstützung, indem im Laufe der Entwicklung unterschiedliche Modelle erstellt, diese verknüpft und über die Fachbereiche kontinuierlich erweitert werden. Dadurch sind die entwickelten Produktionssysteme leichter verständlich als durch eine große Menge abstrakter Dokumente zur Beschreibung eines Systems, die in einem klassischen Entwicklungsprozess erstellt werden [26, 62]. Die in den Dokumenten enthaltene Informationen sind nur mit zusätzlichen manuellen Arbeitsschritten zwischen den

Fachbereichen übertragbar und in ihrer Menge und Komplexität schwer zu überschauen. Des Weiteren sind Anforderungen, Funktionen und Eigenschaften in Modellen direkt nachverfolgbar im Gegensatz zur dokumentenbasierten Entwicklung. Dies sind nach [26] Vorteile der MBSE und ermöglichen, dass spätere Entwicklungsschritte auf bestehende Modelle aufbauen und die bereits modellierten Informationen weiter nutzen können. So kann beispielsweise ein Automatisierungsingenieur die bestehenden CAD-Modelle durch Sensoren und Aktoren erweitern und mit der Steuerungslogik verknüpfen. Anschließend kann an diesen Modellen ein Roboterprogrammierer anknüpfen und die Roboterbahnen simulationsbasiert programmieren. Dadurch entfällt die Übergabe und Sichtung großer Informationsmengen, da bereits vorhandene Vorarbeiten durch die anderen Gewerke weitergenutzt werden können. Die Herausforderung der MBSE liegt im manuellen zeitlichen Aufwand, derartige Modelle zu erstellen und zu pflegen, sowie in den benötigten Softwaretools, welche die Ingenieure bedienen lernen müssen. Dieser zeitliche Mehraufwand wird bei großen, komplexen Produktionssystemen relativiert und durch die Vorteile kompensiert [26]. Als weiterer Mehrwert wird durch die MBSE eine Vielzahl von verknüpften Modellen erstellt, die den Kern für einen späteren Digitalen Zwilling bereitstellen [26]. Infolgedessen ist die Erstellung eines Digitalen Zwillings für Systeme, die nach dem MBSE-Ansatz entwickelt wurden, mit weniger zusätzlichen Aufgaben verbunden. Die aktuell bestehenden Systeme wurden jedoch nicht nach dem MBSE-Ansatz entwickelt. Somit sind diese Modelle nicht verfügbar, beziehungsweise es existieren nur wenige Modellierungsartefakte zum aktuellen System. Daher können bestehende Produktionssysteme in der Regel nicht von den Modellen aus der modellbasierten Systemmodellierung profitieren und müssen aufwendig nachträglich angefertigt werden.

2.1.4 Bestehende Produktionssysteme – Brownfield-Entwicklung

Dieses Kapitel dreht sich um bereits bestehende Produktionssysteme als besonderes Umfeld für die Weiterentwicklung oder Anpassung bestehender Systeme. Dieser Prozess wird als Brownfield-Entwicklung bezeichnet. Der Begriff des Brownfield wurde im Bauwesen geprägt und bezeichnet überbaute Fläche, die oft durch die vorherige Nutzung belastet ist [27]. Im industriellen Umfeld wird dieser Begriff äquivalent eingesetzt und bezeichnet aktiv eingesetzte Produktionssysteme, die nun modifiziert werden [28]. Die Belastung im Kontext der Produktionssysteme sind unter anderem die bestehenden Anlagenressourcen und deren aktive Nutzung für die gegenwärtige Produktion. Bei der Planung und Entwicklung im Brownfield müssen daher die bestehenden Ressourcen und Produktionsprozesse berücksichtigt werden, um einen möglichst störungsfreien Betrieb mit einem nahtlosen Übergang zu einem neuen Produktionssystem zu ermöglichen. Für eine erfolgreiche Arbeit im Brownfield und einen reibungsfreien Übergang werden umfangreiche Dokumentationen oder Modellierungen der Anlagen benötigt. Diese sind jedoch nur selten verfügbar oder durch vorangehende Modifikationen veraltet. Daher werden für die Aufgaben im Brownfield Experten aus verschiedenen Fachbereichen benötigt, welche die Anlage untersuchen und ein Reverse-

Engineering durchführen. Auch wenn das Brownfield Herausforderungen in sich birgt, ist die Neuerstellung ohne bestehendes System, im sogenannten „Greenfield“, nicht unbedingt besser. Es ist zwar kein Reverse-Engineering notwendig und es muss keine Rücksicht auf einen aktiven Produktionsprozess genommen werden, jedoch sind unter anderem hohe Investitionen für die Anschaffung der Produktionsressourcen und eventuelle Gebäude notwendig, welche die Kosten für die Anpassung bestehender Systeme überschreitet [30]. Aufgrund dieser finanziellen Gründe ist die Wiederverwendung und Erweiterung von bestehenden Produktionssystemen aus der Unternehmensperspektive notwendig. Darüber hinaus sprechen auch andere Faktoren wie der ökologische Fußabdruck für die Entwicklung im Brownfield [63]. Insgesamt wird die Anzahl der Brownfield-Projekte aufgrund des ökonomischen Aspekts weiter steigen [64]. Eine grundlegende Aufgabe ist dabei das Reengineering des Produktionssystems als digitale Modelle, die als Grundlage für die Planung genutzt werden können und einen Kernaspekt des Digitalen Zwillings bilden.

2.2 Zielvorstellung: Digitaler Zwilling

Das Konzept des Digitalen Zwillings zur umfassenden, digitalen Nachbildung von Produkten, Produktionssystemen oder Dienstleistungen wird derzeit umfangreich beforscht. In den Veröffentlichungen zum Digitalen Zwilling zeichnen sich verschiedene Anwendungsfälle ab, die unterschiedliche Mehrwerte für die Produktion bieten. Der Digitale Zwilling soll daher für möglichst alle Produktionssysteme verfügbar gemacht werden, auch für die bestehenden Produktionssysteme. Der Digitale Zwilling ist folglich die Zielvorstellung von dieser Forschung. Zur Standardisierung des Digitalen Zwillings haben sich verschiedene Partner aus Industrie und Forschung zusammengeschlossen, zum Beispiel zur Industrial Digital Twin Association (IDTA). Darüber hinaus wurden Formate zur Speicherung und Austausch definiert, wie AutomationML oder wie die Verwaltungsschale. Trotzdem wird der Digitale Zwilling in aktuellen Veröffentlichungen unterschiedlich abgegrenzt. Folglich liegt derzeit keine einheitliche Definition vor. In diesem Kapitel wird deshalb die Entstehung des Begriffs „Digitaler Zwilling“ erläutert, wie er allgemein definiert werden kann und welche Elemente er enthalten muss. Dies dient als Grundlage für die weitere Forschung und skizziert die Zielvorstellung der Methodik.

2.2.1 Entstehung des Konzepts „Digitaler Zwilling“

Das Konzept des Digitalen Zwillings entspringt der Luft- und Raumfahrt und wurde anfangs nicht digital angelegt, sondern als reale, physische Replikation eines „Systems“ beispielsweise einer Rakete durchgeführt. Dadurch konnte das System im Weltall eingesetzt und die baugleiche Nachbildung auf der Erde für Tests verwendet werden. Diese Tests erfolgten, ohne das reale System zu beeinflussen, in den kritischen Einsatz zu überführen oder dessen Einsatz zu unterbrechen. Dieses analoge Prinzip wurde durch die rapide steigende Leistung von verfügbaren Rechnersystemen zunehmend durch eine digitale, virtuelle Nachbildung ersetzt. Dadurch konnten

Ressourcen und Kosten gespart und realgetreue Tests durchgeführt werden. Im Rahmen der Apollo-Missionen wurden somit prinzipiell die ersten Digitalen Zwillinge eingesetzt, wenngleich sie noch nicht als solche bezeichnet wurden. Ebenso hat Grieves in [65] 2005 das Konzept des Digitalen Zwillings erstmals im Rahmen des Produkt Lebenszyklus Management (PLM) für die Fertigungsindustrie beschrieben, jedoch ohne diesen Begriff konkret zu verwenden, sondern den Digitalen Zwilling als digitale Repräsentation im virtuellen Raum benannt. Im Rahmen der NASA Roadmap [66] wurde 2010 dann der Begriff des Digitalen Zwillings erstmals geprägt und im Zusammenhang mit der Raumfahrt verwendet. Grieves hat kurze Zeit später in [67] den Begriff des Digitalen Zwillings auch im Zusammenhang mit der Fertigungsindustrie verwendet. Über die folgenden Jahre haben sich das Konzept und dieser Begriff weiter durchgesetzt. Dies ist unter anderem an den rapide steigenden Veröffentlichungszahlen zwischen 2016 und heute deutlich zu erkennen [68, 69]. In den unterschiedlichen Veröffentlichungen wurde das Konzept anwendungsfallspezifisch ausgelegt und infolgedessen lassen sich unter dem Begriff „Digitaler Zwilling“ eine Vielzahl verschiedener Definitionen auflisten. Die Veröffentlichungen belegen, dass das Konzept des Digitalen Zwillings ein bedeutsamer Teil der vierten industriellen Revolution ist [24] und den virtuellen Part eines cyber-physischen (Produktions-)Systems bildet [70, 71]. Der Digitale Zwilling bezeichnet dabei die virtuelle Repräsentation eines Produktes, eines Produktionssystems oder eines nicht materiellen Dienstes [72], der übergreifend über den kompletten Lebenszyklus von verschiedenen Anwendungen rentabel verwendet werden kann [73]. Der Mehrwert entsteht dabei durch den Fakt, dass das reale System nicht beeinflusst oder für Tests verwendet wird und parallel verschiedene Optionen auswirkungsfrei untersucht werden können. Elementar ist dabei die Konnektivität zwischen dem realen System und dem Digitalen Zwilling als dessen virtuelle Abbildung. So hat beispielsweise Kritzinger et al. in [22] eine Klassifizierung des Digitalen Zwillings in der Fertigungsindustrie anhand der Konnektivität vorgeschlagen und ausschließlich virtuelle Abbildungen mit einem bidirektionalen Austausch zwischen dem Asset und dem virtuellen Abbild als Digitalen Zwilling eingestuft. Darüber hinaus existieren weitere Literaturrecherchen wie zum Beispiel [74–76], in denen Veröffentlichungen zum Digitalen Zwilling und dessen Definitionen systematisch verglichen und nach dessen Anwendungsfall oder Anwendungsfeld eingeordnet werden. Im Folgenden wird die Vielfalt Digitaler Zwillinge für die weitere Forschung auf einen gemeinsamen Nenner reduziert und definiert.

2.2.2 Definition, Bestandteile und Architektur eines Digitalen Zwillings

Die Autoren der existierenden Veröffentlichungen verwenden den Begriff des Digitalen Zwillings für unterschiedliche Konzepte. Dabei lässt sich keine einheitliche Definition feststellen. Daher wird im Folgenden eine Definition des Digitalen Zwillings konzipiert, die die benötigten Eigenschaften und Bestandteile enthält und anschließend eine entsprechende Architektur

wiedergibt. Diese Bestandteile und die Definition bilden die Grundlage, auf die sich der Begriff des Digitalen Zwillinges im weiteren Verlauf bezieht. Ashtari et al. [4] haben für den Bereich der Produktionssysteme eine Literaturrecherche durchgeführt und eine umfassende Untersuchung der Gemeinsamkeiten bestehender Veröffentlichungen zusammengestellt. Die Inhalte decken sich mit den Ergebnissen von anderen aktuellen Literaturrecherchen [74–76] und werden im Folgenden zusammengefasst.

Kurz zusammengefasst ist der Digitale Zwilling das virtuelle Abbild oder auch Gegenstück zum realen, physischen Produktionssystem [48]. Der Grundbaustein eines Digitalen Zwillinges ist die digitale Replikation eines realen Assets (vgl. Abbildung 2.3). Diesen Begriff hat bereits Grieves in [65] 2005 gebraucht, als er das Konzept des Digitalen Zwillinges erstmalig beschrieben hat. In der Definition nach [4] umfasst die digitale Replikation verschiedene Modelle, die abhängig vom Anwendungsfall unterschiedlich sein können. Dabei haben die Autoren festgestellt, dass die digitalen Modelle übergreifend in nahezu allen Veröffentlichungen mit Bezug zum Konzept Digitaler Zwilling als Bestandteil benannt werden. Des Weiteren haben sie die Relationen zwischen den digitalen Modellen als grundlegender, jedoch herausfordernder Bestandteil identifiziert [26, 40, 77]. Diese werden in den unterschiedlichen Veröffentlichungen entweder explizit genannt oder impliziert und bilden zusammen mit den Modellen das reale Asset in der (virtuellen) Cyber-Schicht nach. Die Relationen umfassen dabei Beziehungen zwischen Modellen aus dem gleichen sowie unterschiedlichen Fachbereichen. Dieser Zustand einer digitalen Nachbildung ohne jegliche Kommunikation zum realen Asset deckt sich mit der niedrigsten Stufe nach [22] dem „digitalen Modell“. Aufbauend auf der digitalen Replikation werden drei weitere Eigenschaften benötigt, um daraus einen Digitalen Zwilling zu erhalten:

- Für die Nachbildung der dynamischen Vorgänge des realen Assets werden *simulierbare* Modelle benötigt. Dabei muss nicht jeder Digitale Zwilling eine Simulationssoftware beinhalten, sondern lediglich die simulierbaren Modelle, da dies ansonsten abhängig vom Lizenzmodell kostspielig werden kann. [78]
- Die digitale Replikation muss an die Veränderungen des realen Assets angepasst werden, sodass die (statische) Nachbildung mit dem realen Asset über den Lebenszyklus des Produktionssystems *synchron* gehalten wird. Dies ist unerlässlich, da an aktiv eingesetzten Systemen regelmäßig kleine Anpassungen und Optimierungen durchgeführt werden [79, 80].
- Es wird außerdem eine aktive Datenerfassung aus der realen Anlage benötigt, um die dynamischen Vorgänge in der Simulation den realen Vorgängen korrekt nachzuführen und mehr als nur theoretische Vorgänge zu simulieren. Darüber hinaus werden diese Daten auch für datenbasierte Analysen oder KI-Algorithmen benötigt, welche zur Verbesserung des Digitalen Zwillinges und dessen Modelle oder auch für Services nach außen eingesetzt werden (z.B. Anomaliedetektion innerhalb von Modellen).

In der Abbildung 2.3 ist die Zusammensetzung eines Digitalen Zwillings aus der digitalen Replikation und diese drei Eigenschaften (Ellipsen) als integraler Bestandteil dargestellt. Diese Definition deckt sich auch mit der Einteilung nach der Kommunikation [22] zwischen dem Asset in der Cyber-Schicht und in der physischen Schicht. Die Kommunikation vom realen Asset zum Digitalen Zwilling ist hier durch die Eigenschaften deutlich zu erkennen. Die entgegengesetzte Kommunikation wird durch die auf dem Digitalen Zwilling aufsetzenden Anwendungen initiiert.

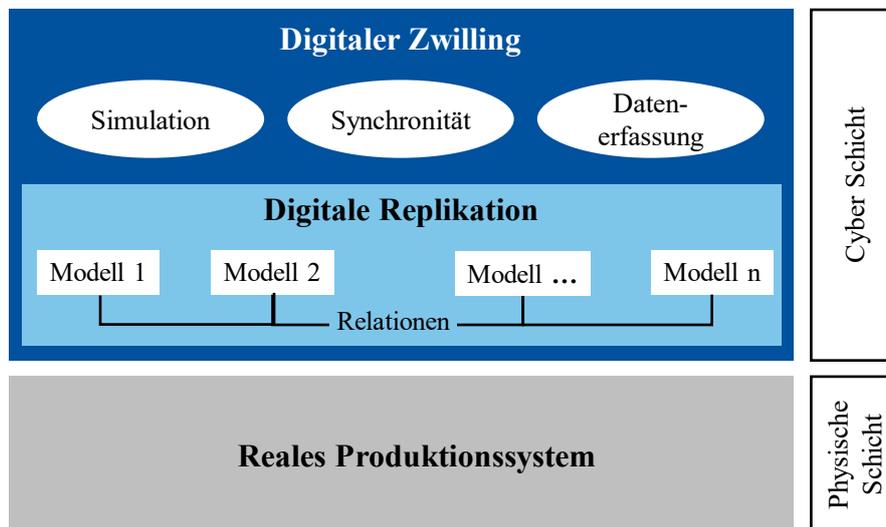


Abbildung 2.3: Komponenten des Digitalen Zwillings nach [4]

Um diese Definition mit den genannten Bestandteilen und Eigenschaften umzusetzen, wurde in [4] eine Architektur vorgeschlagen. Diese wird hier kurz diskutiert und damit die Fokussierung der Arbeit verdeutlicht. Abbildung 2.4 zeigt die vorgeschlagene Architektur, welche die Modelle und Relationen als Bestandteile der Digitalen Replikation beinhaltet. Darüber hinaus werden die drei Eigenschaften (Simulierbarkeit, Synchronität und aktive Datenerfassung) durch entsprechende Schnittstellen und Speichermöglichkeiten (Betriebsdaten⁴) berücksichtigt. Zusätzlich muss eine derartige Architektur Elemente zur eindeutigen Identifikation des Digitalen Zwillings (ID), zur Speicherung von Metadaten und zur Verwaltung der Versionen über den Asset-Lebenszyklus enthalten. In grau sind außerdem Komponenten hinzugefügt, die den Digitalen Zwilling über eine Nachbildung der Anlage hinaus erweitern und mit Intelligenz ausstatten. Allen voran gehören dazu intelligente Algorithmen, welche aus dem Bereich der datengetriebenen KI-Algorithmen (z.B. Anomaliedetektion [81]) oder aus dem Bereich der Expertensysteme (z.B. Rekonfigurationsmanagement [50]) entstammen können. Die Algorithmen verwenden die Komponenten des Digitalen Zwillings, wie die Betriebsdaten, Modelle mit deren Relationen und/oder den Versionsverlauf, um anderen (externen) Systemen und Benutzern höherwertige Informationen zu bieten. Darüber hinaus können sie nicht nur durch externe Akteure genutzt werden, sondern können auch zusätzliche Informationen oder Fähigkeiten innerhalb des Digitalen Zwillings schaffen. Zum Beispiel bedarfsbezogen eine Anpassung der Modellgranularität vornehmen, um Simulationsdauern zu optimieren, Lücken zwischen der

⁴ Als Betriebsdaten werden im Betrieb des Produktionssystems erzeugte Aktor- und Sensorwerte bezeichnet.

Simulation und Realität zu schließen oder Betreibern Ursachen für Ungenauigkeiten aufzuzeigen [82]. Eine weitere Form der intelligenten Algorithmen wird in [83] diskutiert. Die Autoren sprechen anfangs vom modellbasierten Digitalen Zwilling, der durch die Datenerfassung über den Lebenszyklus eine datengetriebene Komponente erhält. Dabei spielt beim datengetriebenen Digitalen Zwilling der Einsatz von Algorithmen des maschinellen Lernens eine Rolle zur Verbesserung der Modellierung durch die Abbildung nicht oder ungenau abbildbarer Informationen durch klassische Modelle.

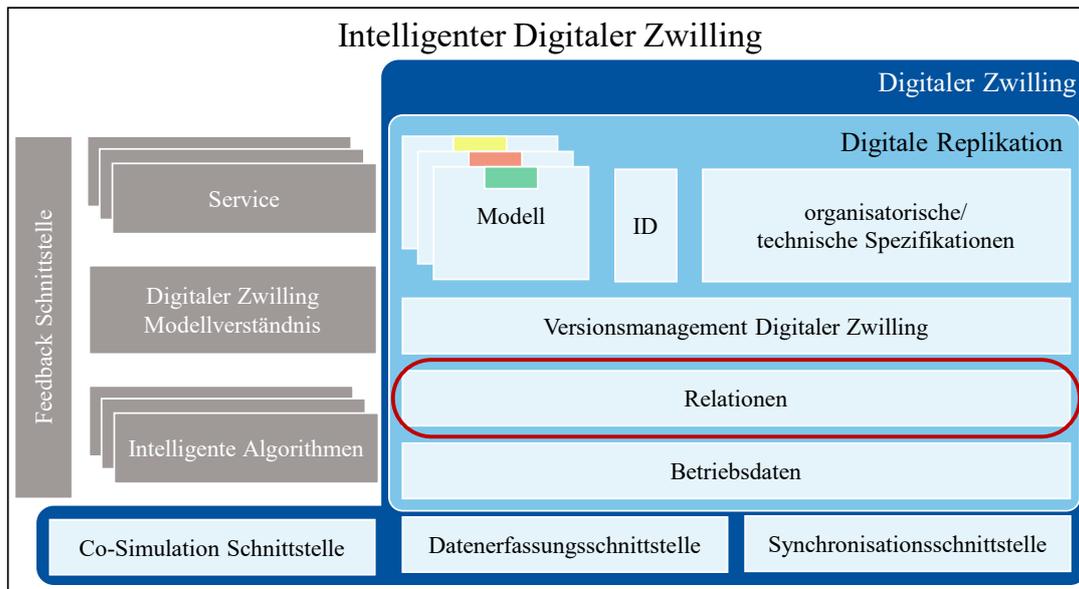


Abbildung 2.4: Architektur eines (intelligenten) Digitalen Zwillings nach [4]

Des Weiteren sind optional verschiedene Services im intelligenten Digitalen Zwilling vorhanden, welche die Funktionalitäten des realen Assets, wie Bohren oder Fördern, als Dienste abbilden und beispielsweise anderen Digitalen Zwillingen zur Verfügung stellen. Diese Services kommunizieren über die Feedback-Schnittstelle mit dem realen Asset und schließen damit die bidirektionale Kommunikation zwischen der physischen und der Cyber-Ebene. Abgesehen von der Feedback-Schnittstelle sind ebenfalls Schnittstellen für die Datenerfassung vom realen Asset, für die Synchronisation des Digitalen Zwillings mit dem realen Asset und die Kommunikation mit anderen Digitalen Zwillingen notwendig. Die Verwaltungsschale (VWS)⁵ ist eine aktuell prominente Umsetzungsmöglichkeit der Schnittstellen, die durch den standardisierten Austausch und der Kommunikation einen Austausch der Daten zwischen verschiedenen Digitalen Zwillingen, Engineering Tools und dem realen System ermöglicht. Ein Digitaler Zwilling in einer derartigen Umsetzung kann als Grundlage für eine Vielzahl von Anwendungen dienen, die auf die Daten, Modelle und Algorithmen des Digitalen Zwillings zugreifen können. Im nächsten Kapitel werden die Relationen des Digitalen Zwillings im Detail betrachtet.

⁵ Besser bekannt unter dem englischen Begriff: Asset Administration Shell (AAS)

zwischen Elementen aus verschiedenen Hierarchieebenen oder Teilästen der CAEX-Struktur verwendet. Diese Relationen bilden unter anderem funktionale oder elektrische Verbindungen ab, die nicht durch ein „besteht-aus“ ausgedrückt werden können.

2.2.4 Anwendungsfälle des Digitalen Zwillings und deren Mehrwerte sowie Herausforderungen

In der Literatur werden bereits zahlreiche Anwendungen aufgeführt, die auf den Einsatz eines Digitalen Zwillings aufbauen und die Informationen aus diesem nutzen. Dabei stammen nach Fuller et al. [86] die meisten Veröffentlichungen und Anwendungen zum Digitalen Zwilling aus dem Feld der Fertigungsindustrie. In diesem Abschnitt werden exemplarisch übliche Anwendungen im Bereich der Produktionssysteme diskreter Produkte vorgestellt und deren Mehrwerte diskutiert. Jones et al. [76] haben in einer Literaturrecherche festgestellt, dass die meisten Veröffentlichungen sich mit Anwendungen zur Realisierungs- und Betriebsphase beschäftigen. Dies deckt sich mit den Erkenntnissen von Kritzinger et al. [22], die zu dem Ergebnis gekommen sind, dass der überwiegende Anteil den Anwendungen aus dem Bereich der Produktionsplanung und Steuerung zuzuschreiben ist, gefolgt von Wartungsanwendungen. Eine Hauptanwendung für Wartungsaufgaben in der Fertigungsindustrie ist die vorausschauende Wartung⁶, die in Kombination mit dem Digitalen Zwilling eine breite Aufmerksamkeit erfährt [87]. Nach van Dinter et al. [87] zielen die meisten Anwendungen auf die Vorhersage der nutzbaren Restlebensdauer ab und sehen hier durch den Einsatz des Digitalen Zwillings Potenziale, diese effizienter vorherzusagen. Eine weitere weit verbreitete Anwendung ist die virtuelle Inbetriebnahme, die nach [88] nicht nur in der Wartung einen zeitlichen Mehrwert bietet, sondern auch für den Umbau oder die Erweiterung. Die virtuelle Inbetriebnahme kann die benötigte Zeit eines Umbaus oder einer Inbetriebnahme signifikant um bis zu 75% reduzieren [21], wenn die benötigten Modelle [89] zur Verfügung stehen. Der Digitale Zwilling kann hierzu die benötigten Modelle bereitstellen und damit die Anfangsinvestitionen und zeitlichen Aufwände für die Modellierung reduzieren. Durch die tendenziell steigende Rekonfigurationshäufigkeit im Brownfield wird diese Anwendung weiter an Relevanz gewinnen.

Bei allen diesen Anwendungsfällen entsteht für den Betreiber ein zeitlicher und kostentechnischer Vorteil dadurch, dass das reale Produktionssystem nicht für die Testphase benötigt und eventuell stillgelegt werden muss, sondern durch den Digitalen Zwilling virtuell nachgebildet wird. Folglich werden das Risiko eines Defekts durch Fehler und die Wiederanlaufzeit des Produktionssystems reduziert sowie gleichzeitige, beschleunigte Tests parallel zum Produktivbetrieb ermöglicht.

Die Verwendung eines Digitalen Zwillings für ein Produktionssystem erzeugt aber auch Herausforderungen. Viele davon treten beim Betrieb und der Aufrechterhaltung eines Digitalen Zwillings auf, unter anderem bei der Synchronisierung mit dem realen Asset [90], bei der

⁶ Besser bekannt unter dem englischen Begriff: Predictive Maintenance

Integration in eine IT Infrastruktur und der Haltung der Daten [86] oder auch bei der Kombination mehrerer Digitaler Zwillinge für ein System aus mehreren Subsystemen⁷ [91]. Für Brownfield-Entwicklungsprozesse besteht des Weiteren die Herausforderung, nachträglich die Digitale Replikation des Brownfield-Produktionssystems als Basis des Digitalen Zwillings zu erstellen. Dies ist ein zeitintensiver Prozess und durch die notwendige Zusammenarbeit von Experten aus verschiedenen Fachbereichen ein fehleranfälliger Prozess [92].

2.3 Zusammenfassung der Ausgangssituation und der Zielvorstellung

Die beschriebenen Grundlagen lassen sich wie folgt für die weitere Arbeit zusammenfassen. Die Brownfield-Produktionssysteme der Fertigungsindustrie sind überwiegend mit SPS automatisierte, mechatronische Systeme, welche über einen längeren Zeitraum in Betrieb sind. Diese mechatronischen Systeme sind ein komplexes Zusammenspiel aus verschiedenen Fachbereichen, welche heutzutage in einem modellbasierten Entwicklungsprozess entworfen werden. Die Weiterentwicklung von bestehenden Produktionssystemen, sogenannte Brownfield-Entwicklung, ist gängiger als der komplette Neuentwurf eines Produktionssystems. Brownfield-Produktionssysteme werden über die Nutzungsdauer ein oder mehrfach rekonfiguriert, wobei die Dokumentation selten (vollständig) angepasst wird oder nicht mehr existiert. Modellierungen des Produktionssystems sind mindestens ebenso selten vorhanden und aktuell wie Dokumentationen. Durch innere und äußere Faktoren, wie die Nachfrage individueller Produkte oder der technische Fortschritt im Rahmen von Industrie 4.0, müssen Brownfield-Produktionssysteme vermehrt rekonfiguriert werden. Die Unterstützung der Rekonfiguration und weitere Anwendungsfälle, beispielsweise die vorausschauende Wartung oder virtuelle Inbetriebnahme, benötigen digitale Modelle und deren Relationen, welche bisher manuell erstellt werden müssen. Das manuelle Reengineering der Systeme und die Erstellung der digitalen Replikation geht einher mit einem großen Aufwand und weist in der Regel aufgrund der Größe und Komplexität der Produktionssysteme ein hohes Fehlerpotenzial auf. Trotz allem wird dieser Aufwand in Kauf genommen, da die digitale Replikation die Grundlage eines Digitalen Zwillings bildet und die erfolgreiche Anwendung des Digitalen Zwillings durch verschiedene Anwendungsfälle perspektivisch den Aufwand relativieren kann.

Im folgenden Kapitel werden daher bestehende Ansätze betrachtet, die für die Unterstützung des Erstellungsprozesses Digitaler Zwillinge von Brownfield-Produktionssystemen verwendbar sind. Der Fokus liegt dabei auf der digitalen Replikation als Kernaspekt des Digitalen Zwillings und speziell den dafür relevanten Relationen. Die zusätzlichen Eigenschaften und Fähigkeiten eines Digitalen Zwillings sind unabhängig von der digitalen Replikation und vom Produktionssystem nachträglich zu ergänzen.

⁷ Besser bekannt unter dem englischen Begriff: Systems-of-systems

3 Stand der Wissenschaft und Technik

In diesem Kapitel erfolgt die Evaluierung des aktuellen Stands der Wissenschaft und Technik mit dem Fokus auf die Unterstützung der Erstellung eines Digitalen Zwillings für bestehende Produktionssysteme. Die verfügbare Literatur zum Digitalen Zwilling nimmt zwar derzeit stark zu, aber die zugrunde liegende Definition des Digitalen Zwillings ist zwischen den Veröffentlichungen sehr unterschiedlich. In dieser Arbeit wird die in Kapitel 2.2.2 ausgewiesene Definition des Digitalen Zwillings zugrunde gelegt und die vorgefundenen Publikationen anhand dieser bewertet. Dabei werden verschiedene Ansätze zur Modellerstellung und zur Nachbildung der Relationen zwischen den Modellen betrachtet.

In der stetig wachsende Anzahl an Veröffentlichungen zum Digitalen Zwilling finden unterschiedliche Kernthemen Beachtung [76], wobei die Erstellung eines Digitalen Zwillings für bestehende Produktionssysteme der Fertigungsindustrie nicht erwähnt wird. Aus dem Bereich der Prozessindustrie gibt es Veröffentlichungen, in denen Methodiken zur Erstellung Digitaler Zwillinge für Brownfield thematisiert werden [93, 94]. Der angedachte Einsatzzweck dieser Methodiken ist identisch zu der hier erfolgten Forschung und soll den zeitlichen Aufwand sowie das benötigte Wissen für die Erstellung der digitalen Replikation herabsetzen, um die Nutzung Digitaler Zwillinge zu erleichtern. Diese Ansätze beinhalten eine vernetzte Integration von 3D-Modellen, Fließschemata (PI&D), Simulationsmodellen sowie vom Steuerungscode und erzeugen eine digitale Replikation dieser Anlagen. Diese Methodiken sind relativ ausgereift und bereits in großen Systemen erprobt [94]. Auf Grund des unterschiedlichen Aufbaus sowie der Schwerpunkte der Produktionssysteme in der Prozess- und der Fertigungsindustrie sind diese Methodiken jedoch nicht übertragbar. Im Bereich der Fertigung diskreter Produkte gibt es bisher keine Methodiken, die ebenso weit ausgereift sind wie die der Prozessindustrie. Es existieren wenige Veröffentlichungen, in denen die Autoren beanspruchen einen Digitalen Zwilling für bestehende Produktionssysteme zu erzeugen [92, 95–98]. In diesen Publikationen werden jedoch entweder ein manueller Vorgang und keine (teil-)automatisierte Methodik beschrieben [95, 96], einzelne Teilaspekte und kein Digitaler Zwilling nach der vorliegenden Definition erstellt [98] oder die Autoren beschäftigen sich mit der Erschaffung Digitaler Zwillinge bei einer modellbasierten Entwicklung neuer Produktionssysteme [92, 97].

Daher werden im Weiteren verfügbare Ansätze betrachtet, die für die Erzeugung einzelner Aspekte des Digitalen Zwillings genutzt werden können. Anschließend werden Ansätze zur Erstellung der Verknüpfungen von Modellen verschiedener Fachbereiche sowie Technologien zur Verknüpfung von Modellen zu einem Digitalen Zwilling begutachtet. Zuletzt erfolgt die Zusammenfassung der Ansätze aus dem Stand der Wissenschaft und Technik sowie die Bewertung hinsichtlich der Anforderungserfüllung. Anhand dieser Beurteilung wird die wissenschaftliche Forschungslücke aufgezeigt.

3.1 Ansätze zur Erfassung und Modellierung bestehender Produktionssysteme

Es liegen bereits Ansätze zur Erfassung und Modellierung von bestehenden Produktionssystemen vor. In manchen Veröffentlichungen wird hierzu der Anspruch erhoben, für die Erstellung Digitaler Zwillinge anwendbar zu sein. Dies wird im Verlauf dieses Kapitels diskutiert und mit der betrieblichen Situation von Brownfield-Produktionssystemen sowie mit der Definition des Digitalen Zwillings aus den Kapiteln 1.5 und 2.2 bewertet. Die überwiegende Anzahl der Ansätze ist ursprünglich nicht primär für die Erstellung Digitaler Zwillinge gedachte, sondern ist für eine abgegrenzte Teilaufgabe ausgelegt. Die erzeugten Artefakte der Ansätze sind jedoch für einen Digitalen Zwilling erforderlich und verwendbar, da sie unterschiedliche Elemente einer digitalen Replikation abdecken. Die Ansätze lassen sich hinsichtlich der Art ihrer Datenquelle gruppieren und werden im Folgenden in diesen Gruppen vorgestellt und diskutiert.

3.1.1 Optische Verfahren

Die Ansätze, die ein Produktionssystem mit optischen Verfahren inspizieren, bilden die größte Gruppe [99]. Diese Verfahren zielen darauf ab, das CAD-Modell aufzubauen und die Mechanik des Produktionssystems nachzubilden. Abgesehen von diesen optischen Erfassungsmethoden existieren noch andere Verfahren zur Erzeugung des CAD-Modells, die auf einer mechanischen Abtastung, Magnetfeldanalysen oder Akustik basieren [100]. Diese sind jedoch selten eingesetzt [100] und werden daher hier nur erwähnt jedoch nicht weiter betrachtet. Die Datenquelle für die optischen Verfahren bildet das physische Produktionssystem, das entweder mit speziellen Geräten oder über normale Kamerabilder erfasst wird. Die spezialisierten Geräte bauen auf eine Abtastung der Oberfläche mit einem Laser oder mit mehreren Kamerasystemen, teilweise mit unterschiedlichen Wellenlängen. Diese Systeme, speziell die laserbasierten Geräte, arbeiten sehr präzise hinsichtlich der Auflösung der Kontur, jedoch können sie alleinstehend keine Farben erfassen. Die Farben müssen aus separaten Bildern nachgeliefert werden [101]. Die Erfassung mit einer oder mehreren Kameras wird Fotogrammetrie genannt. Die Fotogrammetrie berechnet das 3D-Modell aus vielen Bildern des zu erfassenden Objektes aus verschiedenen Richtungen und Winkeln [102]. Dieses Verfahren ist weniger genau als mit Laserscanner. Dafür ist es kostengünstiger und die Farben werden direkt miterfasst [103]. Unabhängig von der Erfassungsmethode wird eine monolithische Punktwolke gebildet, die segmentiert und in CAD-Features aufgeteilt werden muss, bevor das Ergebnis im weiteren Engineering eingesetzt werden kann [104, 105]. Die korrekte Segmentierung und speziell die korrekte Aufteilung kleiner Teile mittels Relationen in die CAD-Struktur ist noch eine Herausforderung [106] für die mehrschrittige Vorgehensweisen entwickelt werden [98, 107]. Die optischen Verfahren erstellen CAD-Modelle und bilden dabei lediglich die Informationen über den Fachbereich der Mechanik ab. In Kombination mit der Segmentierung können auch Teile der fachbereichsinternen Eltern-Kind-

Relationen nachgebildet werden. Dabei werden jedoch nur die von außen sichtbaren Komponenten berücksichtigt. Die Erfassung eines Produktionssystems bedarf vieler Aufnahmen aus verschiedenen Winkeln, um den Grad der Abschattungen zu reduzieren und um ein einigermaßen vollständiges Modell zu erhalten. Infolgedessen werden für Laserscans Betriebsunterbrechungen benötigt [102]. Die Autoren mancher Verfahren behaupten damit Digitale Zwillinge zu erstellen [98], auch wenn nur ein mechanisches Modell geschaffen wird. Dies erfüllt nicht die formulierte Definition eines Digitalen Zwillings nach Kapitel 2.2.2 und deckt einen beschränkten Teil der Modelle der Digitalen Replikation ab.

3.1.2 Dokumentenbasierte Verfahren

Mit dokumentenbasierten Verfahren werden entweder papierförmige Dokumente oder digitale Dokumente wie zum Beispiel PDFs verarbeitet. Trotz des digitalen Engineerings und der modellbasierten Entwicklung werden Informationen zwischen verschiedenen Parteien und Fachbereichen über Dokumente ausgetauscht [108, 109]. Dies gilt für Schaltpläne, Rohrleitungs- und Instrumentenfließschemata, Steuerungscode oder auch für Anforderungsdefinitionen [109]. Diese Dokumente sind unter anderem vor Ort bei der Inbetriebnahme des Produktionssystems erforderlich und es werden beispielsweise kurzfristige Änderungen bei der Inbetriebnahme händisch in die Schaltpläne eingezeichnet [110]. Diese Änderungen müssen zurückgespielt werden, um Inkonsistenzen zwischen dem realen Produktionssystem und dessen Modelle zu vermeiden [111]. Die dokumentenbasierte Verfahren für das Reverse-Engineerings, speziell für den Schaltplan [110], sind von großer Relevanz bei der Erstellung Digitaler Zwillinge. Dieses Konzept kann ebenfalls genutzt werden, um die Informationen der Dokumente zu extrahieren und die Modelle und Relationen aus dem Fachbereich der Elektrik zu generieren. Da die Dokumente fachbereichsbezogen angelegt werden, ist hier auch nicht mit einer Nachbildung der fachbereichsübergreifenden Relationen zu rechnen. Der große Vorteil der Dokumentation liegt darin, dass keine Untersuchung des Produktionssystems notwendig ist und daher kein Stillstand des Betriebs einzukalkulieren ist. Eine Dokumentation birgt jedoch auch immer die Gefahr in sich, veraltet zu sein und nicht mehr den aktuellen Zustand des Brownfield-Produktionssystems wiedergeben zu können [35, 112].

3.1.3 Netzwerkbasierte Verfahren

Die netzwerkbasierten Verfahren werden für die Analyse der Metadaten oder des Inhalts auf dem Kommunikationsnetzwerk verwendet. Sie werden bisher eher selten für die Erstellung von Modellen eingesetzt und daher nur in einer begrenzten Anzahl an Publikationen beschrieben. Die Anwendung der Verfahren, welche die Metadaten des Kommunikationsnetzwerks analysieren, [113, 114] erzeugt die Liste der Ressourcen eines Produktionssystems. Eine Einschränkung dieser Verfahren besteht darin, dass bei der Erstellung der Ressourcenliste eine Planungsbibliothek benötigt wird, die zum Abgleich der Metadaten der aktuellen Teilnehmerdaten herangezogen

wird. Daher ist dieses Verfahren nur geeignet, wenn eine entsprechende Bibliothek bereits existiert und kein individuelles Produktionssystem vorliegt. Die zweite Gruppe der netzwerkbasieren Verfahren basiert auf der Analyse des Kommunikationsinhalts auf einem industriellen Kommunikationsnetzwerk [115]. Die Autoren stellen dabei fest, dass alle Sensoren und Aktoren entweder direkt oder indirekt über industrielle Kommunikationsnetzwerke an eine zentrale SPS angeschlossen sein müssen. Das Ziel dieses Verfahrens ist es den aktuellen Wert eines veränderlichen Parameters über die Kommunikation mit der Steuerung zu identifizieren, um die Simulationsmodelle an Veränderungen im realen Produktionssystem anzupassen. Diese Vorgehensweise erzeugt eine aktualisierte Parametrisierung der Simulationsmodelle eines Produktionssystems basierend auf den bestehenden Modellen dieses Systems. Es werden jedoch dadurch keine neuen, fehlenden Modelle ergänzt.

3.1.4 SPS-Codebasierte Verfahren

Es gibt mehrere verschiedene SPS-Codebasierte Verfahren für verschiedene Zwecke. Denn SPS sind die am häufigsten eingesetzte Steuerungen in der industriellen Fertigung [54]. Darunter befinden sich Verfahren, auch kommerzielle, zur Analyse des SPS-Codes, um diesen für Entwickler aufzubereiten, besser verständlich zu machen oder die unterschiedlichen Möglichkeiten sowie deren Folgen herauszuarbeiten [116]. In einer Methode werden unter anderem Graphen verwendet, um die Aufrufe innerhalb des SPS-Codes (Call Hierarchy) aufzubereiten [55]. Ein anderer Ansatz rekonstruiert aus dem SPS-Code UML-Modelle, um die Wiederverwendbarkeit zu erhöhen und die Rekonfiguration zu erleichtern. Sie liefert dadurch eine Modellierung des SPS-Codes [117]. Diese Modelle können beispielsweise durch den, auf UML und SysML basierenden, modellbasierten Ansatz von Obermeier et al. [118] zur objektorientierten Programmierung von SPS verwendet werden. So kann die Integration in den modellbasierten Entwicklungsprozess geschaffen werden. Diese Ansätze sind jedoch auf die Software beschränkt und zielen auf die Unterstützung des Software-Entwicklungsingenieurs ab. Die Ankerpunktmethode von Ashtari [119], [120] behandelt als einzige Methode die Erstellung von aktuellen Modellen und fachbereichsübergreifenden Relationen als Basis für einen Digitalen Zwilling. Diese Methode setzt jedoch voraus, dass die Namenskonvention der Ankerpunktmethode bei der Implementierung des SPS-Codes berücksichtigt wurde und, dass (Teil-)Modelle des Digitalen Zwillings existieren. Sie ist daher nicht auf alle Brownfield-Produktionssysteme anwendbar. Das primäre Ziel dieser Methode ist es, einen bestehenden Digitalen Zwilling an die Modifikationen aus dem laufenden Betrieb anzupassen. Sie ist daher der Erstellung eines Digitalen Zwillings nachgelagert zu verorten und realisiert dessen Synchronitätseigenschaft.

3.1.5 Sonstige Ansätze ohne eindeutige Informationsquelle

Abgesehen von den bereits beschriebenen Ansätzen, die genau eine Informationsquelle nutzen, existieren Ansätze, die mehrere Quellen verwenden und nicht konkret zugeordnet werden können. In diesem Kapitel werden diese betrachtet und mögliche Zuordnungen zu den vorherigen Quellen aufgezeigt.

Der Ansatz in [121] kommt aus der Prozessindustrie und leitet aus Teilen der vorhandenen Entwicklungsdaten weitere Modelle ab beziehungsweise integriert diese in einen Digitalen Zwilling. Dabei müssen die grundlegenden Modelle aus der Planung manuell erstellt werden oder bereits vorhanden sein. Daraus leiten die Autoren dynamische Simulationsmodelle ab, bereichern diese mit 3D-Modellen, Rohrleitungs- und Instrumentenfließschemata an und erzeugen Steuerungsanwendungen. Die Autoren in [122] verwenden ebenso bestehende Modelle als Quelle für die Erstellung einer Simulation von kontinuierlichen Prozessen. Dabei beschreiben sie die Kombination und das Mapping von Modellen aus dem Bereich der Simulation und Planung, um automatisiert ein Modell für die virtuelle Inbetriebnahme zu erzeugen. Dazu wählen Experten manuell Komponenten aus Bibliotheken aus, die für die Simulationsgenerierung automatisiert zusammengeführt werden. Vergleichbar dazu stellen die Autoren in [123] einen teilautomatisierten Ansatz vor, um CAD-Modelle sowie Verhaltensmodelle anhand ihrer Referenz-IDs zu einem Simulationsmodell zusammenzufügen und dieses als AML-Datei zu exportieren. Dieser Ansatz ist in erster Linie für die frühen Entwicklungsphasen neuer Systeme gedacht und kann, aufgrund der benötigten Modelle, nur begrenzt für die Erstellung von Modellen und Relationen von bestehenden Produktionssystemen genutzt werden.

Als Quelle für die Ansätze in [97, 124] dient die Konfiguration und der Steuerungscode der Montageroboter, um daraus deren physische Anordnung als Layout zu rekonstruieren. Dies ist nach [97] eine relevante Information für die Integration neuer Fahrzeuge und ein integraler Bestandteil eines Digitalen Zwillings. Die Autoren beanspruchen, dass sie dadurch einen Digitalen Zwilling erstellen können. Nach der Definition des Digitalen Zwillings, auf die diese Arbeit aufbaut, ist das Layout nur ein Modell des Digitalen Zwillings und genügt nicht, um von einem Digitalen Zwilling eines bestehenden Produktionssystems zu sprechen.

Mit dem letzten Ansatz werden die Schaltpläne und eine Modellbibliothek kombiniert, um den Entwicklungsprozess zu beschleunigen [125]. Dieses Ziel soll erreicht werden, indem aus den Schaltplänen mit Hilfe einer Komponentenbibliothek Verhaltensmodelle für die virtuelle Inbetriebnahme erzeugt werden. Die Generierung einer Simulation aus Schaltplänen kann helfen, den Modellpool eines Digitalen Zwillings zu vervollständigen. Jedoch kann dadurch nicht der komplette Modellpool inklusive der Relationen zwischen den Modellen erstellt werden.

3.2 Ansätze zur Verknüpfung von Modellen verschiedener Domänen

Die vorhergehenden Ansätze zur Erfassung und Modellierung bestehender Produktionssysteme behandeln einzelne Modelle oder nutzen vorhandene Relationen, um daraus Modelle abzuleiten oder zu aktualisieren. Jedoch resultiert aus keinem dieser Ansätze explizit die benötigten Relationen. In diesem Kapitel werden die Ansätze beurteilt, in denen die Erstellung von Relationen zwischen Modellen eines Digitalen Zwillings thematisiert wird. Darüber hinaus werden bestehende Technologien zur Abbildung der Relationen betrachtet.

3.2.1 Ansätze zur Erstellung der Relationen eines Digitalen Zwillings

Spezialisierte Ansätze, die explizit die Nachbildung der Relationen in einem Digitalen Zwilling behandeln sind derzeit nicht bekannt. Bei einigen der genannten Ansätze [100, 105, 106, 108, 109, 111, 113, 114, 121–123, 125] aus Kapitel 3.1 werden im Zuge der Modellerstellung implizit Teile der Relationen generiert, da sie ein Modell oder Rohdaten aus einem Fachbereich als Quelle verwenden und daraus ein Modell eines anderen Fachbereichs erzeugen. Dabei wird zwar implizit die Relation zwischen diesen Modellen für die Erstellung genutzt, aber meist nicht gespeichert. Außerdem kommt dieses Wissen über die Relationen überwiegend aus vorher angelegten Bibliotheken und wird nicht durch den Ansatz selbst identifiziert.

Guerra-Zubiaga et al. [95] beschreiben ein manuelles Vorgehen zur Erstellung eines Digitalen Zwillings inklusive der Anbindung an das reale Produktionssystem. Dabei wird der Aspekt der Modellrelationen im Rahmen der Modellverknüpfung betrachtet. Diese Facette ist jedoch kein komplett neuer Ansatz, sondern der Grundgedanke der modellbasierten Entwicklung [26]. Sommer et al. [126] beschreiben ein weiteres manuelles Vorgehen, das mehrere Quellen als Eingangsgröße des Ansatzes verwendet und dadurch ebenfalls Relationen zwischen den Fachbereichen abdecken kann. Darüber hinaus existiert die bereits genannte Ankerpunktmethode [119], die als einzige explizit umfangreich Relationen aus allen Fachbereichen berücksichtigt und aktualisiert. Im Vergleich zu anderen Ansätzen, die Relationen erzeugen, verwendet dieser Ansatz nicht mehrere Quellen und verbindet diese oder verwendet eine Bibliothek, welche die Relationen implizit beinhaltet, sondern extrahiert die Information aus der Namenskonvention der SPS-Signale.

3.2.2 Formate zur Abbildung der Relationen eines Digitalen Zwillings

Zusätzlich zu den Ansätzen aus der Literatur zur Erstellung und Abbildung der Relationen eines Digitalen Zwillings existieren Formate beziehungsweise Technologien, die derartige Relationen abbilden. Es handelt sich dabei um technische Lösungen, die beispielsweise bei der modellbasierten Entwicklung von Produktionssystemen, beim Austausch der Modelle und

Informationen oder bei der Nutzung der Modelle als Bestandteil eines Digitalen Zwillings eingesetzt werden.

Die modellbasierte Entwicklung ist der Ursprung vieler Modelle, die im Digitalen Zwilling eingesetzt werden. In diesem Entwicklungsprozess werden verschiedene Softwaretools zur Modellierung der Informationen verwendet. Die Ingenieure der verschiedenen Fachbereiche verwenden weitestgehend unterschiedliche Tools, die jedoch Informationen und (Teil-)Modelle austauschen müssen, sodass andere Bereiche daran anknüpfen und diese Modelle erweitern können. Dazu wurde unter anderem das Computer-Aided Engineering eXchange (CAEX)-Format, als hersteller- und toolunabhängiges Format entwickelt, das die Anlagentopologie abbildet. Das CAEX-Format wurde in der IEC 62424 standardisiert [127] und kürzlich weiterentwickelt [128]. Das CAEX-Format bildet zusätzlich zur Anlagentopologie (Eltern-Kind-Relationen) weitere relevanten Relationen für einen Digitalen Zwilling ab, wie die Klassen-Instanz-Relation (Template Instanziierung) oder Relationen zwischen Instanzen unterschiedlichen Typs (Instanz-Instanz Relationen) [85]. Das AutomationML (AML)-Format baut auf das CAEX-Format auf und erweitert dieses um fachbereichsspezifische Modelle, indem es externe Modelle in den Formaten COLLADA, PLCOpen XML und anderen XML-basierte Formate einbindet. Es ist als unabhängiges Austauschformat zur Datenübertragung zwischen heterogenen Tools gedacht. Die in dieser Arbeit betrachteten Brownfield-Produktionssysteme sind als Voraussetzung durch eine SPS automatisiert. AML hat die grundlegenden Eigenschaften des CAEX-Formats und kann durch die Erweiterung mittels PLCOpen XML auch den SPS-Code dieser Steuerungen in die Struktur des Digitalen Zwillings einbinden. Die Autoren in [19] empfehlen die Verwendung von AML für den Informationsaustausch über alle Bereiche der Entwicklung und Planung. Das AML-Format kann zwar nicht die Erstellung der angestrebten Relationen automatisieren, diese jedoch abbilden und damit als Zielformat genutzt werden. Die Verwaltungsschale (VWS) ist ein weiteres Implementierungskonzept, das explizit für Digitale Zwillinge vorgesehen ist und in der IEC 63278 [129] standardisiert wurde. Die VWS beinhaltet Teilmodelle und kapselt deren Informationen und Modelle über Schnittstellen für den einheitlichen Zugriff [130]. Die Teilmodelle sind Vorlagen, die einheitlich strukturierte Werte oder Eigenschaften beinhalten. Die VWS wird neben AML als Teil des Referenzarchitekturmodells Industrie 4.0 (RAMI 4.0) gehandelt [19].

Demgegenüber stehen native Formate geschlossener PLM-Systeme, wie beispielsweise das PLM XML-Format von Siemens, das zum Austausch zwischen den Tools im Siemens-Ökosystem genutzt wird [131]. Der Import und Export von AML-Dateien in oder aus diesen Systemen ist inzwischen niederschwellig in der Implementierung und eine zweckmäßige Erweiterung oder ist sogar bereits verfügbar [132, 133].

Insgesamt lässt sich feststellen, dass viele XML-basierte Formate und Techniken aus dem Bereich der modellbasierten Entwicklung und des Produktlebenszyklus-Managements für die Abbildung der Relationen genutzt werden können. Das AutomationML-Format ist hierbei weit verbreitet und wird in vielen Lösungen aktiv verwendet oder zumindest als zusätzliche Option unterstützt.

3.3 Zusammenfassung und Bewertung der bestehenden Ansätze

Die aufgezeigten Ansätze werden in diesem Kapitel zusammengefasst und bewertet. Die Tabelle 1 bietet einen Vergleich der bestehenden Ansätze anhand der Anforderungen in Kapitel 1.5. Rekapituliert handelt es sich um die folgenden Anforderungen:

- (A1) Automatisierte Erstellung zur Reduzierung von Fehlern und der benötigten Zeit
- (A2) Erstellung der Relationen automatisierter, mechatronischer Systeme
- (A3) Erstellung der Relationen des Digitalen Zwillings ohne (semi-)formales Expertenwissen
- (A4) Erstellung des Digitalen Zwillings ohne Beeinträchtigung des produktiven Betriebs

Die Bestimmung des jeweiligen Erfüllungsgrads (Harvey Balls) der Anforderungen erfolgt aufgrund der erworbenen Fachkenntnisse im Zusammenhang mit der durchgeführten Literaturanalyse.

Tabelle 1: Vergleich der bestehenden Ansätze in der Wissenschaft und Technik

Ansätze	(A1) autom. Ablauf	(A2) Erstellung der Relationen	(A3) Ohne (semi-) formales Wissen	(A4) parallel zum Betrieb
Buonamici et al. [99]	◐	○	●	◐
Bi und Wang [100]	◐	◐	◐	◐
Lindskog et al. [101]	◐	○	●	◐
Li et al. [102]	◐	○	●	◐
Erdős et al. [105]	●	◐	◐	◐
Stark et al. [106]	◐	◐	●	◐
Son et al. [107]	●	○	○	◐
Berglund et al. [134]	○	○	○	○
Koltun et al. [108]	●	◐	○	●
Koltun et al. [109], [111]	●	◐	○	●
Koltun et al. [110]	●	○	●	●
Biesinger et al. [113], [114]	●	◐	○	●
Zipper et al. [115]	◐	○	○	●
Bougouffa et al. [116]	●	○	●	●

Ansätze	(A1) autom. Ablauf	(A2) Erstellung der Relationen	(A3) Ohne (semi-) formales Wissen	(A4) parallel zum Betrieb
Vogel-Heuser et al. [55]	○	○	○	●
Younis und Frey [117]	●	○	●	●
Ashtari [119], [120]	●	●	○	●
Martinez et al. [121]	◐	◐	○	●
Mathias et al. [122]	◐	◐	○	●
Thongnuch et al. [123]	◐	◐	○	●
Biesinger et al. [97], [124]	●	○	●	●
Westkämper et al. [125]	◐	◐	○	●
Guerra-Zubiaga et al. [95]	○	●	○	○
Sommer et al. [126]	◐	◐	○	◐

Legende: ● erfüllt ◐ teilweise erfüllt ○ nicht erfüllt

Die Ansätze [99–102, 105, 107, 134], die auf einem optischen Verfahren basieren, weisen eine überwiegend automatisierte oder teilautomatisierte Arbeitsweise auf, da sie noch eine manuelle Segmentierung [106] oder manuelle Scans [102] benötigen. Insgesamt wird in diesen Veröffentlichungen nur wenig über den Automatisierungsgrad der Laserscans beziehungsweise der Bildaufnahmen ausgesagt. Lediglich Li et al. [102] und Berglund et al. [134] thematisieren den Aufnahmeprozess weiter und diskutieren den benötigten manuellen Aufwand für einen vollständigen Scan des Produktionssystems. Die Scans können partiell durch Aufnahmen von einer erhöhten Position automatisiert und unterbrechungsfrei durchgeführt werden. Dies erübrigt jedoch nicht zusätzliche, manuelle Scans im Produktionssystem oder unterstützt durch autonome Einheiten, um die Abschattungen zu reduzieren. Daher wurde die Erfassung im laufenden Betrieb (A4) höchstens zur Hälfte erfüllt eingestuft. Denn für eine detaillierte Rekonstruktion müssen zusätzlich Aufnahmen während des Stillstands des Produktionssystems durchgeführt werden. Die meisten Ansätze sind hinsichtlich der Anwendbarkeit ohne bestehendes Wissen in Form von Expertenwissen oder bestehenden Modellen(-bibliotheken) ausgerichtet. Dabei wird jedoch fast immer ein monolithisches Modell erstellt oder eine einfache Segmentierung durchgeführt. In Ansätzen, in denen eine umfangreiche Segmentierung und eine Zuordnung zu CAD-Features oder Modellen vorgenommen wird, muss entsprechendes Wissen vorliegen. Insgesamt lässt sich feststellen, dass die Ansätze überwiegend modellbezogen angelegt sind. So können keine

Relationen geschaffen werden, abgesehen von den Ansätzen mit einer Segmentierung, mit denen fachbereichsinterne Relationen der CAD-Features erzeugt werden können.

Die dokumentenbasierten Ansätze [108–111] arbeiten automatisiert mit Bilderkennungsverfahren und sind unabhängig vom Betrieb des Produktionssystems, da sie auf die papierförmigen oder PDF-basierten Dokumente zum Produktionssystem aufsetzen. Hinsichtlich des benötigten Wissens kann lediglich die dokumentenbasierte Erstellung von Schaltplänen ohne bestehendes Wissen genutzt werden [110]. Mit den verbleibenden dokumentenbasierten Ansätzen werden zum einen digitale Spezifikationsmodelle mit den manuellen Änderungen am Produktionssystem synchronisiert, indem aus dem Modell abgeleitete Dokumente gescannt und analysiert werden. Zum anderen kann ein aktualisierter SPS-Code aus den händisch angepassten Rohrleitungs- und Instrumentenfließschemamodellen abgeleitet werden. Diese Methoden sind daher überwiegend auf vorhandenes Wissen in Form von Modellen angewiesen und bieten daher ohne Modelle als existierendes Wissen keinen Mehrwert. In Anlehnung an die optischen Verfahren werden durch die dokumentenbasierten Ansätze ebenfalls nur Relationen erstellt, wenn entsprechende Bibliotheken oder Modelle bestehen, auf welche die Methode aufsetzt. Bei der Verwendung von Dokumenten muss zusätzlich sichergestellt werden, dass die Dokumentation nicht veraltet ist und mit dem aktuellen Zustand des Produktionssystems übereinstimmt.

Mit den netzwerkbasierten Verfahren erfolgt ein Abgleich der aktuellen Informationen aus dem Kommunikationsnetzwerk eines Produktionssystems mit Parametern aus den Simulationen oder Modellbibliotheken. Dadurch wird eine Aktualisierung der Modellierungen erreicht, die jedoch nur basierend auf diesem vorhandenen Wissen möglich ist. Die Aktualisierung kann, bei einer gut gepflegten Bibliothek sowie übereinstimmenden Benennungen im Netzwerk und dem bestehenden Wissen, überwiegend automatisiert durchgeführt werden. Basierend auf der Bibliothek ist es möglich Teile der Relationen zu erstellen, wenn diese Informationen in den Bibliothekselementen enthalten sind. Dabei ermöglicht es der Ansatz in [113] die Baumstruktur der Ressourcenliste zu erstellen. Unter der Annahme, dass ein Zugriff auf das Netzwerk vorliegt und eine Erfassung der Informationen möglich ist, ohne das Kommunikationsnetzwerk durch die große Datenmenge zu überlasten, sind diese Verfahren parallel zum aktiven Betrieb einsetzbar.

Die Ansätze von Vogel-Heuser et al. [55], Bougouffa et al. [116], Younis und Frey [117] sowie von Ashtari [119], [120] verwenden den SPS-Code als Grundlage, der zum einen immer aktuell mit dem Zustand des Produktionssystems ist, da der SPS-Code sonst nicht für dessen Steuerung verwendbar ist [119], und der zum anderen parallel zum aktuellen Betrieb untersucht werden kann. In [55] wird die Untersuchung des SPS-Codes betrachtet, um dessen Schwächen zu identifizieren, die Modularisierung zu bewerten und Verbesserungsmöglichkeiten aufzuzeigen. Dabei wird kein automatischer Ansatz vorgestellt, der Modelle oder Relationen erzeugt, sondern ein Vorgehen mit Expertengesprächen und der Nutzung eines Aufrufgraphen als Unterstützung beschrieben. Die Autoren in [116] und [117] beschreiben eine automatisierte Analyse des SPS-Codes und darauf aufbauend eine regelbasierte Erstellung eines Aufrufgraphen. Dieser Graph

wird zur Bewertung der Abhängigkeiten und daraus resultierenden Qualitätsproblemen beziehungsweise zur Erstellung von UML-Diagrammen eingesetzt, um das Code Reengineering zu unterstützen. Die Regeln sind dabei unabhängig vom Produktionssystem und beziehen sich auf die Struktur des SPS-Codes. Diese Untersuchungen und der Fakt, dass der SPS-Code eng mit der Automatisierungshardware verbunden ist [55], nutzt auch die Ankerpunktmethode [119]. Diese erstellt als einziger Ansatz Relationen zwischen mehreren Domänen, wobei hierfür eine Namenskonvention vorausgesetzt wird, die explizit Wissen über diese Relationen dem SPS-Code hinzufügt.

Die Ansätze in [97] und [124] beinhalten eine Analyse der Roboterkonfiguration beziehungsweise deren Steuersoftware. Dabei werden automatisiert die Positionen der Roboter als Layout erzeugt ohne vorhandenes (semi-)formales Wissen vorauszusetzen. Solange die Konfiguration oder Software, wie in [97] genannt, auf Backup-Server verfügbar ist, ist diese Methode auch ohne Eingriff in den Betrieb nutzbar. Dieses Vorgehen generiert, abgesehen vom Layout, kein weiteres Wissen und ist deshalb auch nicht relevant für die Nachbildung der Relationen zwischen den Modellen.

Die übrigen Ansätze [121–123, 125] zielen darauf ab, aus vorhandenen Modellen mit Hilfe von Bibliotheken weitere Modelle, vor allem Simulationsmodelle, für die virtuelle Inbetriebnahme abzuleiten. Da diese Ansätze auf Modellen als Quelle aufbauen, können sie parallel zum produktiven Betrieb verwendet werden. Bei Brownfield-Produktionssystemen ist jedoch darauf zu achten, dass diese Modelle noch aktuell sind und an Änderungen im Betrieb angepasst wurden. Zudem können diese aufgrund der verwendeten Bibliotheken nicht ohne zusätzliches Wissen genutzt werden. Der angedachte Einsatzzweck ist die Unterstützung des Entwurfsprozesses, weshalb die Ansätze lediglich teilautomatisiert sind und Ingenieure grundlegende Anteile manuell erstellen müssen. Einschränkungen bestehen auch bei der Berücksichtigung der Modellrelationen, da diese Ansätze jeweils nur die in den Bibliothekselementen hinterlegten Relationen zwischen Modellen des Ausgangsmodells und des abgeleiteten Modells erzeugen können.

Die Ansätze von Guerra-Zubiaga et al. [95] und Sommer et al. [126] charakterisieren weitestgehend manuelle Vorgehen, die Experten einbeziehen und diese unterstützen entsprechende Relationen zu erstellen. Dabei wird auch das reale Produktionssystem eingebunden, sodass der Betrieb unterbrochen werden muss.

Abgesehen von diesen Methoden gibt es die in Kapitel 3.2.2 beschriebenen Technologien zur Speicherung und zum Austausch von Informationen. Diese Technologien stammen aus der modellbasierten Entwicklung oder werden im Rahmen der Modellierung und dem Modellaustausch verwendet. Sie behandeln jedoch nicht den Generierungsprozess, der in dieser Forschung betrachtet wird. Die Technologien werden daher nicht mit in die Bewertung der existierenden Ansätze eingeschlossen. Einzelne Technologien werden im weiteren Verlauf in der Realisierung eingesetzt und für die Speicherung der Relationen genutzt.

3.4 Aufzeigen der wissenschaftlichen Forschungslücke

In den vorangehenden Kapiteln wurde der Stand der Wissenschaft und Technik aufgezeigt. Diese Untersuchung mündet in einem Überblick (Tabelle 1) der verfügbaren Ansätze in der Literatur. Es konnte festgestellt werden, dass keiner der publizierten Ansätze die vier herausgearbeiteten Anforderungen insgesamt erfüllen kann. Folglich lässt sich mit den aktuellen Technologien und Ansätzen aus der Forschung keine Lösung für das angezeigte industrielle Problem erzielen. Basierend auf der Beurteilung der existierenden Ansätze anhand der Anforderungen ergibt sich, dass bisher keine Methodik existiert, die einen Digitalen Zwilling für Brownfield-Produktionssysteme erzeugt. Dabei stehen vor allem die Relationen eines Digitalen Zwillings im Mittelpunkt, die Modelle mehrerer Fachbereiche verbinden. Diese werden in der Literatur derzeit kaum berücksichtigt, oder es werden Modelle des Produktionssystems vorausgesetzt. Schlussendlich ist festzuhalten, dass mit Bezug zum aktuellen Stand der Wissenschaft und Technik folgende Forschungslücke erkennbar ist:

Es fehlt eine Methodik, welche die fachbereichsübergreifenden Relationen eines bestehenden Produktionssystems automatisiert identifiziert und abbildet, ohne dabei den laufenden Produktionsprozess einzuschränken und ohne auf bestehendes Wissen über das Produktionssystem angewiesen zu sein.

Diese Kapitel hat gezeigt, dass die industrielle Wissenslücke und die daraus entstehende Problemstellung (vergleiche Kapitel 1.2) durch die bestehen Technologien und Forschungen bisher noch nicht geschlossen werden kann. Im folgenden Kapitel wird daher eine entwickelte Methodik vorgestellt, die in der Lage ist, die gegenwärtige Forschungslücke zu schließen.

4 Methodik zur automatisierten Erstellung der Relationen eines Digitalen Zwillings

In diesem Kapitel wird die Entwicklung einer Methodik zur Nachbildung der Relationen eines Brownfield-Produktionssystems beschrieben. Diese Methodik soll Ingenieure bei der Erstellung Digitaler Zwillinge von bestehenden Produktionssystemen unterstützen. Die Nutzung der Methodik zur Neuerstellung von Digitalen Zwillingen im Rahmen neuer Produktionssysteme wird ausdrücklich nicht betrachtet. Eingangs wird zur Methodik hingeführt und die konzeptionellen Entscheidungen erläutert. Darauf aufbauend folgt ein Überblick über die erstellte Methodik sowie eine Zusammenfassung der Bausteine der Methodik und deren Nutzen. Anschließend werden in den Kapiteln 4.3 - 4.5 die einzelnen Methoden detailliert erläutert. Die Synchronisierung dieser Methoden und die Zusammenführung des dabei erzeugten Wissens über das Produktionssystem wird in Kapitel 4.6 erklärt. Das Kapitel wird mit einer Beschreibung von der Übertragung dieses Wissens in einen Digitalen Zwilling abgeschlossen.

4.1 Hinführung und grundlegende Entscheidungen zur Erstellung der Methodik

Die anfänglich dargestellte Ausgangssituation und Problematik bilden die Grundlage der nachfolgend beschriebenen Methodik und lassen sich wie folgt zusammenfassen.

Bestehende Produktionssysteme sind das aktuelle Umfeld (Brownfield) für zukünftige (Weiter-)Entwicklungen. Um zukünftigen Trends wie der Nachfrage nach individuellen Produkten und die dafür benötigte Flexibilität der Produktionsgegebenheiten sowie deren häufige Änderungen gewachsen zu sein, müssen sie digitalisiert werden. Der Digitale Zwilling von Brownfield-Produktionssystemen bietet hierfür die Grundlage für eine Vielzahl von unterstützenden Anwendungen, die Mehrwerte erzeugen (vergleiche Kapitel 2.2.4). Bisher fehlende Digitale Zwillinge und deren Erstellung für Brownfield-Produktionssysteme stellen eine Herausforderung dar. Der Grund ist der notwendige manuelle Reengineering-Prozess, der einen hohen Zeitaufwand bedeutet, Expertenwissen verlangt, durch manuelle Vorgänge fehleranfällig ist und für den partiell ein Stillstand der Produktion notwendig ist. Der Stand der Wissenschaft und Technik umfasst bisher keine ausreichende Methodik, die als Lösung der oben zusammengefassten Problemsituation genutzt werden kann. Insbesondere die Relationen zwischen den einzelnen Modellen, die zur Erstellung eines Digitalen Zwillings benötigt werden, finden kaum Beachtung. Diese Ausgangssituation begründet die Entwicklung einer Methodik, die implizite Informationen automatisiert extrahiert und dadurch das benötigte Wissen über die Relationen des Digitalen Zwillings reduziert. Es hat sich herauskristallisiert, dass die bestehenden Ansätze nur jeweils eine Informationsquelle nutzen und daraus Modelle eines Fachbereichs erschaffen. Die wenigen Ansätze, die fachbereichsübergreifende Relationen teilweise berücksichtigen, verwenden mehr

als nur eine Informationsquelle und verbinden diese. Zur Erstellung der Relationen wird die zu entwickelnde Methodik nicht nur mit einer Informationsquelle arbeiten, sondern mehrere Quellen einbeziehen. Aus diesen wird das jeweils enthaltene relevante Wissen extrahiert und zusammengeführt. Der Digitale Zwilling beinhaltet verschiedene Relationstypen zwischen den Modellen der involvierten Fachbereiche Mechanik, Elektrik und Software (vergleiche Kapitel 2.2.3).

In der IEC-Norm 81346 werden diese Fachbereiche als Aspekte eines Objektes bezeichnet, wobei ein Objekt eine physische Existenz haben kann aber nicht haben muss [135]. Die Aspekte werden in Funktions-, Orts- und Produktaspekte aufgeteilt und umfassen die Modelle der Fachbereiche. Diese (Betrachtungs-)Aspekte, auch Views genannt, werden ebenfalls in PLM-Software für den Entwurf von Produktionssystemen verwendet. Mechanische Modelle und Relationen sind im Ortsaspekt abgebildet, Modelle und Relationen der Software spiegeln sich im Funktionsaspekt wider, nur elektrische Modelle und Relationen sind nicht direkt in einen Aspekt übersetzbar. Die Elemente der elektrischen Domäne werden in die drei Aspekte aufgeteilt, abhängig von den abzubildenden Informationen. Dies gilt auch für die realen Komponenten eines Brownfield-Produktionssystems, zum Beispiel für einen Sensor. Dieser besteht aus einer physischen Komponente (Ortsaspekt) und erfüllt mit seinem Signal eine Aufgabe (Funktionsaspekt). Zusammen mit der Stromversorgung und seinen Zuleitungskabeln bildet die Komponente den käuflich erhältlichen Sensor (Produktaspekt). Lediglich die Software besitzt keine weiteren Aspekte, abgesehen von der Funktion, da sie immateriell ist. Die Abbildung dieser Fachbereiche respektive der Aspekte soll durch die Methodik erreicht werden. Im Folgenden wird erläutert, welche Quellen verwendet und warum diese ausgewählt wurden.

Der SPS-Code hat sich in der Recherche sowie in bestehenden Arbeiten [119] als wertvolle Quelle herausgestellt, da er zwangsläufig auf dem aktuellen Stand des realen Systems ist und implizites Wissen über das Produktionssystem beinhaltet. Dies begründet sich darin, dass für die Programmierung des SPS-Codes Wissen über das System einfließt und nur eine aktuelle Version für die Steuerung des gegenwärtigen Produktionssystems geeignet ist. Da die Software einen grundlegenden und weiter steigenden Einfluss auf die Funktionen einer Anlage hat, lässt sich diese analysieren, um die Abhängigkeiten und Beziehungen des Funktionsaspekts zu extrahieren. Außerdem beinhaltet ein SPS-Projekt die Beschreibung der Automatisierungshardware und der eingesetzten Peripherie im Produktionssystem in Form von Signalen der Aktoren und Sensoren. Diese lassen sich analysieren und aufbereiten, um Aussagen über die Relationen der elektrischen Domäne zu treffen. Aus diesen Gründen wird der SPS-Code als Quelle in dieser Forschung herangezogen. Der SPS-Code kann dabei in verschiedenen, genormten Programmiersprachen vorliegen, die sich in textbasierte (Anweisungsliste AWL, Strukturierter Text SCL) und graphische Programmiersprachen (Funktionsblockdiagramm FBD, Kontaktplan LAD, Ablaufsprache SFC) gliedern lassen. Diese beiden Gruppen unterscheiden sich grundlegend in ihrem Aufbau und sind bei der Verwendung als Quelle verschieden zu behandeln. Die

textbasierten Programmiersprachen benötigen eine aufwendigere Vorverarbeitung, da die Semantik umfangreicher und komplexer ist. Die graphischen Programmiersprachen besitzen eine eingeschränktere Semantik und sind daher leichter zu interpretieren. Darüber hinaus haben in den letzten Jahrzehnten diverse Hersteller zahlreiche unterschiedliche SPS-Modelle entwickelt, die ebenso vielfältig als Automatisierungssystem in Brownfield-Produktionssystemen eingesetzt werden. Daher muss die zu entwickelnde Methodik die Vielfalt herstellerspezifischer Codeformate adressieren, um die Methodik entkoppelt vom Hersteller und dem Modell der SPS einsetzbar zu machen. Dies ist ein entscheidender Faktor, um die Methodik auf eine Vielzahl der gegenwärtigen Brownfield-Produktionssysteme anwenden zu können. Im Vergleich zu bestehenden Methoden der SPS-Code-Analyse soll die zu entwickelnde Methodik keine Vorbedingungen an die Brownfield-Produktionssysteme und deren SPS-Code stellen, wie zum Beispiel die Verwendung einer Namenskonvention. Eine solche Namenskonvention mit zusätzlichen Informationen, wie die Ankerpunktmethodik [119] voraussetzt, wurde bei der bereits erfolgten Realisierung von Brownfield-Produktionssystemen nicht berücksichtigt. Folglich sind solche Methoden mit Vorbedingung nicht ohne weiteres für Brownfield anwendbar. Eine Analyse des SPS-Codes muss daher ohne diese und ähnliche spezielle Vorbedingungen funktionieren.

Darüber hinaus gilt es, die mechanische Domäne beziehungsweise den Ortsaspekt abzubilden. Andere Methoden codieren und hinterlegen diese Informationen explizit in anderen Quellen, wie dem SPS-Code oder den Inbetriebnahme-Dokumenten. Diese sind jedoch herstellerübergreifend nicht einheitlich und nicht immer vorhanden. Daher müssen diese Informationen aus einer anderen Quelle extrahiert und ergänzt werden. Die Erfassung des Ortsaspekts, bestehend aus der physischen Gruppierung und Relationen der Peripherie des Produktionssystems, kann durch verschiedene bestehende Ansätze durchgeführt werden. Laserscans sind dabei eine der präzisesten Verfahren zur automatisierten Erstellung des CAD-Modells. Die entstehende monolithische Punktwolke kann mittels Segmentierungsverfahren in Subkomponenten und Teilmodelle zerlegt werden. Diese Methode erfordert jedoch kostspielige Erfassungssysteme und einen physischen Zugang zum System, um Nahaufnahmen der Peripherie und verschatteten Objekten zu machen. Des Weiteren können die einzelnen Komponenten nicht eindeutig identifiziert werden, wenn mehrfach dasselbe Bauteil verbaut wurde. Dadurch ist eine Zuordnung zur funktionalen Struktur nicht möglich. Außerdem ist es für die Erstellung der Relationen des Digitalen Zwillings nicht notwendig, dass ein vollständiges CAD-Modell mit einer hohen Genauigkeit existiert. Vielmehr werden die groben physischen Abmessungen und Gruppierungen benötigt. Daher werden in diesem Methodenbaustein Lokalisierungssysteme für Innenräume zur Erfassung der physischen Eigenschaften betrachtet. Im Vergleich zu Laserscans sind sie günstiger und haben eine ausreichende Genauigkeit zur Abbildung der physischen Abmessungen und Gruppen. Ein weiterer Grund für die Verwendung dieser Lokalisierungssysteme ist, dass sie in vielen industriellen Produktionssystemen vorhanden sind. Dadurch kann auf bestehende Lokalisierungssysteme aufgebaut werden und es muss nicht extra ein komplett neues Erfassungssystem verbaut werden. Existierende Lokalisierungssysteme sind heute häufig in der

Intralogistik von Großunternehmen zu finden, wo bisher Systeme auf RFID- (Radio Frequency Identification) oder Bluetooth Low Energy-Basis am weitesten verbreitet sind [136]. Im Automotive-Sektor sind schon die präziseren Ultra-Wideband (UWB) Systeme im Einsatz [137], die durch den neuen 5G Standard ergänzt werden können [138]. Auch der perspektivisch steigende Einsatz von autonomen mobilen Robotern (AMR) und die damit verbundene notwendige Positionserfassung führt zukünftig zu einer steigenden Verfügbarkeit von Positionsdaten. Aus diesen Gründen bietet sich das Lokalisierungssystem als Datenquelle an, aus der Informationen über die mechanische Form und Segmente des Produktionssystems extrahiert werden können. Zur Nutzung der Positionsdaten werden datenbasierte Verfahren benötigt, welche Muster aus den erfassten Positionen extrahieren können. Die Intention der Analyse ist es, Berechnungen über die ungefähren Positionen der Peripherie zu ermöglichen. Dazu werden zusätzlich zu den Positionsdaten auch Betriebsdaten erhoben und eine gemeinsame, datengetriebene Analyse durchgeführt. Die Auswirkung der Materialbewegung auf die aktuellen Werte der Betriebsdaten ermöglicht es durch eine gemeinsame Korrelationsanalyse der beiden Quellen Rückschlüsse auf die ungefähren Positionen der Peripherie zu ziehen.

Bei den zu extrahierenden Informationen aus dem SPS-Code sowie den Betriebs- und Positionsdaten handelt es sich um Informationen über die einzelnen Aspekte eines Objekts. Dabei kann es sich um Informationen über das Objekt selbst handeln oder um Informationen zur Beziehung eines Objektes zu anderen. Daraus resultiert eine Struktur bestehend aus Objekten sowie den umfangreichen Relationen innerhalb deren Aspekten und zu anderen Objekten. Die Ergebnisse der Analysen werden in einem Graphen anhand der extrahierten, identischen Objekte zusammengeführt. Die identischen Objekte können anhand der Signalnamen der Peripherie im SPS-Code und deren Verwendung als Name in den Betriebsdaten identifiziert werden. Die Graph-Struktur ermöglicht, nach fehlenden, untypischen oder sich wiederholenden Elementen und Anordnungen zu suchen. Dadurch kann die Darstellung des Wissensgraph komprimierter und übersichtlicher gestaltet sowie zusätzliche Erkenntnisse über dessen Aufbau gewonnen werden.

Zusammengefasst sind die zugrundeliegenden Datenquellen dieser Forschung die SPS und das jeweils vorhandene Positionserfassungssystem. Im Rahmen dieser Forschung werden der SPS-Code sowie die gespeicherten, historisierten Positions- und Betriebsdaten als Informationsquellen bezeichnet, da für die Analysen innerhalb der Methodik die bereits erfassten und semantisch beschriebenen Datenbestände verwendet werden. Es handelt sich also nicht mehr um Daten der niedrigsten Stufe der DIKW-Pyramide nach [139], sondern bereits um Informationen. Daher wird im Weiteren von Informationsquellen gesprochen und die Analyse von der Datenerfassung entkoppelt betrachtet. Die Erfassung der Daten und deren kontextuelle beziehungsweise semantische Anreicherung zur Erzeugung von Informationen werden im ersten Abschnitt zur Erfassung und Import jedes Methodenbausteins thematisiert (Kapitel 4.3.1, 4.4.1 und 4.5.1).

Die Neuerstellung von Produktionssystemen mit Digitalem Zwilling wird von dieser Methodik nicht umfasst. Der Verwendungszweck liegt auf der Erstellung der Relationen für einen Digitalen

Zwilling eines Brownfield-Produktionssystems. Bestehende Produktionssysteme sind, wie in Kapitel 1 ausgewiesen, weit verbreitet und haben ein nicht ausgeschöpftes Potenzial, das durch die Anwendung eines Digitalen Zwillings gehoben werden kann. Des Weiteren werden von der Methodik die Safety-Funktionen und Komponenten aus zwei Gründen vernachlässigt: Zum einen unterliegen Safety-Komponenten strikteren Anforderungen hinsichtlich ihrer Nachbildung als digitale Replikation, um die Einhaltung der Safety-Normen und Richtlinien gewährleisten zu können. Zum anderen wird in Industrieunternehmen aufgrund dieser Vorschriften eine automatisierte Nachbildung seltener akzeptiert oder nur mit umfassender manueller Prüfung. Daher werden in der weiteren Forschung die Safety Aspekte ausgeklammert und auf den umfangreichen Teil der funktionalen Komponenten verstärkt eingegangen.

4.2 Überblick über die Methodenbausteine

In diesem Kapitel wird das Konzept der Methodik entwickelt. In diesem Kontext werden die Bausteine der Methodik vorgestellt. Das Konzept baut auf der Hinführung aus dem vorangehenden Kapitel auf und lässt sich wie in Abbildung 4.1 darstellen. Es ist in vier Bereiche mit jeweils mehreren Bausteinen untergliedert, wobei die Bausteine eines Bereichs jeweils ähnliche Aufgaben erfüllen. Die vier Konzeptbereiche sind in Abbildung 4.1 horizontal angeordnet und werden von links nach rechts durchlaufen. Dabei können manche Pfade im linken Bereich unabhängig voneinander parallel durchgeführt werden. Die blau hinterlegten Bereiche sind der Kern der entwickelten Methodik, der graue Bereich bilden deren Grundlage.

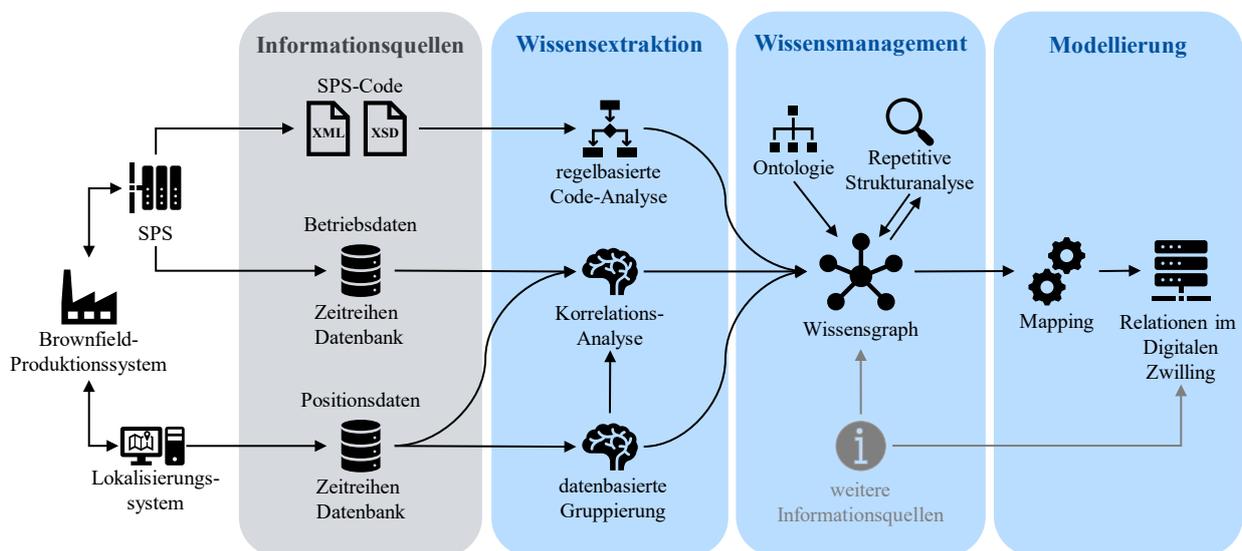


Abbildung 4.1: Das Konzept – Überblick über die Methodik (blauhinterlegte Bereiche)

Der erste Konzeptbereich beinhaltet die Informationsquellen, aus denen das relevante Wissen über das bestehende Produktionssystem extrahiert und verknüpft werden soll. Der Stand der Technik und Wissenschaft zeigt bereits, dass eine Quelle allein zur Nachbildung der Relationen nicht ausreichend ist. Daher werden bei diesem Ansatz drei Quellen verwendet, um die initialen

Anforderungen (siehe Kapitel 1.5) zu erfüllen. Konkret wird der SPS-Code als Quelle verwendet, um die funktionalen Relationen abzuleiten. Die Positionsdaten werden genutzt, um daraus die Gruppierung der Anlagenteile zu extrahieren. Außerdem werden sie zusammen mit den Betriebsdaten verwendet, um die Sensoren und Aktoren physisch zu lokalisieren und den Gruppen zuzuordnen. Diese drei Informationsquellen unterscheiden sich hinsichtlich ihres Informationsgehaltes, ihrer Struktur und Art.

Daher sind im zweiten Konzeptbereich („Wissensextraktion“) unterschiedliche, separate Analysemethoden für die drei Quellen vorgesehen. Die Wissensextraktion hat zum Ziel, aus den Informationsquellen Wissen über das Brownfield-Produktionssystem, speziell zu den Relationen, zu extrahieren. Für die Untersuchung des SPS-Codes ist eine regelbasierte Analyse zur Bestimmung der funktionalen Relationen und zur Quantifizierung der Relationen entwickelt worden. Zur Analyse der Zeitreihendaten werden datengetriebene Methoden aus dem Bereich des Data Mining verwendet. Es kommen dabei sowohl Algorithmen aus dem Bereich des maschinellen Lernens als auch statistische Verfahren zur Extraktion der enthaltenen Muster in den Datenbeständen zum Einsatz. Dadurch werden die physischen Relationen bestimmt. Die Analyse der Positions- und Betriebsdaten findet außerdem im Vergleich zum SPS-Code nicht komplett isoliert voneinander statt. Die Positionsdaten werden einerseits für eine separate Analyse verwendet, andererseits auch in Kombination mit den Betriebsdaten, um die enthaltenen Informationen zu fusionieren. Die datenbasierten Analysemethoden werden als Analysepipeline verknüpft, sodass einzelne Analyseschritte leicht ausgetauscht und an anlagenspezifische Gegebenheiten angepasst werden können.

Der dritte Konzeptbereich („Wissensmanagement“) beinhaltet die Zusammenführung des generierten Wissens aus den drei Analysemethoden der Wissensextraktion in einer verknüpften Wissensbasis und deren Anreicherung mit zusätzlichem Wissen. Das generierte Wissen umfasst die Komponenten des Produktionssystems, deren Beziehungen zueinander und Gruppierung nach ihrer Funktion, ihrer physischen Position und Zusammengehörigkeit. Diese Ergebnisse sind die Eingangsgröße dieses Konzeptbereichs und sind überwiegend durch Beziehungen charakterisiert. Für die Speicherung von Datenpunkten mit vielen Beziehungen haben sich Graphen zur Speicherung etabliert. Graphen zur Speicherung von Wissen werden auch Wissensgraphen genannt. Diese bauen auf spezifische Ontologien zur formalen Beschreibung des Inhaltes und zur maschinellen Verarbeitbarkeit auf. Die Ontologien bilden den Ausgangspunkt für die automatische Anreicherung und Vervollständigung des erzeugten Wissensgraphen. Die maschinelle Lesbarkeit ist für den letzten Konzeptbereich, den Transfer in einen Digitalen Zwilling, eine essenzielle Grundlage. Die Zusammenführung, Verknüpfung und Anreicherung der verschiedenen Analyseergebnisse ergänzen die fachbereichsübergreifenden Relationen.

Dieser letzte Konzeptbereich überführt das Wissen aus dem Wissensgraph in ein Softwaretool oder Datenformat zur Modellierung Digitaler Zwillinge. Dieser Schritt liefert kein weiteres Wissen über das abzubildende Produktionssystem und dient der Konvertierung der Daten. Er ist

für die Evaluierung der Methodik als implementiertes Software-Assistenzsystem relevant, um die Daten in ein exemplarisches Zielsystem zu übertragen.

Das Konzeptbild beinhaltet außerdem externe Informationsquellen, die nach Bedarf in die Wissensbasis eingespeist oder im Zielsystem hinzugefügt werden können. Sie sind nicht Teil der entwickelten Methodik werden hier jedoch der Vollständigkeit halber erwähnt. Sie werden für die Erstellung eines vollständigen Digitalen Zwillings benötigt, da die vorliegende Methodik keine Modelle erzeugt, sondern die Relationen des Digitalen Zwillings. Deshalb müssen diese Quellen anwendungsfallspezifisch ergänzt werden, um die benötigten Modelle einzufügen. Diese Quelle kann auch ein Experte sein und die Ergänzung von diesem manuell durchgeführt werden. Ebenso können die externe Quellen Ansätze wie Laserscans oder Analysen von Rotstiftänderungen an Schaltplänen sein.

In den folgenden drei Unterkapiteln (4.3 - 4.5) wird genauer auf die einzelnen Analysemethoden der Wissensextraktion eingegangen. Zuerst wird jeweils eine Informationsquelle und die korrespondierende Analysemethode zur Wissensextraktion detailliert beschrieben. Anschließend werden die nachfolgenden Konzeptbereiche sequenziell von links nach rechts erörtert, beginnend mit dem Wissensmanagement zur Synchronisierung der Methoden.

4.3 Regelbasierte Analysemethode der SPS-Codestruktur zur Extraktion funktionaler Relationen

Die regelbasierte Analysemethode der SPS-Codestruktur ist der erste Methodenbaustein der Wissensextraktion. Er analysiert die Struktur des SPS-Codes zur Extraktion der Peripherie des Produktionssystems und deren funktionale Relationen [140].

Die Auswahl des SPS-Codes als Informationsquelle wurde in Kapitel 4.1 begründet und wird hier noch um fachliche Details ergänzt. Zum einen sind SPS für bestehende Produktionssysteme die am häufigsten eingesetzte Steuereinheit, wie bereits in Kapitel 2.1.2 erläutert. Daraus resultiert, dass der SPS-Code für die überwiegende Mehrzahl der Brownfield-Produktionssysteme als Quelle verfügbar ist und dadurch eine breite Anwendbarkeit der Methodik ermöglicht wird. Zum anderen ist der SPS-Code als Informationsquelle an die vorangehenden Rekonfigurationen im Lebenszyklus der Anlage angepasst, da ansonsten neue Sensoren oder Aktoren nicht verwendet, alte entfernte Komponenten weiter ausgelesen und Verschiebungen sowie Prozessanpassungen falsch angesteuert werden. Der SPS-Code muss im Zuge solcher Anpassungen entsprechend modifiziert und aktualisiert werden. Daher kann der SPS-Code als kontinuierlich aktuelle Informationsquelle über das Brownfield-Produktionssystem betrachtet werden.

Ein weiterer Vorteil des SPS-Codes als Quelle sind die verfügbaren Standards zur SPS [141–143] sowie zur Zusammenarbeit mit anderen softwarebasierten Steuerungsbausteinen [144]. Diese Standards vereinheitlichen die Sprachen, Strukturen sowie die Zusammenarbeit von SPS-

Programmen und reduzieren dadurch die möglichen Variationen beziehungsweise vereinfachen infolgedessen die Analysemechanismen. Ein Teil des IEC 61131 Standards (IEC 61131-10) [145] behandelt außerdem ein XML-basiertes Austauschformat nach der Vorlage des PLCopen XML Formats. Dieses Austauschformat dient der herstellerunabhängigen Speicherung und Weitergabe von SPS-Programmen, beispielsweise, um auf eine neue Hardware portieren zu können oder um zu einer anderen SPS-Entwicklungsumgebung zu wechseln. Dieses Austauschformat bietet den Grundbaustein für eine herstellerübergreifende Analyse des SPS-Codes und seiner Verwendung als Informationsquelle. Die Ankerpunktmethod, die als einziger Ansatz aus dem Stand der Technik und Forschung domänenübergreifend Relationen bildet, nutzt auch das XML-Austauschformat und den weitverbreiteten Standard IEC 81346 zur Betriebsmittelkennzeichnung [135, 146]. Sie untersucht ebenfalls den SPS-Code, um daraus Informationen über das Produktionssystem abzuleiten. Jedoch wird für die Ankerpunktmethod die Verwendung einer Namenskonvention für Signalnamen in der SPS als spezifische Anwendung der Betriebsmittelkennzeichnung vorausgesetzt. Diese Benennung wird methodisch analysiert und die Informationen extrahiert. Die Norm zur Betriebsmittelkennzeichnung ist zwar weit verbreitet, aber ihre Auslegung wird individuell nach den Spezifikationen des Anlagenbetreibers adaptiert (z.B. Integra Standard von Daimler). Daher kann ein Digitaler Zwilling nicht für alle Produktionssysteme nach dieser Method erstellt werden.

Der SPS-Code als Software des Automatisierungssystems beinhaltet in erster Linie die Anweisungen zur Steuerung von Aktoren basieren auf den aktuellen Sensorwerten. Jenseits von diesen expliziten Steueranweisungen soll die Strukturanalyse die impliziten Informationen über das Produktionssystem extrahieren, das dem Automatisierungssystem untergeordnet ist. Die Strukturanalyse untersucht dazu die Abhängigkeiten und Beziehungen zwischen Funktionsbausteinen und der verwendeten Peripherie. Dadurch werden Rückschlüsse zur Funktion der Anlage, die Zuordnung der Peripherie zu separaten Funktionen und Abhängigkeiten zwischen der Peripherie ermöglicht. Die vermehrte Realisierung von Funktionen eines Produktionssystems durch Software wird den Informationsgehalt der Steuerungssoftware über den funktionalen Aspekt weiter erhöhen und dadurch detailliertere Aussagen zulassen. Die Analyse der Struktur des SPS-Codes extrahiert diese impliziten Informationen und stellt somit die Relationen zwischen den Software-Komponenten, die funktionale Abgrenzung dieser Komponenten sowie deren Zusammenhänge zur Peripherie her.

Das nächste Unterkapitel behandelt den Import des SPS-Codes verschiedener Formate in ein einheitliches Informationsmodell, welches als Basis für eine übergreifende Analyse dient. Die Analyse selbst, deren Schritte, Regeln und Ziele werden im Anschluss betrachtet. Danach werden die Ergebnisse und der Wissensgewinn vorgestellt und der Methodenbaustein zusammengefasst.

4.3.1 Import des nativen SPS-Codes als Informationsquelle in ein herstellerunabhängiges Informationsmodell

Der Import des SPS-Codes zielt darauf ab, die Informationen im SPS-Code in ein einheitliches Informationsmodell (siehe Abbildung 4.3) zu bringen, auf das die Analyse generisch aufgesetzt werden kann. Dabei erweisen sich die existierenden verschiedenen Hersteller von SPS und deren native Entwicklungsumgebungen als eine Herausforderung. Denn diese verwenden native Formate zur Speicherung des SPS-Codes. Dadurch können nicht alle SPS-Code-Dateien in einer gemeinsamen Importfunktion verarbeitet, sondern müssen in separaten Funktionen behandelt werden. Darüber hinaus sind sogar die SPS-Formate eines Herstellers über die verschiedenen Versionen der Entwicklungsumgebung unterschiedlich strukturiert und codiert. Es ist daher keine Option mehrere Importfunktionen zu entwickeln, um die Mehrzahl der existierenden SPS-Dateien verarbeiten zu können. Der XML-Standard 61131-10 beziehungsweise der zugrunde liegende PLCopen XML-Standard ermöglichen einen vereinheitlichten Import des SPS-Codes durch die definierte Datenstruktur. Nahezu alle SPS-Hersteller bieten inzwischen für die Speicherung des SPS-Codes, abgesehen von deren nativen Formaten, einen Export in diesem XML-Format oder in einem daran stark angelehnten Format an. Diese Exportfunktionen können manuell über die Benutzeroberfläche der Entwicklungsumgebung genutzt oder über eine entsprechende Schnittstelle automatisiert durch andere Programme aufgerufen werden. Beispielsweise bietet Siemens für seine Entwicklungsumgebung TIA Portal die Schnittstelle TIA Openness an, Beckhoff bietet die ITcPlcIECProject-Schnittstelle für seine Entwicklungsumgebung TwinCAT 3 XAE an. Dabei erzeugt das ITcPlcIECProject XML-Dateien nach dem PLCopen XML-Standard, wohingegen TIA Openness lediglich Dateien in Anlehnung an die IEC 61131-10 erzeugt. Die Exportfunktionen der Hersteller erstellen diese XML-Dateien nicht alle zwangsläufig genau nach den Standards und auch der Standard selbst hat Freiheitsgrade hinsichtlich der exakten Benennung der Elemente. Deshalb werden Mechanismen zur Interpretation der Dateien im Import benötigt. Üblicherweise werden hierfür XML Schema Definitionen (XSD) eingesetzt, die eine semantische Beschreibung aller möglichen Elemente und deren strukturelle Anordnung definieren. Im Sinne der objektorientierten Programmierung entspricht eine XSD dem Prinzip einer Klasse und deren Instanz (Objekt) ist eine XML-Datei. Basierend auf diesen XSD-Dateien können die XML-Dateien interpretiert, in ein einheitliches Informationsmodell überführt und anschließend analysiert werden. Grundlage für das Importmodul ist daher die Verfügbarkeit der XSD-Dateien. Um diese zu erhalten, existieren mehrere Möglichkeiten:

Die Verwendung einer *XSD nach der Norm 61131-10* ist die erste Option und ein generisches Vorgehen für alle XML-Dateien, die streng nach dem Standard aufgebaut sind. Die Verwendung einer normbasierten XSD hat den Vorteil, dass sie ohne Modifikation für viele XML-Dateien herangezogen werden kann. Im besten Fall ist diese XSD universell einsetzbar, wenn sich die Norm komplett durchsetzt. Für alle XML-Dateien, die nur in Anlehnung an die Norm erstellt wurden, kann eine normbasierte XSD zumindest die Grundelemente beschreiben.

Die Ableitung einer *XSD aus einer oder mehreren XML-Dateien* bildet die zweite Option und erzeugt eine Sammlung aller strukturellen Elemente aus den XML-Dateien. Daher kann eine abgeleitete XSD nie vollständig alle existierenden Elemente abbilden und muss für neue XML-Dateien eventuell erweitert werden, wenn diese neue Strukturelemente implementieren. Die Erweiterung der XSD kann entweder durch die Generierung einer neuen XSD mit den neuen XML-Dateien und anschließender Zusammenführung mit der bestehenden Strukturdefinition erreicht werden oder durch die Erstellung einer neuen XSD basierend auf allen alten und neuen XML-Dateien. Dies führt in beiden Fällen zu identischen Ergebnissen und bildet die Summe aller involvierter XML-Dateien ab. Die Erzeugung einer XSD aus XML-Dateien kann aufwandsarm mit Softwaretools vollautomatisiert durchgeführt werden. Daher ist sowohl die (Neu-)Erstellung als auch die Erweiterung unproblematisch. Alle Elemente der XML-Dateien, die in die Generierung der XSD einbezogen wurden, sind vollständig durch die so erzeugte Definitionsdatei abgebildet. Diese Version der XSD-Erstellung ist für XML-Dateien, die nur an der Norm angelehnt oder nicht normgerecht sind, zu bevorzugen.

Die Nutzung der *herstellerspezifischen XSD-Dateien* ist die dritte Option. Diese basiert auf dem Fakt, dass die Hersteller der XML-Generatoren für den Export des SPS-Codes als XML-Dateien bereits XSD-Dateien benötigen. Basierend auf diesen Definitionsdateien werden die Elemente des SPS-Codes in XML-Elemente übersetzt und abgespeichert. Daher verfügen die Hersteller der Exportfunktion bereits über die benötigten XSD, die für alle XML-Dateien verwendet werden können, die mit dieser Funktion exportiert wurden. Im Vergleich zur XSD-Generierung aus einer begrenzten Anzahl von XML-Dateien, sind diese nicht nur kompatibel zu diesen XML-Dateien und ähnlichen, die dieselben Elemente beinhalten, sondern zu allen durch eine Exportfunktion entstandenen XML-Dateien. Der Zugang zu diesen XSD-Dateien der Hersteller ist im Rahmen dieser Arbeit beschränkt. Die XSD-Dateien sind in den allermeisten Fällen nicht frei verfügbar oder werden überhaupt nicht herausgegeben. Eine kommerzielle Umsetzung mit entsprechenden Finanzmitteln und überzeugender Marktmacht kann diese Option voraussichtlich nutzen und die direkte Anwendbarkeit der Methode noch weiter erhöhen.

Der Aufwand für die Anpassung der XSD und der darauf aufbauende Import sind jedoch überschaubar. Es wurde deshalb darauf verzichtet, eine vollständige Implementierung zu erzielen. Im Folgenden wird die Option der XSD-Generierung aus XML-Dateien weiter betrachtet und der Import darauf ausgerichtet.

Der Import der XML-Dateien in ein Modell beispielsweise in Java oder Python, basierend auf den XSD-Dateien, läuft nach den Schritten in Abbildung 4.2 ab. Dabei können, wie in dieser Abbildung angedeutet, die herstellerspezifische XSD für den Einlesevorgang übernommen oder wie vorher beschrieben, generierte XSD beziehungsweise normbasierte XSD-Dateien verwendet werden. Vor dem eigentlichen Einlesevorgang werden für alle Elemente und Attribute der Strukturdefinition eine entsprechende Klasse oder Parameter angelegt. Diese Klassen und ihre Parameter bilden das korrespondierende Metamodell der XSD. Beim Import der XML-Dateien

wird nun anhand der XSD entschieden, wie die XML-Dateien in das Metamodell übertragen und dieses instanziiert werden muss.

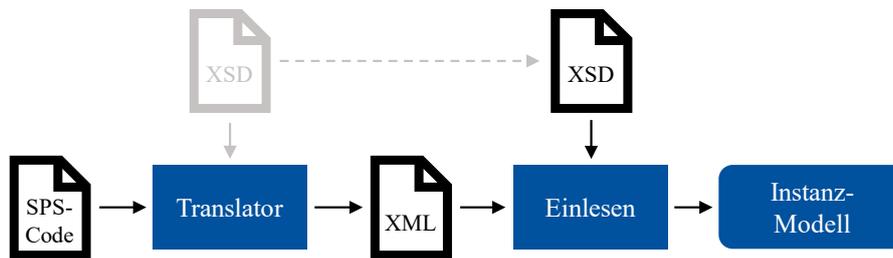


Abbildung 4.2: Importschritte

Diese Instanz des Metamodells kann nun durch entsprechende Analysen verarbeitet und Wissen über die funktionale Struktur extrahiert werden. Diese Schritte werden im nächsten Kapitel erläutert.

4.3.2 Anreicherungs- und Analyseregeln der Analysemethode

Das instanziierte Metamodell des Imports dient als Ausgangsposition für die weiteren Schritte. Prinzipiell könnte es direkt analysiert und verarbeitet werden, jedoch sind in diesem Metamodell noch Informationen zum herstellerspezifischen Exportmodul und Metainformationen zu XML-Strukturelementen enthalten. Die Folge davon ist, dass das Metamodell nicht komplett identisch ist zwischen allen Herstellern. Des Weiteren sind einige Verbindungen zwischen den Klassen des Metamodells, das aus XSD erstellt wurde, nur unidirektional verknüpft. Dies resultiert daraus, dass ein Element in der XML-Struktur ein direktes Kind eines anderen Elements ist und somit im Metamodell nur die Klasse des Elternknotens die Liste aller Kinder besitzt oder da hierarchisch gleichwertige Elemente nicht mit redundanten Informationen zu deren Beziehung untereinander versehen werden. Die Analyse dieser Verbindungen in beide Richtungen kann abhängig von der Anzahl der Klasseninstanzen zeitaufwendig sein. Insbesondere wenn viele Objekte nach dem gesuchten Gegenstück durchsucht werden müssen. Aufgrund dieser Faktoren wird das Metamodell in ein Informationsmodell überführt, das die Struktur des SPS-Codes abbildet und wichtige Informationen für die Analyse mehrfach beinhaltet. Dabei wird das Prinzip der nicht redundanten Informationsspeicherung bewusst verletzt, um die Analyse zu vereinfachen und zu beschleunigen. Darüber hinaus ermöglicht das einheitliche, bereinigte und angereicherte Informationsmodell eine gleiche Anwendbarkeit der Analyse im Gegensatz zum Metamodell, das wegen den kleinen Variationen der XSD nicht immer exakt gleich sein muss. Das erstellte Informationsmodell ist in Abbildung 4.3 dargestellt und bildet die Elemente der standardisierten SPS-Sprachen und der referenzierten Hardware im SPS-Code ab. Es wurden die strukturbasierten Sprachen (FBD, LAD, SFC) untersucht und die textbasierten Programmiersprachen (AWL und SCL) außen vor gelassen.

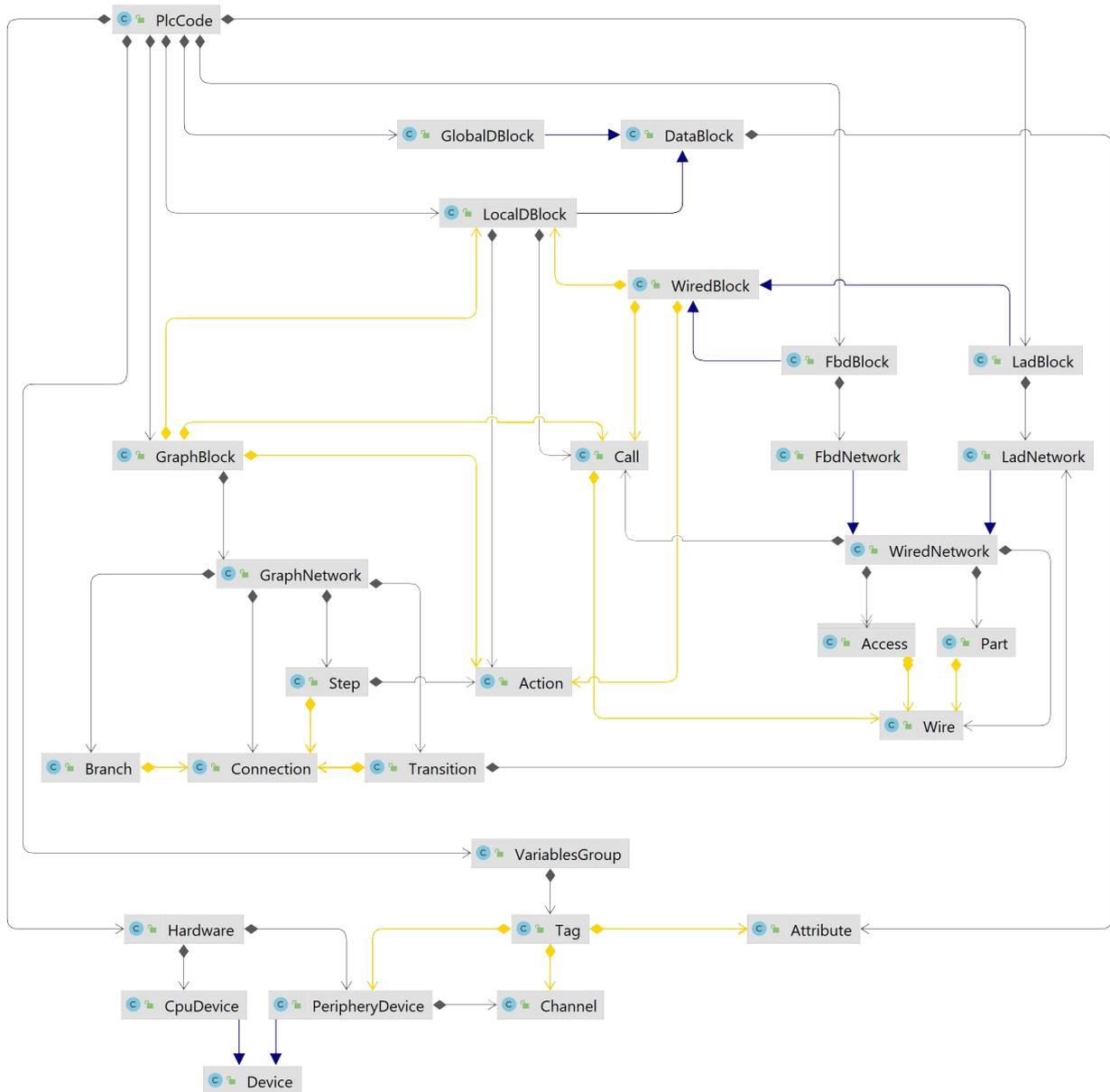


Abbildung 4.3: Klassendiagramm des herstellerunabhängigen Informationsmodells des SPS-Codes

Die textbasierten Programmiersprachen wurden aus aufwandstechnischen Gründen nicht in die Analyse und Modellierung eingeschlossen, da diese ein umfangreicheres Sprachmodell zur Interpretation der Informationen benötigen und damit den zeitlichen Rahmen der Forschung überschreiten würden. Prinzipiell ist die Analyse auch auf die textbasierten Sprachen übertragbar, wenn diese in das Informationsmodell integriert sind. Dies wird unter dem Gesichtspunkt der Erweiterung der Methodik auf hochsprachenbasierte Steuerungen im Ausblick dieser Arbeit angerissen.

Das übergeordnete Element des herstellerunabhängigen Informationsmodells (siehe Abbildung 4.3) ist die Klasse `PlcCode`, die dem SPS-Projekt entspricht. Diese Klasse umfasst alle Funktionsbausteine, die sukzessive die Substrukturelemente und Hardware beinhalten. Die

Struktur des Informationsmodells leitet sich direkt aus der baumartigen Struktur des Metamodells und der zugrundeliegenden XSD ab. Daher sind die Subelemente, wie bei einer Baumstruktur üblich, nicht untereinander verknüpft. Ein Beispiel ist die Klasse `Connection`, die Verbindungen von Transitionen und Schritten in der Ablaufsprache repräsentiert. Diese Verbindungen bestimmen die Sequenzreihenfolge im Ablaufgraphen und enthalten dafür Attribute, welche die zugehörigen Schritte und Transitionen beschreiben. Diese Attribute werden bei der Instanziierung analysiert und das referenzierte Element im restlichen Baum gesucht. Über die Bedeutung der Elemente und deren Position im Baum können eindeutige IDs als Referenzen auf das Gegenstück gefunden werden. Die ID des jeweils verbundenen Elements wird den beiden Elementen einer Beziehung hinzugefügt. Es werden überwiegend 1:1 Beziehungen referenziert, wodurch den Klassen eine zusätzliche Variable hinzugefügt werden muss. Die wenigen 1:n Beziehungen erhalten auf der mehrdeutigen Seite eine unsortierte Liste aller IDs. Die Verbindung wird dabei, wie bereits erläutert, aus Gründen der Analyseperformance redundant in beiden Elementen gespeichert. Ebenso werden die bereits vorhandenen Attribute zur Darstellung der Verbindung bestehen gelassen. In Gelb sind die ergänzten Assoziationen zwischen den Klassen des Informationsmodells eingezeichnet, wobei der Pfeil auf das Element zeigt, das bereits die Verbindungsbeschreibung innehatte. Das Informationsmodell ist nach dieser Ergänzung nicht mehr baumartig aufgebaut, sondern bildet eine Netzwerkstruktur.

Die Analyse der Struktur des SPS-Codes über den Zwischenschritt des Informationsmodelles hat das Ziel, die funktional zusammenhängenden Aktoren und Sensoren sowie die zugehörigen Softwareelemente (Bausteine) zu identifizieren und zu kennzeichnen. Diese funktionalen Gruppen lassen sich über die gemeinsame Nutzung der Peripherie innerhalb einzelner Bausteine detektieren. Einzelne Bausteine bilden hierbei die kleinste funktionale Gruppe. Jedoch sind nicht alle Bausteine einer SPS voneinander funktional unabhängig, sondern erfüllen größtenteils zusammen eine Funktion. Daher wird regelbasiert die Struktur des SPS-Codes und der verwendeten Peripherie-Signale analysiert, um die abhängigen Bausteine zu identifizieren. Dabei werden zwei Arten von Abhängigkeitspfaden durch die Analysemethode untersucht. Diese zwei Arten der Abhängigkeit sind in Abbildung 4.4 zwischen den zwei Bausteine (FB1 und FB2) eines generalisierten SPS-Codes und zwei Peripheriegeräte (Hardware) visualisiert. Als erste Möglichkeit kann die Abhängigkeit zweier Bausteine direkt durch deren gegenseitigen Aufruf identifiziert werden. Dies ist vergleichbar mit einem Funktions- oder Methodenaufruf in Hochsprachen und ist eine direkte gegenseitige Verwendung. Bausteine im SPS-Code beinhalten sogenannte Netzwerke, die zyklisch nacheinander ausgeführt werden. Dabei greifen sie auf die Sensoren und Aktoren (Tags) zu oder rufen andere Bausteine auf. Diese Abhängigkeit der Bausteine FB1 über dessen Netzwerk 2 zum Baustein FB2 (vergleiche Abbildung 4.4) wird als Own-Call-Beziehung bezeichnet und bildet die erste Art eines funktionalen Abhängigkeitspfades. Dieser Pfad ist einfach zu identifizieren, indem alle Netzwerke eines SPS-Codes analysiert und die Aufrufe anderer Bausteine gesammelt werden.

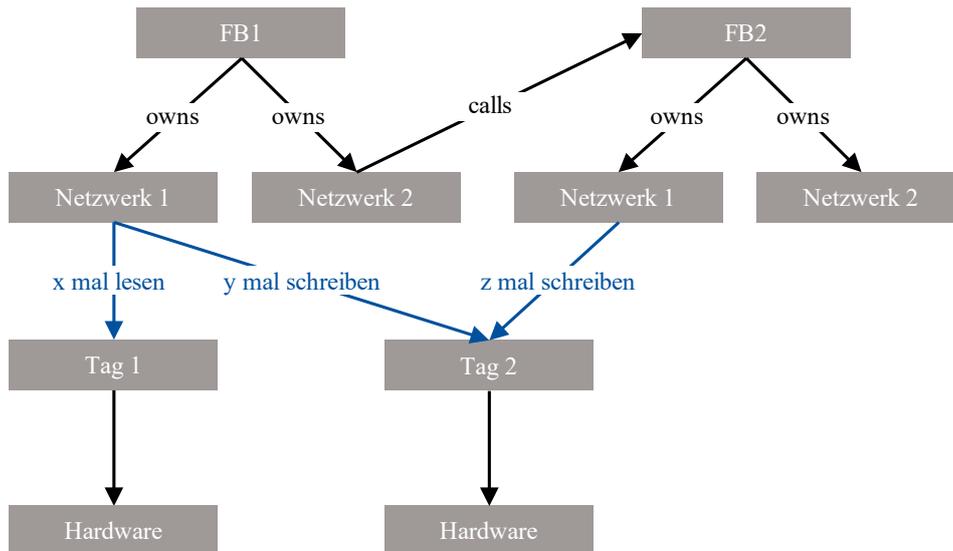


Abbildung 4.4: Generalisierter Aufbau eines SPS-Codes

Der zweite Abhängigkeitspfad zwischen den Bausteinen FB1 und FB2 bildet gemeinsam genutzte Hardware. In der vorangehenden Abbildung ist dieser Abhängigkeitspfad durch den Zugriff beider Bausteine über ihr jeweiliges Netzwerk 1 auf den Tag 2 abgebildet. Diese Tags repräsentieren die Peripherie (Sensoren und Aktoren) in der Software und lassen sich daher auch, wie in Abbildung 4.4 dargestellt, jeweils direkt einer Hardware-Komponente zuordnen. Diese Art der Abhängigkeitspfade sind aufwendiger zu analysieren, da hier viele Zugriffe von verschiedenen Stellen im SPS-Code stattfinden können. Diese Zugriffe sind außerdem nicht immer direkt ersichtlich, wenn die Tags über Parameter von Bausteinen oder Variablen durchgereicht sind, anstatt direkt im Code verwendet zu werden. Deshalb wird die Belegung von Variablen und Parametern bei deren Instanziierung durch die Analysemethodik nachverfolgt und ausgewertet. Diese implizite Nutzung von Tags wird im ersten Analyseschritt aufgelöst und auf die Tags verlinkt. Zusammen mit der Markierung der Own-Call-Beziehungen ist damit die Grundlage für die funktionale Gruppierung der Peripherie und der Software-Bausteine gelegt.

Ausgehend von der Hauptfunktion als Einstiegspunkt eines jeden SPS-Codes wird der Code auf diese Abhängigkeitspfade analysiert und dabei die Informationen in eine netzwerkartige Struktur gebracht, welche die Abhängigkeiten abbildet. Den Beziehungen zwischen Bausteinen und Peripherie werden dabei die gemeinsam genutzten Tags als Attribut hinzugefügt. Ebenso werden die Informationen aus dem SPS-Code zu den Knoten verlinkt. Auf dieser Struktur wird nun die Bindungsstärke einzelner Bausteine analysiert.

Diese Bindungsstärke wurde eingeführt, um eine Aussage treffen zu können, welche Bausteine zusammen eine Funktionsgruppe bilden und welche Hardware in Form von Tags dazugehört. Dabei wird anhand eines berechneten Wertes bestimmt, wie stark die Bindung zwischen einzelnen Blöcken ist. Die Bindungsstärke eines Funktionsbausteines Eins ermittelt sich aus der Anzahl der

assozierten Funktionsbausteinen und deren gemeinsamen Tags. Dieser Zusammenhang wurde mit dieser Formel mathematisch ausgedrückt:

$$\text{binding}_1 = (1 - \text{norm}(a_1)) + \text{norm}(t_1) \quad (1)$$

Dabei ist die Anzahl der weiteren Funktionsbausteine, mit denen sich ein Funktionsbaustein Tags teilt und eine Abhängigkeitsbeziehung besitzt, in der Variable a_1 angegeben und in der Menge der Block-Assoziationen $A = \{a_1, a_2, \dots, a_n\}$ aller Funktionsblöcke enthalten. Die Variable t_1 gibt an, wie viele Tags in Summe ein Funktionsbaustein mit anderen Funktionsbausteinen gemeinsam hat. Diese Variable ist Teil der Menge $T = \{t_1, t_2, \dots, t_n\}$, welche die Zahl gemeinsamer Tags für jeden Funktionsblock beinhaltet. Die *norm*-Funktion dient der Normierung des übergebenen Werts a_i beziehungsweise t_i über alle Werte der jeweiligen Menge auf einen Wertebereich von $0 \leq \text{norm}(x) \leq 1 \mid \text{norm}(x) \in \mathbb{R}$ nach der folgenden Formel:

$$\text{norm}(a_i) = \frac{a_i - \min(A)}{\max(A) - \min(A)} \text{ bzw. } \text{norm}(t_i) = \frac{t_i - \min(T)}{\max(T) - \min(T)} \quad (2)$$

Diese Bindungsstärke stuft Bausteine mit geringer Vernetzung zu anderen Bausteinen aber mit vielen gemeinsamen Tags als aussagekräftig ein. Ebenso werden Bausteine ohne Vernetzung zu anderen Bausteinen als aussagekräftig eingeordnet, da diese klar abgegrenzt werden können. Als wenig aussagekräftig werden Bausteine beurteilt, die mit vielen anderen Bausteinen verbunden sind, sich jedoch nur wenige Tags teilen. Diese Bausteine bilden eine diffuse Abhängigkeit und sind nicht eindeutig voneinander abgrenzbar. Daraus resultieren funktionale Gruppen, die so groß wie nötig sind, jedoch so klein wie möglich bleiben. Die Trennung der Bausteine anhand dieses Wertes der Bindungsstärke erfolgt dynamisch basierend auf dem Mittelwert der Bindungsstärke aller Bausteine. Dadurch wird berücksichtigt, dass unterschiedliche Programmierstile zu verschiedenstarker Trennung von einzelnen Funktionen in separaten Bausteinen oder zur Vermischung und Vernetzung durch gemeinsame Nutzung von Tags in mehreren Bausteinen führen kann. Die Trennung in die funktionalen Gruppen, basierend auf dem projektbezogenen Mittelwert der Bindungsstärken, erhöht die Anwendbarkeit auch bei verschiedenen Stilen und Vorgaben durch Vorschriften hinsichtlich des SPS-Codeaufbaus. Die Zuordnung der Peripherie zu den funktionalen Gruppen wird des Weiteren noch über eine einfache Summe quantifiziert. Hierfür wird für die Tags berechnet, wie viele Zugriffe alle Bausteine einer funktionalen Gruppe auf diesen besitzen. Die Beziehung des Tags zu den funktionalen Gruppen wird mit dieser Summe als Attribut versehen. Dadurch kann entschieden werden, welcher funktionalen Gruppe die Hardware des assoziierten Tags zugeordnet wird, wenn mehrere funktionale Gruppen auf einen Tag zugreifen. Die Beziehung mit dem höchsten Wert wird als essenzielle Zugehörigkeit markiert, alle weiteren Beziehungen des Tags zu anderen Funktionsgruppen bleiben bestehen, werden im weiteren Verlauf jedoch ignoriert.

4.3.3 Zusammenfassung der Analyseergebnisse

Die entwickelte Analysemethode importiert nativen SPS-Code, überführt diesen in ein unabhängiges Informationsmodell und untersucht die Struktur, um Wissen über die funktionalen Relationen der Komponenten eines Brownfield-Produktionssystems zu extrahieren. Als Teilschritte werden dabei bestehende Informationen aus dem SPS-Code strukturiert und angereichert. Das extrahierte Wissen ermöglicht die Beantwortung zusätzlicher Fragen zum funktionalen Aufbau des bestehenden Produktionssystems. Somit entsteht in der Analysemethode Wissen über das Produktionssystem nach der Definition von Rowley [147]. Dieses Wissen beinhaltet außerdem auch die bereits explizit im SPS-Code enthaltenen Informationen bezüglich der verbauten Hardware und deren Eigenschaften, wie deren Typ (Sensor/Aktor, digital/analog), deren Portnummer, Hardwareadresse und deren Anschluss an der Automatisierungshardware. Diese Informationen über den elektrischen Aufbau sind bereits im SPS-Code enthalten und werden im Rahmen der Analyse extrahiert, strukturiert und mit dem Wissen zusammengeführt. Dabei werden Informationen zu den Zusammenhängen mit der Software und den Software-Bausteinen berücksichtigt.

Abbildung 4.5 zeigt einen symbolischen Ausschnitt eines derartigen Analyseergebnisses. Die detektierten funktionalen Gruppen und die Zuordnung der Software und Hardware zu diesen sind als blaue Kreise dargestellt. Die Zuordnung der Hardware (zweite von links), die in mehreren Funktionsbausteinen verwendet wurden, zu den extrahierten funktionalen Gruppen erfolgt anhand der Zugriffe. Die funktionale Gruppe eins hat mit sechs Zugriffen die höhere Anzahl. Die Hardware wird daher der ersten Gruppe zugeordnet.

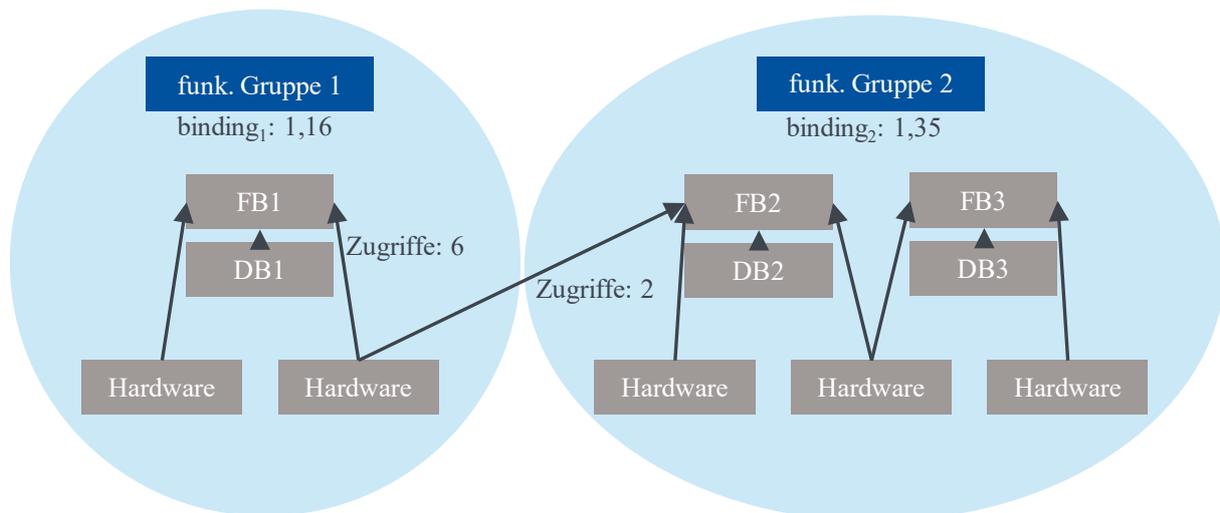


Abbildung 4.5: Beispiel eines Analyseergebnisses der SPS-Code-Analyse

Diese Netzwerkstruktur aller Bausteine und der Hardwareperipherie, deren funktionale Abhängigkeit und daraus identifizierten funktionalen Gruppen decken die Relationen im funktionalen Bereich (Software) ab und besitzen Schnittstellen zu den mechanischen und elektrischen Domänen. Diese gilt es, mit weiteren Informationen zu verknüpfen und anzureichern.

4.4 Datenbasierte Analysemethode der Positionsdaten zur Identifikation räumlicher Relationen

Die datenbasierte Analyse der Positionsdaten wird verwendet, um eine räumliche Ausdehnung des Produktionssystems aus den Positionen zu extrahieren und eine räumliche Gruppierung der Peripherie zu ermöglichen. Die Methodik verwendet die Positionen, um mit wenig Aufwand Aussagen über die physikalische Gruppierung der Anlagenabschnitte und Komponenten erlaubt in Anlehnung an Akanmu et al. [148]. Die Autoren haben in [148] die Verwendung von Echtzeit-Lokalisierungssystemen für die Aktualisierung von Komponentenpositionen im Gebäudesektor vorgestellt. In [138] haben die Autoren verschiedene Anwendungen aus der Intralogistik und dem Sektor der Automobilproduktion gelistet, wie zum Beispiel das Werkzeugtracking. Dieses wird beispielsweise in der Automobilindustrie eingesetzt, um ein direktes Feedback zu Bearbeitungsschritten zu geben oder um Verarbeitungsprozesse wie manuelle Verschraubungen zu dokumentieren, indem die Position des Werkzeugs, dessen Nutzungsdaten sowie die Position des Fahrzeugs gespeichert und analysiert werden. Dabei können hochpräzise Systeme basierend auf UWB-Technologie oder 5G Genauigkeiten von 10-20 cm erreichen [138]. Auch der Einsatz von selbstfahrenden, autonomen Transportsystemen und anderen autonomen, mobilen Robotern fördern den Einsatz von Lokalisierungssystemen.

Diese Positionsdaten gilt es zu strukturieren, aufzubereiten und entweder direkt der Anwendung als Streaming-Daten zuzuführen oder abzuspeichern. Dieser Methodenbaustein arbeitet nicht mit den unstrukturierten Streaming-Daten, sondern auf den gespeicherten, aufbereiteten Daten. Es wird, wie in 4.1 eingeführt, von den Positionsdaten als Informationsquelle gesprochen und der gespeicherte Datenbestand referenziert. Im Folgenden werden zuerst die Systeme zur Positionserfassung als Datenquelle betrachtet und die Erfassungsschritte sowie die Speicherung der Positionen dargelegt. Anschließend wird die eigentliche Analyse der Positionsdaten zur Extraktion des Wissens über das zugrundeliegende Produktionssystem und zur Klassifikation weiterer Positionen vorgestellt. Abschließend werden die Ergebnisse und das gewonnene Wissen zusammengefasst.

4.4.1 Erfassung und Speicherung der Positionsdaten als Informationsquelle

Die Erfassung der Positionsdaten ist durch verschiedene Technologien möglich: durch dedizierte Echtzeit-Lokalisierungssysteme (RTLS) für die Lokalisierung in Gebäuden basierend auf der UWB-Technologie oder auch mittels des Mobilfunkstandards 5G. Einfachere und kostengünstigere Varianten verwenden Bluetooth Low Energy, RFID oder WLAN. Diese bieten jedoch eine geringere Auflösung der Positionsgenauigkeit. Alle diese Systeme bestehen aus beweglichen Elementen (Tag), deren Position durch das ortsfeste Lokalisierungssystem (Anker) erfasst wird. Die Tags können aktiver oder passiver Natur sein und mittels unterschiedlicher

Methoden zur Positionsbestimmung durch die Anker ausgelesen werden. Dabei erzeugen die Anker, abhängig von der Methode zur Positionsbestimmung, grundlegende Daten wie beispielsweise einen Zeitstempel (Time difference of Arrival [TDOA]) oder eine Antwortdauer der Tags (Round Trip Time [RTT]). Der zentrale Knoten eines Lokalisierungssystems verarbeitet die Daten der Antennen und berechnet daraus die Position des Tags als höherwertige Information. Diese Information beinhaltet mindestens die Position eines bestimmten Tags zu einem Zeitpunkt und kann entweder direkt abgespeichert oder noch im System vorverarbeitet werden. Üblich ist eine Vorfilterung der Daten im Zentralknoten des Lokalisierungssystems. In diesem Zuge wird, abhängig von der Geschwindigkeit des Tags, etwa die Aktualisierungsrate der Position und damit die zeitliche Granularität an der Schnittstelle zur Datenverarbeitung für unbewegte Tags reduziert oder komplett auf null gesetzt. Darüber hinaus bieten manche Lokalisierungssysteme die Option, die Positionsgenauigkeit zu verbessern, indem mehrere Positionen über die Zeit gesammelt und Ausreißer vor der Übermittlung entfernt werden. Zusätzlich harmonisieren manche Systeme den Positionsverlauf mit Methoden, wie die des gleitenden Fensters, zur Berechnung eines gleitenden Mittelwerts. Teilweise werden die Informationen vor ihrer Bereitstellung über entsprechende Schnittstellen auch noch um die gegenwärtige Geschwindigkeit, Beschleunigung oder den Akkuzustand des Tags angereichert.

Die Ebene der Datenverarbeitung muss entsprechend auf die Schnittstellen, die bereits getätigten Aufbereitungen und Anreicherungen des Lokalisierungssystems angepasst sein. Stellt das Lokalisierungssystem nicht alle der benötigten Funktionalitäten zur Verfügung, können diese in dieser Ebene implementiert werden. Die Datenverarbeitungsschicht kennt dabei zwei grundlegend unterschiedliche Arbeitsweisen (siehe Abbildung 4.6). Zum einen können die entstehenden Informationen direkt verarbeitet und an die Anwendungen weitergereicht werden. Diese Livedaten werden auch Streaming-Daten genannt und in Pipelines verarbeitet. Zum anderen besteht die zweite Option, die Informationen in Dateien oder Datenbanken persistent zu speichern. In industriellen IT-Umgebungen ist zu erwarten, dass mindestens eine der beiden Optionen oder sogar eine Kombination der beiden verfügbar ist. Diese Methode zur Positionsdatenanalyse baut auf die asynchrone Datenbereitstellung über den Umweg einer Datenbank oder Datei und arbeitet nicht mit Streaming-Daten.

Der Methodenbaustein der Positionsdatenanalyse ist unabhängig davon, welche Technologie zur Lokalisierung und zur Erzeugung der Rohdaten verwendet wurde. Die Genauigkeit des Erfassungssystems beeinflusst lediglich die mögliche Genauigkeit der Aussagen durch die Analyse. Des Weiteren beeinflusst das Lokalisierungssystem durch seine Schnittstelle die Schritte der Datenerfassung und der Aufbereitung vor der persistenten Speicherung. Die Analyse ist abhängig von der Datenspeicherung hinsichtlich der enthaltenen Informationen und deren Aufbereitung. Benötigt werden die historischen Verläufe der Materialbewegungen im Produktionssystem sowie die mit den Positionen assoziierten Zeitstempel. Es ist gut, wenn ebenfalls die zu diesem Zeitpunkt aktuelle Bewegungsgeschwindigkeit verfügbar ist, jedoch

funktioniert die Methodik durch eine Rückfallebene auch ohne diese Information. Abgesehen von den eigentlichen Informationen über die Positionen der Tags werden Metadaten über die bereits durchgeführten Aufbereitungen benötigt. Es ist für die weitere Analyse relevant, wie weit die Positionsdaten im Lokalisierungssystem oder der Datenverarbeitung um stationäre Tags bereinigt wurden.

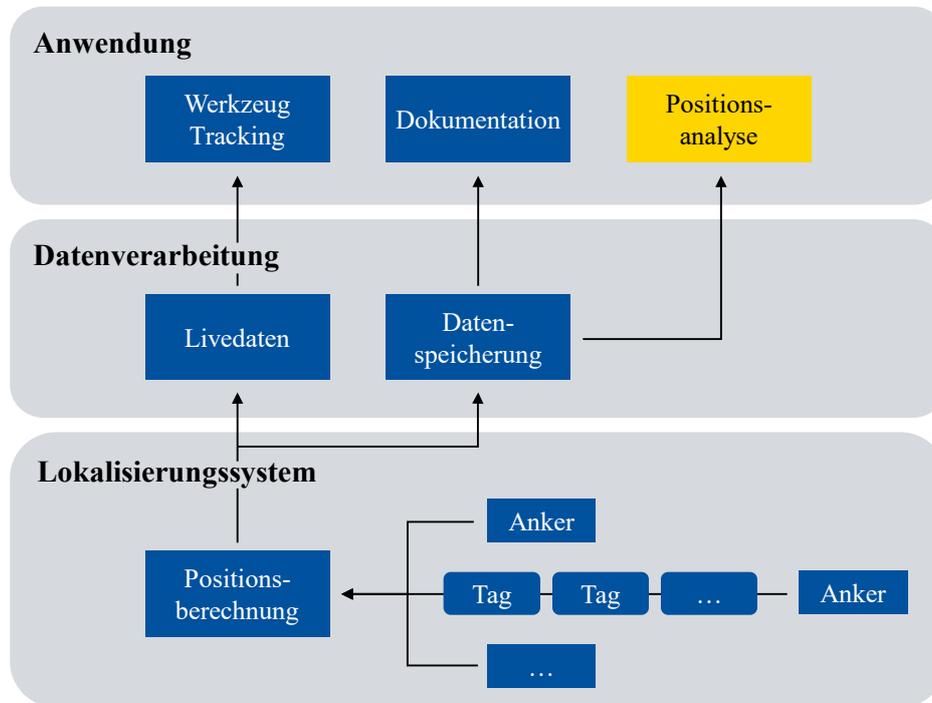


Abbildung 4.6: Schemata der Positionserfassung und Speicherung

Für die Speicherung der Positionen können unter anderem klassische relationale Datenbanken verwendet werden. Aufgrund der möglichen Datenmengen und der benötigten Zeitinformation sind jedoch auch Zeitreihendatenbanken eine sinnvolle Alternative. In der industriellen Automatisierung sind Zeitreihen die häufigste Art der Daten [149], für die mit Zeitreihendatenbanken eine optimierte Speichermöglichkeit hinsichtlich der Erfassungsrates, dem Zeitbezug und der Bereitstellung als Zeitreihe verfügbar ist. Bei einem minimalen System mit einem Tag und einem Erfassungssystem mit vier Anker⁸ sowie einer Erfassungsrates von 100 ms werden pro Sekunde 40 Rohdatenpunkte erzeugt beziehungsweise 30 verarbeitete Positionsinformationen (x/y/z). In realen Systemen wird normal mehr als ein Tag eingesetzt, denn es zu lokalisieren gilt, was zu einer linear steigenden Datenrate mit jedem weiteren Tag führt. Ebenso werden nicht nur die minimale Anzahl Anker verwendet, sondern zur Verbesserung der Positionsgenauigkeit durch reduzierte Abschattung mehr Anker verbaut. Da die Positionen keine Beziehungen zu weiteren Datenpunkten haben und einen Zeitbezug aufweisen, ist eine Zeitreihendatenbank zur Speicherung bei einer erhöhten Datenrate zweckmäßig. In dieser Art der Datenbank werden die Positionsdaten als Zeitreihen gruppiert nach ihrem Tag abgelegt.

⁸ Für eine zweidimensionale Positionsbestimmung ist bei zeitbasierten Verfahren mindestens drei Anker notwendig, für eine räumliche Positionsbestimmung mindestens vier. Richtungs-basierte Verfahren verwenden zwei beziehungsweise drei Antennenarrays.

4.4.2 Analyse der Positionen zur Wissensextraktion

Im Gegensatz zur Analyse des SPS-Codes ist die Analyse der Positionen eine datenbasierte Analyse. Das Ziel ist es, anhand der Positionsverläufe die Ausdehnung des Produktionssystems und speziell die Segmentierung in physikalische Gruppen zu identifizieren. Da für die bestehenden Produktionssysteme kein einheitliches Regelwerk existiert, nach dem sich ein System untergliedern lässt, kommen hier datenbasierte Verfahren zum Einsatz. Mit diesen Verfahren werden Muster oder Systematiken in den Daten gesucht und Zusammenhänge in den zeitlich und räumlich variierenden Daten identifiziert. Im Bereich des maschinellen Lernens kommt dabei die Klassifizierung und das Clustering in Frage.

Das Clustering als unüberwachtes Lernverfahren ist eine Option, die ohne zusätzlichen Labeling-Aufwand verwendet werden kann. Jedoch kann diese Klasse der Algorithmen zu ungewollter, physikalisch unbegründeter Gruppenbildung führen. Abbildung 4.7 visualisiert den vereinfachten Bewegungsverlauf eines einzelnen Tags mit dem Material in einem Produktionssystem. In der unteren Bildhälfte ist die gewollte Gruppierung dargestellt. In der oberen Hälfte sind die zu erwartende Resultate der beiden Klassen von Algorithmen dargestellt. Dabei sind die Tags vor und nach deren Stillstand (gelbe Positionsmarker) aufgrund eines Anlauf- oder Abbremsvorgangs langsamer und daher dichter angeordnet. Dies führt zur Häufung der Positionen rund um eine Ruheposition. Diese Anhäufung wird unter anderem noch verstärkt, wenn die Positionen ruhender Tags in derselben Frequenz wie bewegte Tags erfasst werden oder keine Filterung redundanter Positionen erfolgt. Daher ist als erste Maßnahme die Betrachtung von Bewegungssequenzen⁹ sinnvoll, anstatt direkt die Positionen zu analysieren. Dadurch wird die Verteilungsdichte als Gruppierungsfaktor ausgeschlossen und die physisch sinnhafte Gruppierung unterstützt. Des Weiteren bietet die Verwendung der Bewegungssequenzen den Vorteil, dass einzelne Positionen nicht knapp an der Gruppengrenze liegen und uneindeutig sind, sondern der Bewegungsverlauf aus mehreren Punkten besteht und dadurch in den meisten Fällen einer Gruppe eindeutiger zugeordnet werden kann.

⁹ Eine Bewegungssequenz besteht dabei aus einem Start- und Endpunkt, welche ruhende Positionen sind, und einem gemittelten Bewegungsablauf bestehend aus allen Positionen eines Tag zwischen diesen zwei Ruhepositionen.

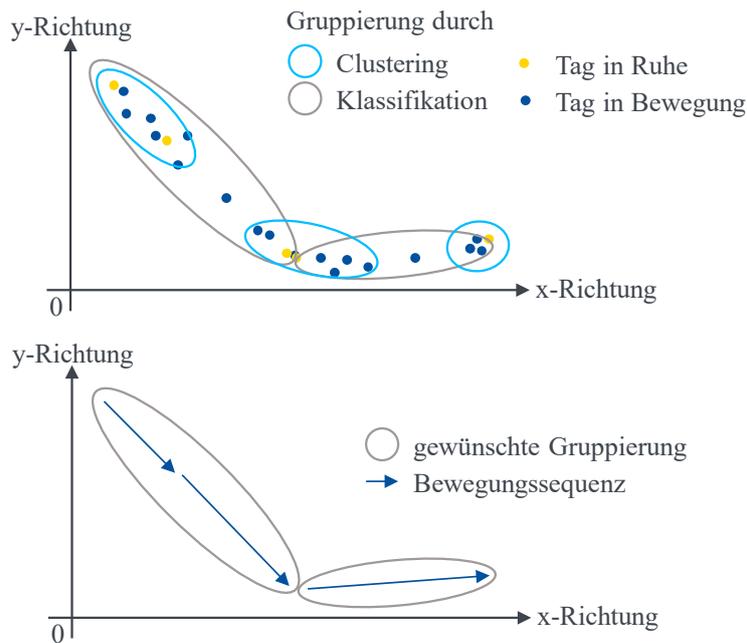


Abbildung 4.7 oben: unterschiedliche Gruppierung der Positionen durch zwei Klassen von Algorithmen; unten: die erwünschte Gruppierung mit Bewegungssequenzen

Darüber hinaus gibt es eine weitere Möglichkeit die Gruppierung der Positionen zu ermitteln, die einfacher beeinflusst werden kann hinsichtlich der zu bildenden Gruppen. Die Analyse der Positionen mittels einer Klassifikation als überwachtes Lernverfahren ermöglicht eine gezielte Aufteilung in Klassen, wie in Abbildung 4.7 dargestellt. Die benötigten Labels im Trainingsdatensatz eines überwachten Lernverfahrens sind dabei ein nachteiliger Aufwand im Vergleich zum Clustering. Diese Labels müssen bei der Erfassung der Positionsdaten oder im Nachgang den Positionen angefügt werden. Dieser Mehraufwand kann im Gegenzug zu einer treffenderen Gruppierung führen. Prinzipiell kann diese Aufgabe sowohl durch ein Clustering als auch durch eine Klassifikation gelöst werden und muss anwendungsfall-spezifisch entschieden werden. Die übergeordnete Methodik benötigt am Ende lediglich eine gute Gruppierung der Bewegungssequenzen nach den physikalischen Gruppen (räumlich zusammengehörige Komponenten der Anlage).

Die Analyse der Positionen und deren Gruppierung wird im Weiteren als Klassifikationsproblem betrachtet, konkret als ein multivariantes Multiklassen-Klassifikationsproblem aufgrund der geplanten Zuweisung zu einer von mehreren Klassen für die mehrdimensionale Eingangsgröße Position (x, y, z) als multivariate Zeitreihe [150]. Klassische Feature-basierte Verfahren sind hierfür nicht direkt einsetzbar, da keine Features vorliegen und erst eine aufwendige Datenvorverarbeitung und -Transformation benötigt wird, um geeignete Features herauszuarbeiten [150]. Dies ist eine herausfordernde Aufgabe, die in anderen Arbeiten untersucht und unter anderem durch LSTM [151, 152] oder CNNs gelöst wurden [153]. Diese Lösungen benötigen für jedes neue Produktionssystem eine große Menge an Trainingsdaten und Trainingsaufwand. Die Erstellung dieser gelabelten Datensätze und das Training steht in keinem Vergleich zum Aufwand, die Anlage manuell zu vermessen und die Gruppen händisch

festzulegen. Im Vergleich zu distanzbasierten Klassifikationsverfahren bieten sie hingegen eine höhere Aussagegenauigkeit [154]. In [155] wurde der k-nearest Neighbors (kNN) Algorithmus als distanzbasiertes Klassifikationsverfahren untersucht und als effizienter Ansatz zur Klassifikation von Zeitreihendaten identifiziert. Ebenso kommt die Forschung zu dem Schluss, dass kNN speziell mit dynamic time warping (DTW) als Distanzmetrik ein robuster, effizienter Ansatz ist, der präzise Ergebnisse liefert und nur durch komplexe Modelle in der Aussagegenauigkeit geschlagen wird [150, 154, 156]. Ein Kompromiss ist die Kombination mehrerer paralleler Verfahren, deren Ergebnisse am Ende zu einer Gesamtvorhersage zusammengeführt werden. Das sogenannte Ensemble Learning führt ebenfalls zu einer sehr guten Aussagegenauigkeit [157]. Die Verwendung verschiedener Verfahren erzeugt jedoch einen hohen Aufwand hinsichtlich der Datenaufbereitung für alle genutzten Verfahren sowie für die Implementierung und das Training der parallelen Klassifikationsverfahren. Ein Vergleich dieser Verfahren hinsichtlich deren Eignung für diesen Methodenbaustein ergibt die Gegenüberstellung nach Tabelle 2. Die Kriterien wurden mit einem Punkt für eine schlechte Erfüllung der Eigenschaft (großer Datensatz, viel Aufbereitung, lange Dauer, geringe Genauigkeit) und mit bis zu drei Punkten für erfüllte Eigenschaften bewertet. Durch den geringen zeitlichen Aufwand und erforderliche Datenmenge eignet sich nach diesem Vergleich der kNN-Algorithmus mit DTW am besten als Klassifikationsverfahren für die Identifikation der räumlichen Gruppen. Prinzipiell kann auch eine andere Klassifikation verwendet werden, wenn für das zugrunde liegende Produktionssystem entsprechend genügend Daten vorliegen und die notwendigen manuellen Untersuchungsaufgaben, beispielsweise aus Komplexitätsgründen des Produktionssystems, mehr Zeit benötigen als das Klassifikationsverfahren. Ebenso kann die Verwendung eines anderen Verfahrens sinnvoll sein, wenn nicht nur eine Strukturierung vorgenommen werden soll, sondern auch präzise Aussagen für die Modellbildung als Grundlage für den Digitalen Zwilling benötigt werden. Die mehrfache Nutzung relativiert in diesem Fall den benötigten zeitlichen Aufwand.

Tabelle 2: Vergleich von Klassifikationsverfahren für Zeitreihen

	kNN mit DTW	LSTM	Ensemble
Datensatzgröße	3	1	3
Datenaufbereitung	3	2	1
Trainingsdauer	3	1	2
Implementierungsdauer	3	2	1
Genauigkeit	2	3	3
Summe	14	9	10

Der trainierte Klassifikator wird anschließend verwendet, um die Sensoren und Aktoren anhand der berechneten ungefähren Positionen aus dem Methodenbaustein in 4.5 geometrisch zu gruppieren und damit eine Zuordnung im Ortsaspekt zu schaffen.

4.4.3 Zusammenfassung der Analyseergebnisse

Dieser Methodenbaustein resultiert in zwei Ergebnissen: einem Wissensgewinn über das Brownfield-Produktionssystem hinsichtlich seiner räumlichen Ausdehnung und einem trainierten Klassifikator zur Gruppierung zusammengehöriger Positionen. Der Wissensgewinn der Analyse der Positionsdaten beziehungsweise Bewegungssequenzen entstammt aus der Klassifikation, die die Positionen zu definierten Gruppen mit derselben räumlichen Zugehörigkeit ordnet. Aus diesen Gruppen können die grobe Ausrichtung und Form der Anlage abgeleitet werden. Das zentrale Ergebnis dieses Methodenbausteins ist der trainierte Klassifikator, der für die Gruppierung der Peripherie-Positionen aus Kapitel 4.5 benötigt wird. Durch den Einsatz des Klassifikators in Kombination mit den Ergebnissen der Betriebsdatenanalyse wird die Peripherie auch diesen physischen Gruppen zugeteilt.

Dieser Baustein der Methodik erzeugt durch das Labeling der Daten eine zusätzliche manuelle Aufgabe, deren zeitlicher Aufwand in der Evaluierung gegen die benötigte Zeit des klassischen manuellen Vorgehens abgewogen werden muss. Alternativ kann als Methode zur Gruppierung der Positionen auch ein unüberwachtes Clustering-Verfahren genutzt werden, jedoch resultiert dies nicht sicher in einer physisch zielgerechten Gruppierung.

4.5 Korrelationsbasierte Analyseverfahren zur Berechnung der Peripheriepositionen

Um die örtlichen Relationen zwischen den Komponenten des Brownfield-Produktionssystems abbilden zu können, werden die Positionen der Peripherie benötigt. Diese können dann mit dem Klassifikator der vorhergehenden Methode gruppiert werden. Zur Bestimmung der ungefähren Position der Peripherie wird in dieser Methode eine weitere Informationsquelle herangezogen, da die bisher eingeführten Informationsquellen keine Aussage über die Position der Peripherie erlauben. Die optischen Verfahren sind dafür auf den ersten Blick die geeignete Wahl, da sie optisch die Position erfassen. Sie haben jedoch ein Problem bei der Identifikation der Peripheriegeräte. Da diese nicht vorrangig an exponierten Stellen montiert sind, können sie durch optische Verfahren nicht kartographiert werden. Außerdem werden Sensoren beziehungsweise Aktoren des gleichen Typs oft mehrfach in einem Produktionssystem verbaut, wodurch eine eindeutige Identifikation und darüber hinaus eine Zuordnung zu den Signalen des SPS-Codes schwierig bis unmöglich ist. Daher muss die Position der Peripherie mit einer anderen Methode aus einer anderen Quelle bestimmt werden, die eine Verknüpfung zu den Signalen der SPS ermöglicht. Die Betriebsdaten des Produktionssystems bieten alle nötigen Eigenschaften, da sie direkt mit den Signalen der SPS verknüpft sind. Jedoch fehlen diesen Daten die benötigten Informationen über örtliche Eigenschaften, um daraus die Position zu ermitteln. Daher werden zusätzlich Positionsdaten über das Produktionssystem für diese Analyse benötigt. Die Positionsdaten des Materials in dem Produktionssystem aus Kapitel 4.4.1 sind eine geeignete

Quelle, da die Bewegung des Materials im Produktionssystem zu Änderungen von Sensorsignalen führt und durch Aktoren beeinflusst wird. Die Methode in diesem Kapitel analysiert die Korrelation zwischen den Betriebsdaten und den Positionsdaten des Materials, um daraus die Position der Peripherie zu bestimmen [158]. Das folgende Unterkapitel beschäftigt sich mit den Betriebsdaten, deren Erfassung, Vorverarbeitung und Speicherung. Die Positionsdaten wurden ausführlich in 4.4 beschrieben, da diese dort als einzige primäre Quelle dienen, und werden hier nur kurz behandelt. Anschließend werden die Analyseschritte der Daten dargelegt und abschließend die Ergebnisse sowie der Wissensgewinn durch die Analyse beschrieben.

4.5.1 Erfassung und Speicherung der Betriebsdaten als Informationsquelle

Die Grundlage dieser Analyse sind die Betriebsdaten des Produktionssystems aus den vergangenen Produktionsdurchläufen sowie die Positionsdaten des Materials im selben Zeitraum. Die Betriebsdaten bezeichnen dabei die Werte der Sensoren und Aktoren des Produktionssystems und können, je nach vorhandener Kommunikationsarchitektur der Automatisierungshardware, über verschiedene Wege erfasst werden. Dabei ist es für die spätere Zusammenführung mit den funktionalen Relationen wichtig, dass die Namen der SPS-Signale miterfasst werden und nicht nur deren Werte mit einer generischen Kennung. Eine Erfassung der Betriebsdaten kann über einen Abgriff der bestehenden Kommunikation der Betriebsdaten an das übergeordnete Prozess- oder Betriebsleitsystem stattfinden oder gegebenenfalls gespeicherte Daten aus diesen Bereichen genutzt werden. Darüber hinaus können die Betriebsdaten auch über dedizierte Edge-Geräte, die an der SPS angeschlossen und die Daten zum Beispiel an eine Cloud übertragen, aufgezeichnet werden. Diese werden unter anderem für die vielen Big Data Anwendungen benötigt, die in den vergangenen Jahren entstanden sind, wie eine datengetriebene vorausschauende Wartung oder zur Erweiterung klassischer Entscheidungsunterstützung [159]. Fernerhin sind viele der heute eingesetzten SPS in der Lage, mittels OPC UA oder MQTT, ergänzend zur Anlagensteuerung Betriebsdaten bereitzustellen und damit eine parallele Erfassung zu ermöglichen.

Da während der Analyse kein Vergleich der Signalwerte untereinander oder eine Auswertung der Signalwerte erfolgt, sondern nach einer Änderung des Signalwerts gesucht wird, ist es nicht relevant, den Wertebereich oder die Codierung des Wertes zu definieren. Die Harmonisierung der Werte ist gerade bei der Nutzung von Daten, die für einen anderen Zweck erhoben wurden, normal und wirkt sich nicht negativ auf die Analyse in dieser Methode aus. Die Erfassungsrate der Betriebsdaten richtet sich dabei nach der Änderungsrate der Betriebsdaten und sollte schneller sein als der Wechselzyklus (hin und zurück) eines Signals, um keine Signaländerung zu verpassen. Diese Bedingung muss eine Steuerung für ihre Steueraufgaben bereits erfüllen, daher ist eine Erfassungsrate ähnlich der Taktrate der SPS geeignet. Eine Reduzierung der Datenmenge kann durch eine intelligente Versionierung der aufgezeichneten Daten erreicht werden, da lediglich die Zeitpunkte der Signaländerung benötigt werden. Es werden dabei nicht nur die

gegenwertigen Signale der Sensoren und Aktoren benötigt, sondern die Verläufe der einzelnen Peripherie als aufgezeichnete Zeitreihen. Auch für diese Betriebsdaten stellen Zeitreihendatenbanken eine optimierte Speichermöglichkeit hinsichtlich der Erfassungsrate, dem Zeitbezug und der Bereitstellung als Zeitreihe dar.

4.5.2 Analyseschritte

Die Korrelationsanalyse der Betriebsdaten und der Materialpositionen (in Abbildung 4.8: Positionsdaten) erfolgt gemeinsam nach den in Abbildung 4.8 dargestellten Schritten. Hintergrund für diese Analyse ist, dass die Materialbewegung einen Wechsel der Sensorsignale auslöst und die Aktoren zu einer Änderung der Bewegung des Materials führen. Diese Korrelation zwischen den Daten wird durch die Analysepipeline identifiziert und daraus eine ungefähre Position der Peripherie berechnet.

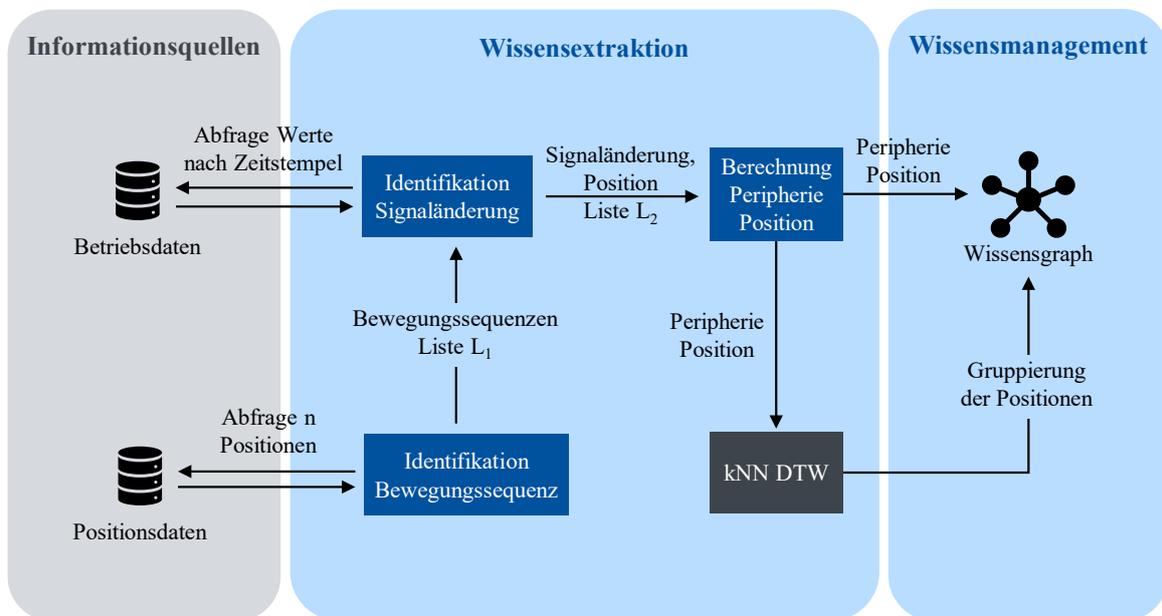


Abbildung 4.8: Analyseschritte zur Bestimmung der Position der Peripherie

Startpunkt der Analyse sind die Materialpositionen, die in der Zeitreihendatenbank der Positionsdaten vorliegen. Sind diese bei der Erfassung bereits vorverarbeitet und als abgeschlossene Bewegungssequenzen B ,

$$B = \left\{ \begin{pmatrix} \vec{p}_n \\ t_n \end{pmatrix}, \begin{pmatrix} \vec{p}_{n+1} \\ t_{n+1} \end{pmatrix}, \dots, \begin{pmatrix} \vec{p}_m \\ t_m \end{pmatrix} \right\} \quad (3)$$

bestehend aus der Positionsinformation \vec{p} und dem dazugehörigen Zeitstempel t , gruppiert worden, können sie direkt verwendet und die Identifikation der Bewegungssequenzen übersprungen werden. Im Regelfall liegt die Bewegung lediglich als Zeitreihe von Positionen $\vec{p} \in \mathbb{Q}^2 | \mathbb{Q}^3$ des Materials mit Zeitstempel vor. Es müssen deshalb zuerst die abgeschlossenen Bewegungssequenzen eines Materials identifiziert und die Positionen gruppiert werden. Dazu

werden iterativ n aufeinanderfolgende Positionen eines Materials aus der Zeitreihendatenbank abgefragt und nach dem Ende einer Bewegungssequenz mittels mehrerer Kriterien gesucht (Abbildung 4.8: Identifikation Bewegungssequenz). Die Positionen einer identifizierten Bewegungssequenz werden als solche gruppiert und weitergereicht. Alle Positionen, die in dieser Iteration keiner Bewegungssequenz zugeordnet werden konnten, werden in der nächsten Iteration zu den neuen n Positionen hinzugenommen. Zur Identifikation einer abgeschlossenen Bewegungssequenz werden drei Kriterien verwendet:

- Als erstes Kriterium wird untersucht, ob die zeitliche Erfassungsfrequenz der Positionen variiert, weil viele Lokalisierungssysteme die Erfassungsfrequenz für unbewegte Positionstransponder reduzieren. Wird eine Unterbrechung zwischen zwei Erfassungszeitpunkten festgestellt, die größer ist als das fünffache¹⁰ der normalen Zykluszeit, wird davon ausgegangen, dass entweder die Erfassung abgebrochen ist oder das System die Übermittlung eines unbewegten Positionstransponders unterbrochen hat. In beiden Fällen wird überprüft, ob die nächste erfasste Position gleich ist wie vor der Erfassungsunterbrechung/-pause, um einen Ausfall des Lokalisierungssystems oder des Übertragungssystems auszuschließen.
- Als weiteres Kriterium werden die Positionen untersucht und mit den vergangenen beziehungsweise folgenden Positionen verglichen. Dabei werden alle zeitlich aufeinanderfolgenden, räumlich identischen Positionen als ruhende Position zwischen zwei Bewegungssequenzen eingestuft und zusammengefasst. Die verbleibenden Positionen, die zeitlich zwischen zwei ruhenden Positionen liegen, werden als Bewegungssequenz eingestuft. Da die Positionserfassung Schwankungen unterliegt, sind die ruhenden Positionen nicht perfekt identisch, sondern leicht unterschiedlich und bilden einen quasi-stationären Zustand. Das heißt, die Differenz des gleitenden Mittelwerts der ruhenden Positionen liegt nahe Null und bildet auf der zeitlich makroskopischen Ebene eine stabile Position. Daher wird der gleitende Mittelwert der Positionswerte (4) verglichen und n so lange erhöht

$$\left| \frac{1}{3} \sum_{x=n}^{n+2} (\vec{p}_x) - \frac{1}{3} \sum_{x=n+1}^{n+3} (\vec{p}_x) \right| = \varepsilon \quad (4)$$

bis eine unveränderte Position ($\varepsilon \approx 0$) gefunden und eine ruhende Position identifiziert ist.

- Das dritte Kriterium kann nur genutzt werden, wenn der Positionstransponder ebenfalls die aktuelle Geschwindigkeit überträgt und diese mit in der Zeitreihendatenbank abgelegt ist. In diesem Fall kann über eine einfache Abfrage geprüft werden, ob die aktuelle Geschwindigkeit

¹⁰ Die Grenze eines Fünffachens der normalen Zykluszeit unterliegt keinem harten Kriterium und kann auch mehr oder weniger gewählt werden. Um robust gegen Übertragungsfehler zu sein, sollte mindestens das Zwei- bis Dreifache gewählt werden. Aufgrund der typischen Zykluszeit von wenigen hundert ms wurde das Fünffache gewählt, sodass eine Pause ungefähr im Sekundenbereich liegt.

an dieser Position null entspricht und damit eine ruhende Position ist. Alternativ kann die Geschwindigkeit auch über die Distanz zwischen Positionen und der verstrichenen Zeit berechnet werden.

Die identifizierten Bewegungssequenzen werden als Liste $L_1 = \{B_1, B_2, \dots, B_n\}$ weitergereicht, um korrelierende Änderungen in Signalen in diesen Zeiträumen zu identifizieren (Abbildung 4.8: Identifikation Signaländerung). Dazu werden die Signalwerte im Zeitraum der Bewegungssequenzen abgefragt und pro Signal nach einer Änderung des Signalwerts gesucht. Jeder Signaländerung wird die Position in der Bewegungssequenz zu diesem Zeitpunkt zugeordnet. Für alle Bewegungssequenzen in L_1 werden diese Paare aus dem Signalname des SPS-Signals s und die Position \vec{p} gesammelt und in einer weiteren Liste $L_2 = \left\{ \begin{pmatrix} s_1 \\ \vec{p}_1 \end{pmatrix}, \begin{pmatrix} s_2 \\ \vec{p}_2 \end{pmatrix}, \dots, \begin{pmatrix} s_n \\ \vec{p}_n \end{pmatrix} \right\}$ gespeichert. Diese Liste L_2 dient als Eingangsgröße für die Berechnung der Position der Peripherie.

Für die Berechnung der Peripherie Positionen werden zuerst die Positionen in der Liste L_2 mit demselben Signalnamen gruppiert, sodass eine Auflistung aller gesammelten Positionen zu jedem Sensor beziehungsweise Aktor vorliegt.

$$\left\{ \begin{array}{l} s_a: \{ \vec{p}_1, \vec{p}_3, \vec{p}_4 \dots \} \\ s_b: \{ \vec{p}_6, \vec{p}_8 \dots \} \\ \vdots \end{array} \right\} \quad (5)$$

Gesammelte Positionen eines jeden Signals werden zuerst um die Ausreißer bereinigt. Lokalisierungssysteme mit einer großen Schwankungsbreite der Position profitieren von der Entfernung der Ausreißer, da hierdurch die ermittelte Position genauer wird. Das hierzu verwendete Verfahren ist nicht entscheidend und kann frei gewählt oder prinzipiell sogar übersprungen werden. Beispielsweise kann hierzu ein komplexes Filterverfahren aus der sklearn Bibliothek für maschinelles Lernen verwendet werden oder eine einfache Bereinigung anhand der Schwankungsbreite der Position vorgenommen werden. Dazu wird für jedes Signal die Standardabweichung σ der zugehörigen Positionen vom jeweiligen Schwerpunkt \bar{p} berechnet und zur Bewertung der Positionen herangezogen.

$$\sigma = \sqrt{\sum_{i=1}^n |\vec{p}_i - \bar{p}|^2} \quad \text{mit} \quad \bar{p} = \frac{1}{n} \sum_{x=1}^n \vec{p}_x \quad (6)$$

Alle Positionen \vec{p}_i deren Abstand vom Schwerpunkt \bar{p} größer als die Standardabweichung ist, werden als Ausreißer eingestuft und aus der Menge in (5) aussortiert.

$$\vec{p}_i = \begin{cases} |\vec{p}_i - \bar{p}|^2 > \sigma: \text{Ausreißer} \rightarrow \vec{p}_i \text{ aus (5) entfernen} \\ |\vec{p}_i - \bar{p}|^2 \leq \sigma: \text{kein Ausreißer} \rightarrow \vec{p}_i \text{ verbleibt in (5)} \end{cases} \quad (7)$$

Alternativ kann auch ein Vielfaches von σ als Grenze gewählt werden, um weniger Positionen auszusortieren, zum Beispiel, wenn nur wenige Datenpunkte pro Position verfügbar sind. Die Wahl von 1σ schließt ca. 68% der Positionen in die Berechnung ein, die Verwendung von 2σ bereits etwa 94%. Anschließend wird für jedes Signal ein neuer Schwerpunkt berechnet basierend auf der bereinigten Menge der Positionen. Dieser Schwerpunkt ist die ungefähre Position der Peripherie und kann aufgrund dessen, dass der Signalname aus der SPS verwendet wurde, dem extrahierten Wissen über die funktionalen Gruppen aus Kapitel 4.3 über diese Namen zugeordnet werden.

4.5.3 Zusammenfassung der Analyseergebnisse

Das Ergebnis der Analyse der Betriebsdaten und deren Korrelation mit den Positionen des Materialflusses ist die Position der Peripherie. Diese Positionen sind zwar nicht exakt, jedoch ausreichend genau für die Bestimmung der räumlichen Relationen. Die Positionen werden dazu als Eingangsgröße für den Klassifikator des vorhergehenden Methodenbausteins verwendet. Dieser wurde darauf trainiert, die Positionen nach dem Ortsaspekt zu gruppieren und deren räumliche Relation dadurch zu extrahieren. Das Wissen über die Position der Peripherie und ihre räumliche Relation werden zusammen mit dem Signalnamen, der identisch ist mit dem Signalnamen in der SPS, im Wissensgraph abgespeichert. Der Signalname ist dabei relevant für die spätere Zusammenführung mit den Ergebnissen aus Kapitel 4.3. Die Methode aus diesem Kapitel berechnet nur die Position der Peripherie, die mit dem Material interagiert. Peripherie zur Absicherung der Anlage, zum Beispiel Sicherheitslichtschranken, Sicherheitsvorhänge und andere Schutzeinrichtungen, die nicht im Produktionsprozess verwendet werden, können durch die Methode nicht bestimmt werden, da keine Korrelation zum Produktionsprozess vorliegt. Außerdem stellen Normen und Vorschriften hohe Anforderungen an sicherheitskritische Einrichtungen, die diese Methode nicht erfüllen kann. Derartige Prozesse werden in der Industrie weiter von Experten durchgeführt und präzise kontrolliert.

4.6 Methodensynchronisation: Wissensmanagement

Die vorherigen drei Methodenbausteine beschreiben die automatische Extraktion von Wissen über die Relationen eines bestehenden Produktionssystems aus verschiedenen Quellen. Dabei liegt der Fokus auf den Relationen und Eigenschaften der Funktions-, Orts- und Produktperspektive der Peripherie des Produktionssystems. Das generierte Wissen besteht überwiegend aus Beziehungen zwischen Entitäten sowie deren Eigenschaften und muss nun zu einer Wissensbasis zusammengeführt werden, um die Ergebnisse der einzelnen Methoden zu synchronisieren. In 4.6.1 ist beschrieben, wie das erzeugte Wissen strukturiert ist und zusammengeführt wird. Anschließend werden in 4.6.2 Speichermöglichkeiten betrachtet und eine geeignete Speicherform für die Wissensspeicherung ausgewählt. In den folgenden beiden

Kapiteln werden Mechanismen besprochen, die das gespeicherte Wissen strukturieren, ergänzen und zugänglich für die weitere Nutzung machen.

4.6.1 Wissenszusammenführung

Das extrahierte Wissen aus den drei vorangehenden Methodenbausteinen betrifft überwiegend die Peripherie des Produktionssystems. Durch die Methodenbausteine werden erstens die Komponenten der Automatisierungshardware und die Peripherie aufgelistet, hardwaretechnische Eigenschaften hinzugefügt und deren Relationen zueinander aus der Funktionsperspektive ergänzt. Zweitens werden die physischen Positionen der Peripherie bestimmt und mit dem Signalnamen der Peripherie in Bezug gebracht. Drittens werden die Positionen der Peripherie aus der Ortsperspektive in Beziehung gesetzt. Dahinter stehen jeweils weitere Elemente mit Eigenschaften, wie Funktions- oder Datenbausteine der SPS, Automatisierungshardware oder Containerelemente zur hierarchischen Gruppierung. Dabei entsteht eine Wissensstruktur mit vielen Verbindungen zwischen den einzelnen Elementen. Beispielsweise erzeugt der Methodenbaustein der SPS-Code-Analyse Instanzen der Knoten und Kanten der Grundstruktur aus Abbildung 4.9.

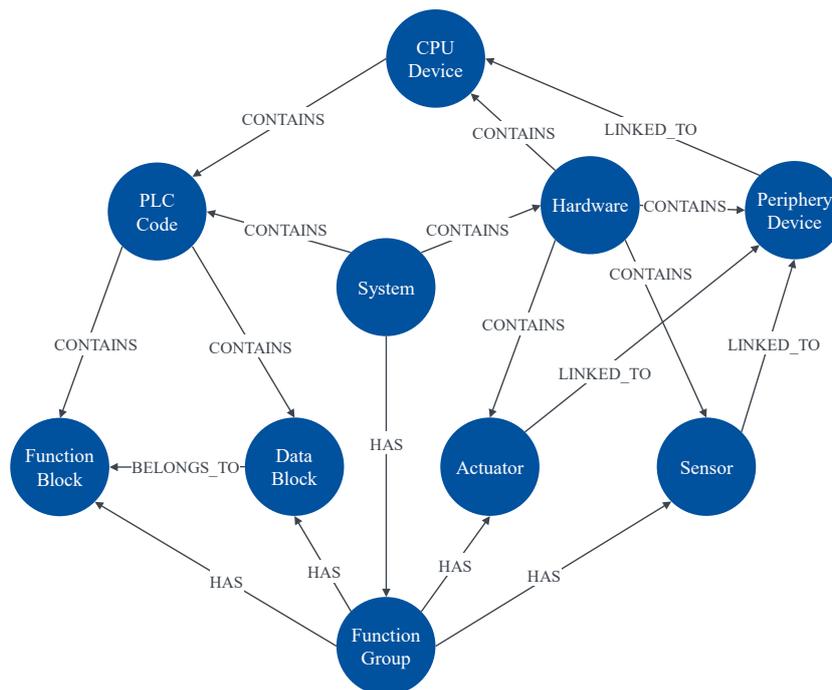


Abbildung 4.9: Grundstruktur des extrahierten Wissens mit der ersten Analysemethode

Dabei sind die Beziehungen zwischen den Knoten elementar, da sie die Relationen des Brownfield-Produktionssystems nachbilden. In begrenztem Umfang sind die Attribute der Knoten relevant für die Generierung der Relationen des Digitalen Zwillings. Die Struktur der beiden anderen Methodenbausteine sind vergleichbar und besitzen als Gemeinsamkeit die Knoten „Sensor“ und „Actuator“, welche die Peripheriekomponenten beschreiben. Dabei wird in allen Strukturen der Signalname aus der SPS als eindeutiger Identifikator verwendet und den

Analyseergebnissen zugeordnet. Der Signalname als Attribut der Knoten wird im Rahmen der Wissenszusammenführung genutzt, um identische Knoten zu verschmelzen und deren Attribute zusammenzuführen. Die Analysemethoden erzeugen kein Wissen, das sich in einem Knoten doppelt oder konkurriert, weshalb bei der Zusammenführung alle Attribute direkt übernommen werden.

Die Zusammenführung mit weiteren, externen Quellen ist eine Herausforderung. Quellen mit demselben Identifier, Granularität und Benennung sowie nicht konkurrierenden Attributen können nach demselben Prinzip eingefügt werden. Für alle weiteren, externen Quellen muss vorher eine Datentransformation durchgeführt sowie die Struktur, Benennung und Codierung der Attribute homogenisiert werden. Die semantische Anreicherung des Wissens über Ontologien, wie in Kapitel 4.6.3, ist ein Mechanismus, mit dem die anwendungsfallspezifische Integration von zusätzlichem, externem Wissen vereinfacht oder automatisiert werden kann. Dazu muss das Wissen aus der externen Quelle mit den Konzepten der Ontologien angereichert sein, um dann anhand dieser Konzepte mit dem generierten Wissen der Methodik kombiniert werden zu können. Alternativ kann das externe Wissen auch in ein neutrales Austauschformat oder das Speicherformat des Zielsystems, in dem der Digitale Zwilling erstellt werden soll, gespeichert und erst im Zielsystem integriert werden. Dieser Prozess ist, abhängig von den Funktionalitäten des Zielsystems und der Abdeckung des externen Wissens von der Ontologie mit mehr oder weniger manuellen Aufgaben verbunden als über die Ontologien.

4.6.2 Wissensspeicherung

Das aus den Analysemethoden resultierende Wissen zu den Relationen des Brownfield-Produktionssystems muss für weitere Untersuchungen, Ergänzungen sowie als Grundlage für die Erstellung Digitaler Zwillinge persistent gespeichert werden. Das erzeugte Wissen besteht, wie im letzten Kapitel beschrieben, überwiegend aus Beziehungen und Eigenschaften in den Wissensknoten. Die Relationen zwischen einzelnen Entitäten sind fast ausschließlich von der Kardinalität 1:n. Die Entitäten haben jedoch nicht nur eine Beziehung, sondern mehrere. Der Ausschnitt aus der Struktur der Wissens Elemente nach Abbildung 4.9 lässt erkennen, dass keine Elemente mit weniger als drei Beziehungen existieren. Mit der Ergänzung um die Elemente aus den anderen Methodenbausteinen erhöht sich die Zahl der Entitäten und Beziehungen zwischen diesen noch weiter. Das Wissen über jede der Perspektiven umfasst Entitäten, die mehrfach sequenziell und parallel verknüpft sind. Daher sind für deren Abfrage mehrfache Verknüpfungen von Entitätsklassen notwendig. Hierfür gilt es eine geeignete Speicherform zu wählen.

Prinzipiell können für die Speicherung dieses Wissens klassische relationale Datenbanken verwendet werden, solange die betrachteten Produktionssysteme klein sind. Bei steigender Größe des untersuchten Produktionssystems wächst das Wissen überproportional an. Dadurch steigt der Rechenaufwand durch die Menge der aufzulösenden Relationen mit Join-Operationen deutlich an [160]. Stanescu hat in [160] relationale Datenbanken mit Graphdatenbanken verglichen und

gezeigt, dass für einfache Joins die relationalen Datenbanken besser performen, jedoch bei drei und mehr Join-Operationen sowie komplexeren Abfragen deutlich den Graphdatenbanken unterlegen sind. Ein weiterer Nachteil von relationalen Datenbanken für die Wissensspeicherung sind die festen Strukturen der Datenbank [161]. Zusätzliche optionale Eigenschaften, die abhängig vom SPS-Hersteller im Code verfügbar sein können, zum Beispiel die Produktnummer der Hardware, sind dann nicht ohne zusätzliche Arbeitsschritte im Datenmodell der Datenbank ableibar.

Aufgrund des hierarchischen Aufbaus der Wissensstruktur ist auf den ersten Blick ebenfalls eine Speicherung in XML-basierten Dateiformaten denkbar. Im Vergleich zu einer reinen Baumstruktur sind in dieser Struktur jedoch Querverbindungen zwischen Elementen im Baum vorhanden sowie loopförmige Verbindungen. Die Speicherung eines vermaschten Baums beziehungsweise eines Netzwerks ist im XML-Format möglich, bringt jedoch Herausforderungen durch die Querverbindungen mit sich. Dabei können die Knoten, die in mehreren Zweigen vorkommen und diese verbinden, entweder mehrfach abgelegt sein, wodurch Redundanzen und die Gefahr einer Inkonsistenz entsteht. Oder die Knoten werden durch eine ID referenziert, die zu Suchvorgängen nach dem referenzierten Gegenstück führen. Diese Form der Speicherung in XML-basierten Formaten ist für eine effiziente Abfrage sowie die Analyse und Anreicherung der Wissensstruktur ungeeignet.

Graphdatenbanken, als Vertreter der NoSQL (Not only SQL) Datenbanken, sind für die Speicherung von Daten dieser Form optimiert und können die vorher genannten Einschränkungen anderer Speicherformen überwinden. Zur Speicherung von Wissen über die reale Welt werden Wissensgraphen ebenfalls schon seit einiger Zeit erfolgreich eingesetzt und sind spätestens seit der Veröffentlichung durch Google 2012 weitreichend bekannt [162]. Auch im Bereich der Digitalen Zwillinge werden Wissensgraphen bereits zur Speicherung heterogener Daten sowie zur Verlinkung, Vervollständigung, Fehlerdetektion und Schlussfolgerung auf und aus diesen Daten eingesetzt [163]. Die Verwendung einer Graphdatenbank zur Speicherung hat darüber hinaus noch Vorteile hinsichtlich der Extraktion impliziten Wissens im Graphen. Existierende Algorithmen zur Graphanalyse eignen sich, um die Struktur eines Wissensgraphen zu untersuchen und dieses implizite Wissen zu extrahieren [162]. Graphdatenbanken speichern die Daten nach unterschiedlichen Datenmodellen mit unterschiedlicher Struktur und entsprechenden Vor- und Nachteilen [162]. Die beiden bekanntesten und leistungsstärksten Graph-Datenmodelle sind der Triple Stores und der Labeled Property Graph [164]. Unter anderem durch das Resource Description Framework (RDF), das Informationen als Triple aus zwei Entitäten mit einer gerichteten Beziehung dazwischen darstellt, sind die Triple Stores bekannt. Im Vergleich dazu besitzen Labeled Property Graphs (LPG) Entitäten und Beziehungen mit eindeutigen Identifikatoren und eine zusätzliche interne Struktur, die ergänzende Eigenschaften der Elemente beinhaltet. Mit diesen Eigenschaften kann beispielsweise die Verbindung zwischen einem Sensor-Knoten und dem SPS-Knoten spezifiziert werden und eine Hardwareadresse, ein Datentyp sowie

ein Signalname hinterlegt werden. Dadurch ist es möglich Informationen in einem LPG kompakter zu speichern als in RDF und schneller abzufragen. Ein Vorteil von RDF ist die automatische Deduktion von implizitem Wissen durch formale Logik. Diese Fähigkeit wird jedoch für die Wissensspeicherung nicht zwangsläufig benötigt und erfordert zusätzliche Schritte in der Anpassung der Methode für neue Produktionssysteme. In diesem Methodenbaustein wird daher ein LPG für die Wissensspeicherung verwendet, um eine einfache Übertragbarkeit und eine effiziente Abfrage des Wissens für weitere Schritte zu erlauben. Im folgenden Abschnitt wird eine Anreicherung des gespeicherten Wissens im LPG mittels Ontologien beschrieben, welche die kleinen Nachteile des gewählten Graph-Datenmodells ausgleichen können.

4.6.3 Semantische Anreicherung mit Ontologien

Der gespeicherte Wissensgraph beinhaltet die wichtigen Elemente zur Erstellung der Relationen des Digitalen Zwillings. Für die Anbindung weiterer Quellen an den gespeicherten Wissensgraphen ist ein fundiertes Verständnis des Graphen notwendig, sodass die Ingenieure das zusätzliche Wissen der anderen Quelle in diesen Wissensgraphen integrieren können. Alternativ kann in einem nachgelagerten, eher manuellen Schritt das zusätzliche Wissen direkt in die Modelle des Digitalen Zwillings integriert werden. Zur Vereinfachung dieses Integrationsvorgangs, der maschinellen Verarbeitung des Wissens und der Nutzung des Wissens in einem erweiterten Kontext ist der Einsatz von Ontologien notwendig. Ontologien helfen ein Verständnis der Struktur zu teilen, das gespeicherte Wissen wiederzuverwenden, das Wissen zu analysieren, Schlüsse daraus zu ziehen und zu gruppieren [165]. Ebenso bilden Ontologien die Basis für Aufgaben wie ein Plausibilitätscheck oder die automatische Vervollständigung des Wissens im Digitalen Zwilling beziehungsweise im Wissensgraphen des Digitalen Zwillings [163].

Aus diesen Gründen unterliegt dem Wissensgraphen in diesem Methodenbaustein eine Ontologie. Es handelt sich dabei um eine sogenannte Lightweight Ontologie, welche im Gegensatz zu einer Heavyweight Ontologie keine mathematische Logik beinhaltet [166]. Im Rahmen der Methodik wurde, wie in [167] beschrieben, eine hybride Modellierung nach dem Vorbild von [168] mit hierarchisch aufbauenden Ontologien verwendet, auf denen ein LPG zur Instanziierung aufsetzt. Im Vergleich zur Forschung in [167] wird auf den systeminternen Bereich fokussiert und die Abbildung des systemübergreifenden Wissens sowie der externe Kontext vernachlässigt, da diese nicht Teil der vorliegenden Forschung sind. Die Wissensspeicherung ist ansonsten, wie in Abbildung 4.10 dargestellt, identisch zu [167] aufgebaut.

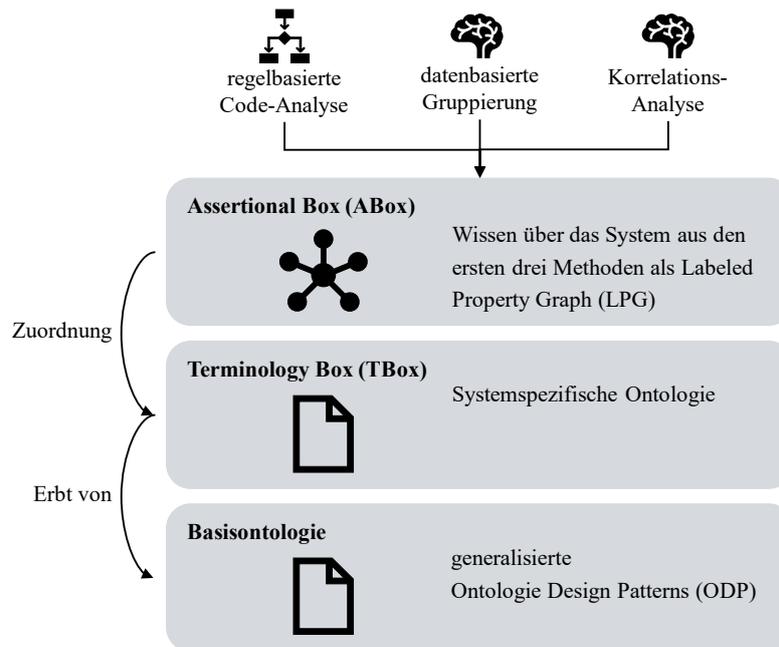


Abbildung 4.10: Hybride, hierarchische Wissensspeicherung

Als Basisontologie wurden zwei Ontologie Design Patterns (ODP) verwendet, die auf Wiederverwendbarkeit optimiert sind und so eine Anwendbarkeit auf viele bestehende Produktionssysteme erreichen sollen. Das erste ODP beinhaltet die Elemente zur Beschreibung der Struktur mechatronischer Systeme und basiert auf den VDI-Normen 2206 [49] und 3682 [169]. Dieses ODP wird für die semantische Beschreibung des Wissens über das System benötigt und ist in Abbildung 4.11 dargestellt. Die ODPs und TBox wurde im Tool Protegé entwickelt.

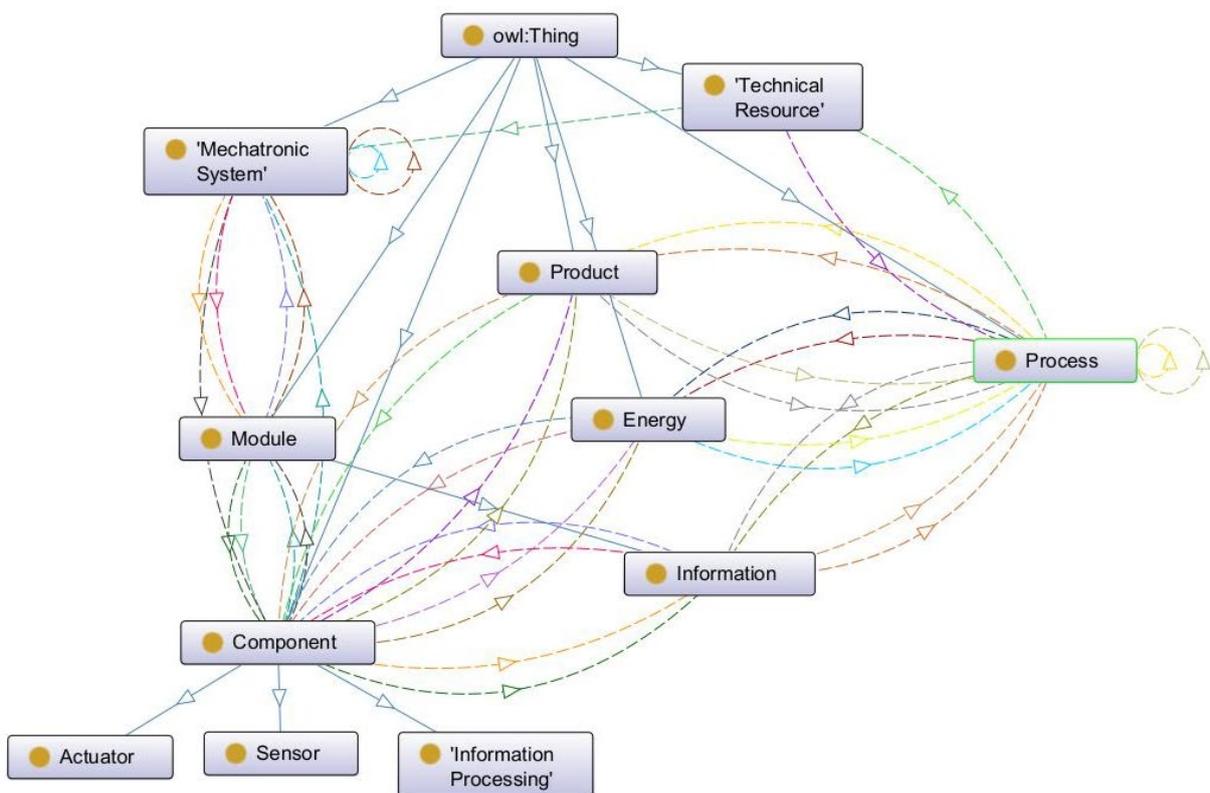


Abbildung 4.11: ODP mechatronischer Systeme basierend auf der VDI 2206/3682 in Protegé

Das zweite ODP basiert auf der ISO-Norm 23247 [170–173] zur Beschreibung Digitaler Zwillinge für Produktionssysteme und wird für die semantische Annotation des Wissens für die Übertragung in einen Digitalen Zwilling verwendet. Darauf aufbauend kann optional eine systemspezifische, detaillierte Ontologie definiert werden, welche die Elemente eines Systems und deren Eigenschaften präziser definiert und spezifische Abhängigkeiten ergänzt. Die Terminology Box (TBox) als systemspezifische Ontologie ermöglicht mehr Aussagen über das System und differenziertere maschinelle Verarbeitung des Wissens als die generische Basisontologie. Da diese Ontologie systemspezifisch ist, kann keine einheitliche Lösung vorgestellt, sondern muss auf die individuelle Ergänzung der TBox verwiesen werden. Derzeit werden (teil-) automatisierte Methoden zur Erstellung von Ontologien in mehreren Forschungen untersucht (z.B. in [174, 175]), die in Zukunft die Erstellung einer TBox erleichtern können. Aktuell ist davon auszugehen, dass die TBox manuell erstellt oder darauf verzichtet und direkt auf die Basis Ontologie aufgebaut wird.

Das extrahierte Wissen aus den drei Methoden wird dynamisch in einem Labeled Property Graph gespeichert, der effizient die große Anzahl an Elementen und den Beziehungen dazwischen verarbeiten kann. Dabei werden die extrahierten Elemente als Knoten dargestellt und deren Eigenschaften als Key-Value-Pairs (Property) den Knoten beziehungsweise den Beziehungen dazwischen zugeordnet. Dieser Wissensgraph bildet die Instanziierung der zugrundeliegenden Ontologien und wird auch als Assertional Box (ABox) bezeichnet. Die Labels der Knoten referenzieren dabei die Klassen der verwendeten Ontologien und reichern den erzeugten Wissensgraphen semantisch an. Die Labels werden zuerst in der TBox als spezifischere Ontologie gesucht. Wenn sie dort nicht enthalten sind oder die TBox nicht existiert, werden sie in zweiter Instanz in der Basis Ontologie gesucht und referenziert. Über eine zentrale Zuordnungstabelle können die vergebenen Labels durch die Analysemethoden angepasst werden.

4.6.4 Identifikation repetitiver Strukturen und Kennzeichnung als Template

Der Wissensgraph gespeichert als LPG ist das Resultat der bisher beschriebenen Methoden und besteht aus einer Vielzahl von Knoten und Kanten. Diese sind als Mensch nicht leicht zu interpretieren und zu überblicken. Bereits kleinere Systeme, wie zum Beispiel in 6.3.1 beschrieben, resultieren in einem großen Graphen wie in Abbildung 4.12 (linkes Bild) dargestellt. Der Wissensgraph als Ergebnis der Analysen kann nur schwer durch einen Ingenieur validiert, ergänzt oder angepasst werden. Daher muss das Wissen um eine Gruppierung angereichert werden, um die Knoten wiederholender Strukturen zusammenzufassen und dadurch die Größe des Wissensgraphs zu reduzieren [176]. Hierfür werden sogenannte Templates für wiederholt verbaute Komponentengruppen beziehungsweise deren Abbildung im Wissensgraphen erstellt. Die Templates entsprechen den Klassen aus dem Konzept der objektorientierten Programmierung, die in einer separaten Graphdatenbank abgelegt werden. Die Instanzen der Templates in der ABox

sind das Äquivalent der Klassenobjekte. Diese Komponentengruppen werden in Softwaretools zur Erstellung und Modellierung von Produktionssystemen ebenfalls als wiederverwendbare Engineering-Templates angelegt. Dadurch muss der Anwender nicht mehrfach identische Strukturen nachbauen, sondern kann diese mit wenigen Klicks automatisch erzeugen lassen. Dieses Vorgehen reduziert den zeitlichen Aufwand und die Fehlermöglichkeiten durch Flüchtigkeitsfehler beim mehrfachen Nachbau einer Struktur. Die Templates bieten außerdem eine Möglichkeit, gleichzeitig mehrere Templateinstanzen über die Anpassung eines Templates schnell zu modifizieren. Die Schwierigkeit für einen Menschen ist es, diese wiederholenden Strukturen zu erkennen und richtig abzugrenzen. Dieser Teil der Methodik grenzt über Graphalgorithmen automatisiert wiederholende Strukturen als Templates ab.

Wie in Abbildung 4.12 dargestellt, werden dazu repetitive Strukturen identifiziert und gekennzeichnet. Für diese Strukturen werden generische Templates angelegt, welche die Struktur abbilden und lediglich die Eigenschaften in den Key-Value Pairs der Knoten und Kanten individuell instanziiert werden müssen. Im Wissensgraphen wird die sich wiederholende Struktur durch einen Platzhalterknoten mit einem Verweis auf das Template und den Parametern für die Instanziierung ersetzt. Die Herausforderung in dieser Aufgabe besteht darin, dass die Klassenobjekte durch die vorgelagerten Analysen erstellt werden und das zugrundeliegende gemeinsame Template als Klasse identifiziert werden muss. Die Identifikation von Templates in großen Strukturen für die wiederholte Verwendung stellt für menschliche Experten aufgrund der Strukturgröße und der unsortierten visuellen Anordnung eine Herausforderung bei der Erstellung des Digitalen Zwillinges dar (vergleiche 6.3.3). Die Ersetzung vieler Knoten durch einen Knoten als Stellvertreter dieses Templates erleichtert es Ingenieuren die Struktur zu überblicken (vergleiche letzter Schritt in Abbildung 4.12). Des Weiteren erlauben Templates eine effiziente Modifikation des Graphen. Indem das Template verändert wird, verweisen alle Stellvertreter-Knoten auf die angepasste Struktur. Dadurch muss die Struktur im Graph nicht mehrfach angepasst werden, wenn eine Baugruppe ausgetauscht wird, sondern nur einmal im Template.

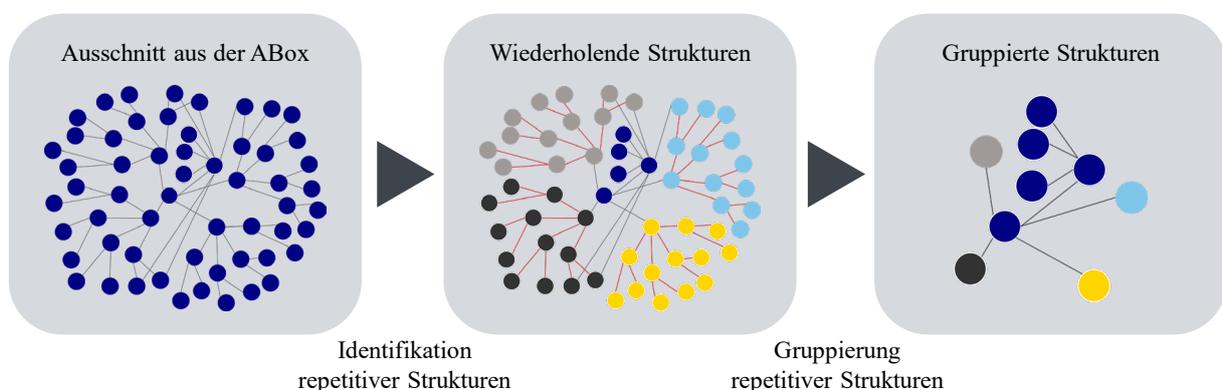


Abbildung 4.12: Identifikation repetitiver Strukturen und Kennzeichnung als Templates

Die Identifikation repetitiver Strukturen in Graphen ist auch in der Chemie beziehungsweise der Biologie eine wichtige Aufgabe, um häufig auftretende Molekülstrukturen zu identifizieren und die Medikamentenentwicklung zu unterstützen [177]. In diesem Rahmen werden in großen

Graphdatenbanken mittels verschiedenen Mining Algorithmen nach häufig wiederholt auftretenden Teilgraphen durchsucht. Aktuell gibt es noch keine Graphdatenbanken mit nativ implementierten Lösungen für die Identifikation häufiger Teilgraphen, sondern selbständige Implementierungen und erste Vorabversionen von integrierten Algorithmen. Wörlein et al. [177] haben verschiedene Algorithmen auf Datensätzen von Molekülen getestet und verglichen. Dabei sind sie zu dem Schluss gekommen, dass Algorithmen, die mit Graph-Embedding arbeiten, bei großen Teilgraphen schneller sind. Für kleine und mittlere Teilgraphen konnten sie keinen signifikanten Vorteil gegenüber Algorithmen ohne Embedding erkennen. Im Gegenteil benötigen diese Algorithmen weniger Rechenleistung, da keine Embeddings berechnet werden müssen. Der gSpan (graph-based Substructure pattern) Algorithmus [178] ist ein solcher Algorithmus, der kein Graph-Embedding verwendet und häufig in der Biologie beziehungsweise Chemie verwendet wird. Außerdem wird dieser Algorithmus auch bereits auf Graphen von industriellen Produktionssystemen angewendet, die durch Ontologien angereichert wurden [179].

Vergleichbar mit den obengenannten Algorithmen ist die Community Detektion, welche die Knoten in einem Graphen in eng verbundene Gruppen aufteilt. Diese Gruppen sind nicht direkt identisch mit den häufigen Teilgraphen aus den vorangehenden Algorithmen. Unter der Annahme, dass die Knoten innerhalb eines solchen Teilgraphen enger vernetzt sind als zu anderen Knoten entsprechen diese Gruppen den Instanzen der Templates. Um die Templates aus den Instanzen nachzubauen, müssen die Elemente in den Gruppen ohne deren IDs, deren individuelle Namen und Parameter verglichen werden. Wenn die verbleibende Struktur der Gruppen (Verbindungen und Knotentypen) identisch ist, handelt es sich um dasselbe Template. Die Gruppen können dann dem Template zugeordnet beziehungsweise die Templates daraus abgeleitet werden. Auf diesem Weg wird dasselbe Ergebnis, wie durch den gSpan Algorithmus erreicht. Dieser zweite Weg hat im Vergleich zum gSpan Algorithmus, zusätzlich zur kürzeren Laufzeit, den Vorteil, dass nicht bereits vorher die ungefähre Größe der gesuchten Teilgraphen bekannt sein muss.

Eine dritte Möglichkeit, diese Templates zu identifizieren, ist über eine feste Trennung des Graphen mittels apriori Wissen über die Template-Struktur. Dabei wird angenommen, dass jeder Knoten vom Typ einer funktionalen oder räumlichen Gruppe der Ursprung eines Templates sein kann. Um die Kohäsion des Wissensgraphen zu reduzieren und die Struktur aufzubrechen, wird die Verbindung dieser Knoten zum übergeordneten Systemknoten getrennt. Dadurch entstehen unverbundene Graph-Segmente, deren Struktur, wie im vorangehenden Abschnitt beschrieben, verglichen und daraus Templates abgeleitet werden können. Dieser Ansatz funktioniert jedoch nur dann, wenn die Systemstruktur flach ist und nur eine direkte Verbindung von Gruppen zum Systemknoten besteht. Da Produktionssysteme in einigen Fällen mehr als eine hierarchische Ebene besitzen und somit keine direkte Verbindung der Gruppenknoten zum Systemknoten vorhanden ist und weil oft mehr als diese eine Verbindung existiert, ist dieser Ansatz nur in beschränkten Fällen anwendbar.

Allgemein für alle oben genannten Algorithmen ist es eine Herausforderung, dass kein Baum mit einem Ursprung sowie klaren hierarchischen Ebenen ohne Querverbindungen zwischen den Ästen vorliegt, sondern ein Graph mit Querverbindungen auf vielen Ebenen. Aufgrund der zahlreichen möglichen Pfade und Kombinationen der Knoten ist es schwierig, alle Subgraphen zu identifizieren [180]. Um die Laufzeit der Algorithmen zu reduzieren, kann die Größe des Graphen und der Möglichkeiten reduziert werden, indem die Knoten und Kanten im Wissensgraph über die Steuerelektronik zur Anlagenautomatisierung für diesen Schritt ignoriert oder entfernt werden. Diese haben primär keinen Bezug zur Struktur des Produktionssystems und erzeugen viele Querverbindungen zwischen der baumartigen Struktur eines Produktionssystems, da sie die Sensoren und Aktoren in der SPS zusammenführen. Nach der Identifikation der Templates müssen diese Elemente wieder hinzugefügt oder die Gruppierung im vollständigen Wissensgraphen ergänzt werden.

Alle drei Varianten sind für die vorliegende Aufgabe nutzbar und können abhängig von den Gegebenheiten ausgewählt werden. Die Wahl der Graphdatenbank spielt dabei eine Rolle, da nicht für jede Implementierung jeder Algorithmus zur Verfügung steht beziehungsweise nur über den Umweg mit einer Datenkonvertierung und Export in ein anderes Format genutzt werden kann.

4.7 Überführung in einen Digitalen Zwilling

Das im Wissensgraph gespeicherte und durch die Schritte aus 4.6.3 und 4.6.4 angereicherte Wissen kann nun genutzt werden. Entweder wird das Wissen visualisiert und von einem Menschen verwendet, um die Zusammenhänge des Brownfield-Produktionssystems leichter und umfassender zu verstehen und manuell einen Digitalen Zwilling zu erstellen. Besser noch wird dieses Wissen automatisiert in die gängigen Softwaretools zur Anlagenplanung und -programmierung überführt und dadurch der zeitliche Aufwand reduziert sowie menschliche Fehler beim Übertrag vermieden.

Das Zielsystem sind Produktlebenszyklus Management (PLM)-Tools, welche durch beachtliche Investitionen und Entwicklungsaufwand vorangetrieben wurden und weiter wachsen [181]. Dabei sind viele Tools mit herstellerepezifisch implementierten Datenformaten involviert [181]. Die Speicherung und der Austausch zwischen diesen Tools hängen von der gesamten IT-Landschaft der verwendeten Software ab. Geschlossene IT-Systeme aus einer Hand (End-to-End Lösung) haben strukturierte, herstellerepezifisch optimierte Datenmodelle, über welche die Tools auf die gespeicherten Daten zugreifen können. In heterogenen IT-Systemen, bestehend aus Softwaretools verschiedener Hersteller (Best-of-Breed-Lösung), werden für den Austauschprozess entsprechende Formate, oft XML-basiert, eingesetzt. Siemens hat mit PLM XML ein derartiges Format definiert, das für den Datenaustausch vor allem zwischen Siemenstools genutzt wird.

AutomationML (AML) als herstellerunabhängiges, standardisiertes und XML-basiertes Format, das auf etablierten Formaten aus den Anlagenengineering Domänen aufbaut und diese integriert,

ist ein verbreitetes Austauschformat [182]. Dies ermöglicht eine breitere Anbindung von Softwaretools unterschiedlicher Hersteller im Vergleich zu PLM XML. Die meisten Entwicklungstools können inzwischen AML importieren und exportieren oder sind aufgrund der verwendeten Normen in AML leicht aufrüstbar, wie verschiedene Beispiele und Open Source Implementierungen zeigen [183]. Somit ermöglicht ein Export der Relationen aus dem Wissensgraphen als AML-Datei eine breite Verwendbarkeit. Des Weiteren ist auch eine Integration von AML in Systeme mit strukturierten Datenmodellen oder nativen Austauschformaten möglich [133].

Aus diesem Grund wird als letzter Baustein der Methodik ein Export des Wissens als AML-Datei verwendet (vergleiche Abbildung 4.1: Mapping). Für den Export stehen auch bereits erste Implementierungen zur Verfügung, die in der Realisierung verwendet und angepasst werden können [132]. Das Exportmodul arbeitet die Knoten und Kanten des Wissensgraph ab und übersetzt diese anhand eines Mappings in AML-Elemente. Dieser Baustein ermöglicht eine leichtere Evaluierung der Methodik und kann bei Bedarf durch jede beliebige Exportfunktion ausgetauscht werden.

In Kapitel 4 wurde die entwickelte Methodik vorgestellt, mit der die Relationen von Brownfield-Produktionssystemen automatisiert erzeugt werden können. Dabei wurden drei verschiedene Informationsquellen und deren Analysemethoden sowie die Zusammenführung, Speicherung und Anreicherung des daraus gewonnenen Wissens vorgestellt. Im folgenden Kapitel wird die prototypische Implementierung dieser Methode als Software-Assistenzsystem vorgestellt. Dieses ist eine Realisierung der Methodik, die eine automatisierte Ausführung erlaubt und für die Evaluierung herangezogen wird.

5 Realisierung der entwickelten Methodik als prototypisches Software-Assistenzsystem

In diesem Kapitel wird die prototypische Implementierung der im letzten Kapitel entwickelten Methodik als Software-Assistenzsystem beschrieben. Dieses führt automatisiert die einzelnen Analysen und Verarbeitungsschritte aus den Bausteinen der Methodik aus. Die hier beschriebene Implementierung erhebt keinen Anspruch auf eine optimale, kommerzielle Software, sondern dient der Demonstration der Methodik und wird für die Evaluierung herangezogen. Speziell die Import- und Exportmodule sind an die Gegebenheiten des bestehenden Produktionssystems individuell anzupassen. Auf diese Anpassungsoptionen und Alternativen wird im jeweiligen Absatz explizit verwiesen. Das Kapitel startet mit einem Gesamtüberblick über die Softwaremodule des Assistenzsystems, deren Schnittstellen und dem Datenfluss. Anschließend wird die Realisierung der einzelnen Softwaremodule detailliert erläutert.

5.1 Gesamtüberblick über die Softwaremodule

Dieses Kapitel bietet einen Überblick über die Realisierung der Methodik in Form eines Software-Assistenzsystems. Dabei wird das System in funktional abgeschlossene Module untergliedert. Abbildung 5.1 ist die Visualisierung der Übersicht über alle Module. Die blau eingefärbten Module in Abbildung 5.1 sind im Rahmen dieser Forschung implementiert worden, die Grauen stellen kommerziell verfügbare Software dar und werden unverändert vom Assistenzsystem über deren API eingebunden.

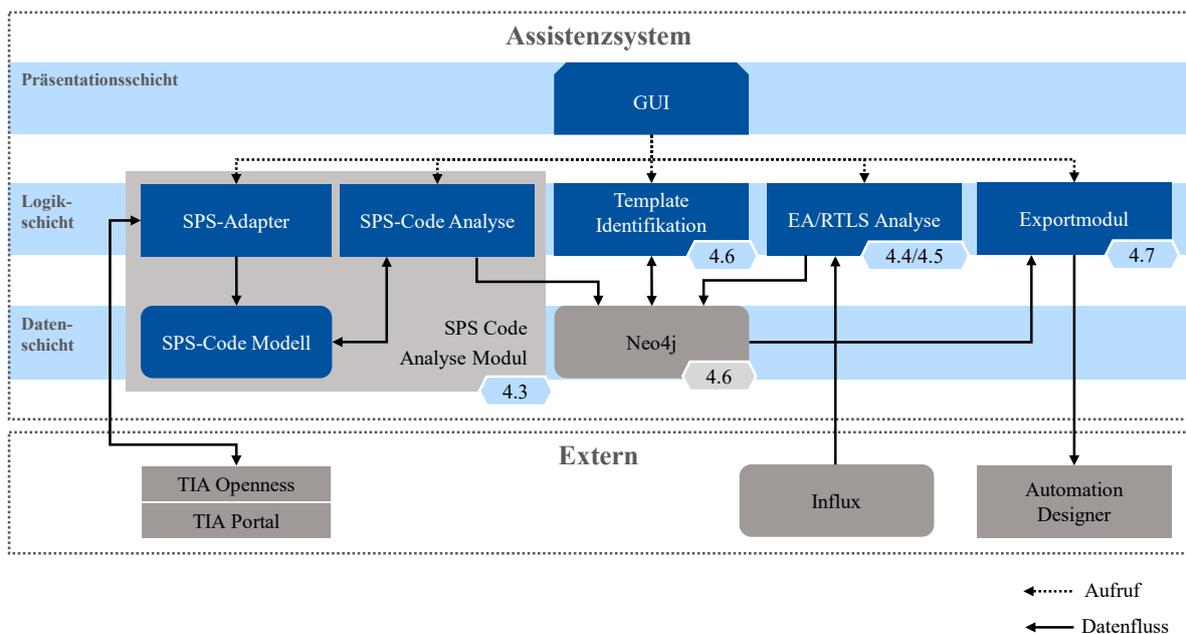


Abbildung 5.1: Überblick über die Softwaremodule des Assistenzsystems

Die Module des Assistenzsystems sind in die klassische drei Schichten Architektur eingeordnet, mit der Datenhaltungsschicht, der Logikschicht und der Präsentationsschicht. Außerdem wurde die Beziehung zu den Kapiteln der Methodik in Sechsecken hinzugefügt.

Das Assistenzsystem verfügt über eine GUI (siehe Abbildung 5.2), die als Element der Präsentationsschicht von den Benutzern zur Ansteuerung der einzelnen Funktionen und der Anzeige der Ergebnisse genutzt wird. Über verschiedene Buttons startet der Benutzer die Ausführung der anderen Softwaremodule der Logikschicht. Die Benutzer werden durch eine selektive Verwendbarkeit der Funktionen durch die korrekte Anwendung des Assistenzsystems geführt. Dabei sind anfangs nur die Module freigeschaltet, welche die Informationen einlesen und analysieren. In Abbildung 5.2 ist der Import und die Analyse des SPS-Codes bereits abgeschlossen und ausgegraut. Nach der Durchführung aller Analysen und dem Export der Ergebnisse in die Neo4j Graphdatenbank ist die Identifikation der Templates zur Anreicherung des Graphen verfügbar. Abschließend wird die Exportfunktionalität freigeschaltet und das Wissen kann exportiert werden. Die Ausführung der einzelnen Module ist bewusst schrittweise umgesetzt, sodass der Bediener die Zwischenergebnisse überprüfen kann. Es wird erwartet, dass durch die Transparenz der einzelnen Schritte der Methodik die Akzeptanz für die Methodik erhöht werden kann. Darüber hinaus bieten auch die Zwischenergebnisse einzelnen Benutzern bereits Mehrwerte. Beispielsweise kann ein SPS-Programmierer aus dem Ergebnis der SPS-Code-Analyse ein besseres Verständnis über die Abhängigkeiten in der Steuerungssoftware gewinnen. Ein vollautomatischer Betrieb des Assistenzsystems ist durch die Verkettung aller Schritte bei Bedarf nachrüstbar.

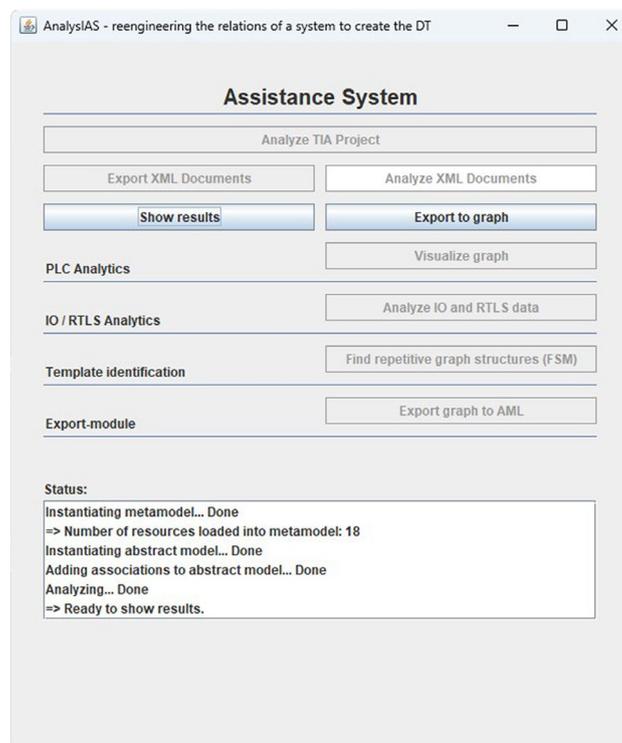


Abbildung 5.2: Graphische Benutzeroberfläche des Assistenzsystems

Die durch die GUI aufgerufenen Softwaremodule der Logikebene und Datenhaltungsebene lassen sich in vier Abschnitte einteilen. Wie in Abbildung 5.1 dargestellt, werden die Module mit Bezug zur SPS sowie mit Bezug zum Wissensgraphen in Neo4j jeweils als Abschnitt zusammengefasst.

Der erste Abschnitt setzt sich dabei aus den Modulen der SPS-Code-Analyse für das Einlesen, Zwischenspeichern und Analysieren der Informationen zusammen. Das Modul SPS-Adapter ist für das Einlesen verantwortlich und auf den herstellerspezifischen SPS-Code beziehungsweise dessen Schnittstelle abgestimmt. Im realisierten Assistenzsystem wurde der SPS-Code einer Siemens ET200sp analysiert und daher ein Adapter für das TIA-Portal, die Siemens Entwicklungsumgebung für SPS-Code, erstellt. Der SPS-Adapter ist nach dem Entwurfsmuster eines Adapters [184] aus der objektorientierten Programmierung aufgebaut und verbindet die herstellerspezifische Software für den Export des SPS-Codes als XML-Dateien mit dem Assistenzsystem. Wenn der SPS-Code eines anderen Herstellers verwendet werden soll, muss dieses Modul ausgetauscht oder ergänzt werden, um mit der API eines anderen Herstellers kompatibel zu sein. Das Resultat des SPS-Adapters ist das SPS-Code-Modell, welches die benötigten Informationen für die Analyse herstellerunabhängig als Java Klassen (vergleiche Abbildung 4.3) bereitstellt. Der eigentliche Analysevorgang verarbeitet den Inhalt des SPS-Code-Modells und speichert das extrahierte Wissen über die funktionale Gruppierung durch Betätigung des „*Export to Graph*“ Buttons in der Neo4j Graphdatenbank. Die beiden Module können entweder manuell einzeln (Abbildung 5.2: „*Export XML Documents*“, „*Analyze XML Documents*“) oder als Sequenz („*Analyze TIA Project*“) automatisch nacheinander ausgeführt werden. Dadurch ist es möglich, bei wiederholter Ausführung des Assistenzsystems oder wenn bereits XML-Dateien vorhanden sind, den im Vergleich zur Analyse zeitaufwendigen XML-Export zu überspringen.

Der zweite Abschnitt beinhaltet den Button für die datenbasierten Analysen der zweiten (4.4) und dritten (4.5) Methode. Diese sind in einem Softwaremodul zusammengefasst implementiert, welches die E/A und RTLS-Daten aus einer Influx Zeitreihendatenbank als gemeinsame Quelle bezieht. Das Softwaremodul extrahiert die physischen Zusammenhänge der mechatronischen Komponenten und speichert die Analyseergebnisse ebenfalls im Wissensgraphen in der Neo4J Graphdatenbank. Dieses Modul und die Module der SPS-Code-Analyse können durch den Benutzer in beliebiger Reihenfolge ausgeführt werden.

Anschließend an eine erfolgreiche Analyse aller Informationen und die Speicherung der Ergebnisse in der Neo4j Graphdatenbank sind die beiden verbleibenden Softwaremodule für die Template-Identifikation (dritter Abschnitt) und den Export (vierter Abschnitt) freigeschaltet. Die Ausführung des Softwaremoduls zur Template-Identifikation ist optional und kann, wenn lediglich die Struktur eines einzelnen Produktionssystems und keine Segmentierung für die Wiederverwendung benötigt wird, übersprungen werden. Die Verwendung beider Softwaremodule mit dem Ziel, dass der Export wiederverwendbare Templates beinhalten soll, setzt voraus, dass zuerst die Templates identifiziert werden und anschließend der Export

durchgeführt wird. Wird diese Reihenfolge vertauscht, erfolgt eine Warnung auf der grafischen Benutzeroberfläche, dass die Templates nicht im vorherigen Export enthalten sind und dieser erneut durchgeführt werden muss. Der Export wird in dieser Realisierung der Methodik als AutomationML Datei ausgeführt und anschließend händisch im Siemens Automation Designer importiert. Diese Software führt die Fachbereiche Mechanik, Elektrik und Software zusammen und wird in dieser Realisierung für die Evaluierung der Methodik herangezogen.

Die beschriebenen Softwaremodule sind in verschiedenen Submodulen umgesetzt und unter der Verwendung bestehender Softwarebibliotheken implementiert worden. In den folgenden Kapiteln werden die Softwaremodule des Assistenzsystems feiner untergliedert in die enthaltenen Submodule und detailliert betrachtet.

5.2 Softwaremodul für die Analyse des SPS-Codes

Dieses Kapitel beinhaltet die Beschreibung aller Submodule für die Analyse des SPS-Codes, die enthaltenen Funktionen und deren Implementierung. Die Aufgabe dieser Softwaremodule ist es, den herstellerepezifischen SPS-Code einzulesen, in eine übertragbare Struktur zu bringen und über die Analyse daraus Aussagen über die funktionalen Zusammenhänge zu extrahieren. Die Module sind die Implementierung der in Kapitel 4.3 beschriebenen Methode.

Zur Entkopplung der Analyse von der herstellerepezifischen Informationsbereitstellung ist ein spezialisierter Adapter implementiert, der den SPS-Code in das SPS-Code-Modell importiert. Das SPS-Code-Modell wird durch das anschließende Analysemodul erst angereichert und dann analysiert, um die funktionalen Gruppen zu extrahieren. Alle Module, abgesehen von der Export Applikation, sind in Java geschrieben, weil für die Datenmodellierung und den Import in das Modell ein geeignetes Framework in Java verfügbar ist. Der genaue Aufbau der beiden Module wird im Folgenden erläutert.

5.2.1 Adapter zum Import des nativen SPS-Codes

Dieses Softwaremodul wird als SPS-Adapter bezeichnet, da es in mehrfacher Hinsicht die Aufgaben des gleichnamigen Entwurfsmusters der objektorientierten Programmierung erfüllt. Ein solcher Adapter, auch Wrapper-Klassen genannt, hat die Aufgabe, inkompatible Schnittstellen zu verbinden und zwischen diesen zu übersetzen. Der SPS-Adapter verbindet den nativen SPS-Code mit dem SPS-Code-Modell und sorgt für eine Übersetzung des herstellerepezifischen Codes über ein XML-Austauschformat in die Klassenstruktur des Modells für die Analyse. Außerdem verbindet er das in Java implementierte Assistenzsystem mit der nativen Software zum Export des SPS-Codes als XML, die in einer beliebigen anderen Programmiersprache geschrieben sein kann.

Der Adapter ist mehrstufig implementiert und übersetzt in erster Instanz den nativen SPS-Code in XML-Dateien. Die Konvertierung in XML-Dateien erfolgt dabei über die vom Hersteller

bereitgestellte Software. In dieser Forschung wird in Siemens TIA Portal implementierter SPS-Code verarbeitet und deshalb die TIA Portal API für den Export als XML-Dateien verwendet. TIA Openness stellt die Schnittstelle des TIA-Portals jedoch als API in C# bereit, welche nicht direkt in die Java Implementierung des Assistenzsystems eingebunden werden kann. Daher wurde eine Export-App in C# implementiert, welche die benötigten Exportfunktionen aus TIA Openness aufruft und den Export in ein spezifiziertes Austauschverzeichnis als XML-Dateien startet. Die Automatisierungshardware wird dabei nicht als reine XML-Dateien, sondern im XML-basierten AML-Format abgespeichert. Der SPS-Adapter abstrahiert die Exportfunktion für das Java-basierte Assistenzsystem, startet die Export-App über dessen Schnittstelle und wartet auf die erfolgreiche Beendigung des Prozesses. Sobald der Prozess der Export-App ohne Fehlermeldung terminiert, beginnt der SPS-Adapter die XML-Dateien im spezifizierten Austauschverzeichnis auf das korrekte Format und einen aktuellen Zeitstempel zu überprüfen. Wenn keine Fehler festzustellen sind, können die XML-Dateien in die Java-Umgebung des Assistenzsystems eingelesen werden. Hierzu wird das Eclipse Modeling Framework (EMF) verwendet, das die XML-Dateien als Java Objekte importiert.

Erstellung des Metamodells zum Entwicklungszeitpunkt

Als Vorarbeit wurde zur Entwicklungszeit das zugrunde liegende Metamodell erstellt, das aus den Java Klassen besteht und durch das EMF als Objekte instanziiert wird. Das Metamodell selbst ist eine Beschreibung aller möglichen Elemente, die durch die XML-Dateien abgebildet und als Objektinstanzen erstellt werden können. Es beschreibt die Struktur und Inhalte des daraus generierten Instanzmodells. Für die Erstellung des Metamodells wird eine XML-Schemadefinition (XSD) benötigt, welche den Aufbau und den Inhalt der zugehörigen XML-Dateien beschreibt. Diese dienen dem EMF als Grundlage zur automatischen Erzeugung des Metamodells. Die XSD-Datei kann dabei aus verschiedenen Quellen entstammen, wie in Kapitel 4.3.1 beschrieben. In dieser Forschung standen weder vollständige, herstellerepezifische XSD-Dateien zur Verfügung noch ist eine Umsetzung der Norm 61131-10 zu PLCopen XML als XSD umfangstechnisch realisierbar. Daher wird die Variante zur Generierung der XSD aus bestehenden XML-Dateien verwendet. Mit der Software „Liquid XML Studio“ wurden aus TIA-Projekten von bestehenden SPS-gesteuerten Produktionssystemen am IAS automatisiert eine XSD erzeugt (vergleiche hellblaue Elemente in Abbildung 5.3). Dabei wurden TIA-Projekte der Version 14 und 15 verwendet, sodass die XSD und das daraus resultierende Metamodell zukünftig für diese SPS-Code Varianten genutzt werden kann. Die Anwendbarkeit der XSD auf andere XML-Dateien von SPS-Codes kann durch das Liquid XML Studio oder andere Tools wie OxygenML überprüft werden. Gegebenenfalls muss die XSD erweitert oder aus den neuen XML-Dateien mit einem der Tools neu erzeugt werden. Mittels dem EMF wurde aus dieser XSD automatisiert das Metamodell bestehend aus 114 Java Klassen erzeugt, das im Anhang 1 als UML-Diagramm dargestellt ist.

Diese Schritte wurden für SPS-Codes des Herstellers Siemens in den Versionen 14 und 15 durchgeführt und müssen für weitere Analysen anderer SPS-Codes mit diesen Versionen nicht mehr durchlaufen werden. Um SPS-Code anderer Versionen oder Hersteller analysieren zu können, müssen diese Schritte erneut durchgeführt und ein angepasstes oder erweitertes Metamodell erstellt werden. Dieses Metamodell wird vom SPS-Adapter für den Import des zu analysierenden SPS-Codes benötigt.

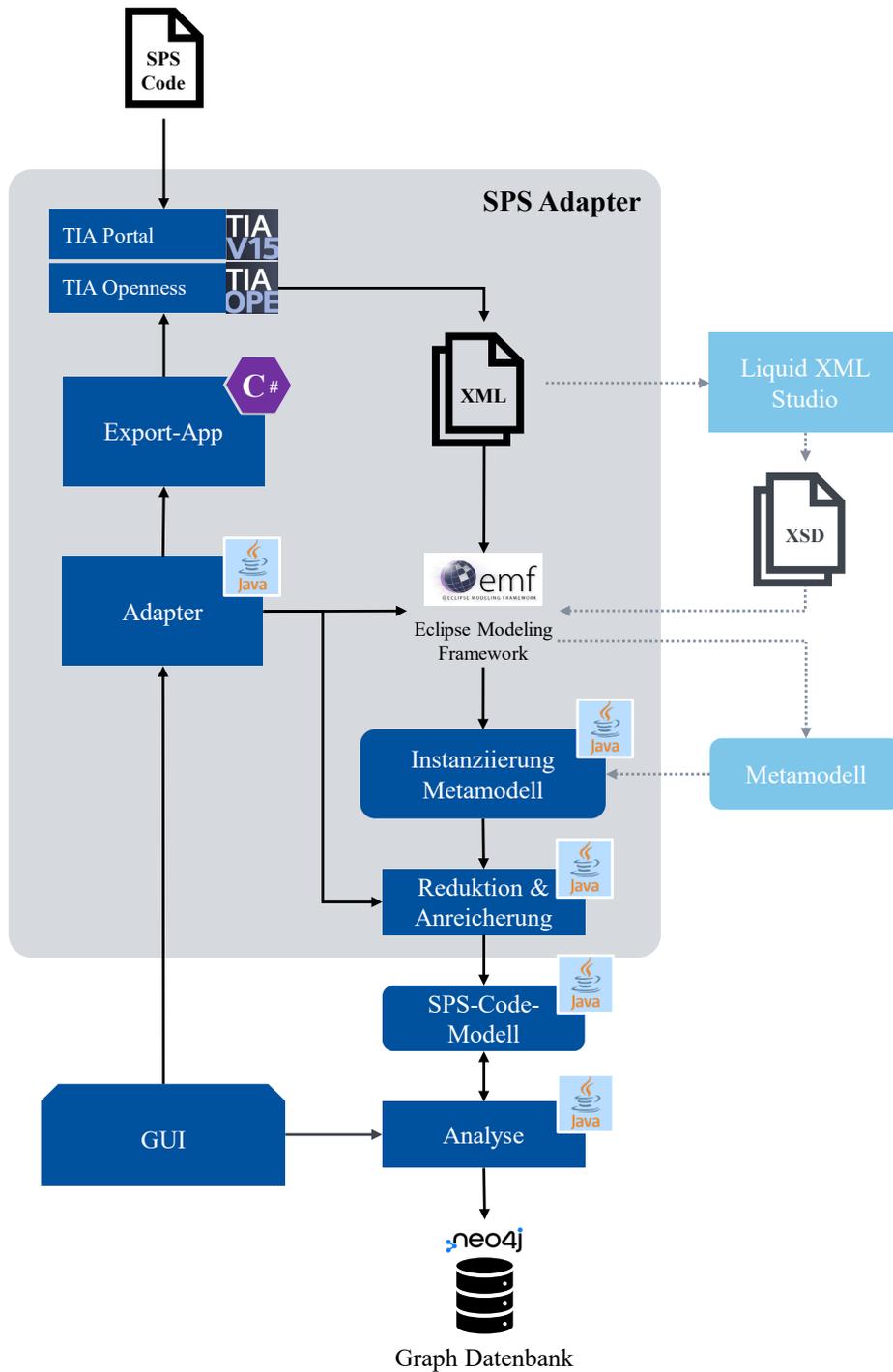


Abbildung 5.3: Submodule der SPS-Code-Analyse (Dunkelblau) und der einmaligen Entwicklungsschritte (Hellblau)

Ablauf des SPS-Code Imports mit dem SPS-Adapter und dem erstellten Metamodell

Der Import des SPS-Codes läuft basierend auf diesem Metamodell und der beschriebenen Export-App, wie in Abbildung 5.3 dargestellt, ab. Die Export-App übersetzt den SPS-Code des bestehenden Produktionssystems, das mit der Methodik als Digitaler Zwilling abgebildet werden soll, in XML-Dateien. Anschließend ruft der SPS-Adapter das EMF auf, das anhand der XSD für die XML-Dateien Instanzen des Metamodells erzeugt. Die dadurch entstehende Instanziierung des Metamodells beinhaltet Elemente, welche die Struktur und die Informationen der XML-Dateien beschreiben sowie generische Bezeichner basierend auf dem XML-Format. Um eine aussagekräftige Abbildung des SPS-Codes als Modell zu erreichen, wird die Instanz des Metamodells um Informationen über die XML-Datei bereinigt (zum Beispiel: die Kodierung der Zeichen oder die IDs der XML-Tags) und anschließend die baumartige Struktur der XML-Datei um Querverbindungen zu einem Netzwerk ergänzt. Implementiert wurde dies als Funktion des SPS-Adapters.

Dieser überträgt anhand einer Lookup-Tabelle nur die benötigten Informationen über den SPS-Code in das SPS-Code-Modell und iteriert anschließend über das Modell, um die Elemente mit Querverbindungen zu assoziieren, wie zum Beispiel Step über Connection mit Transition (vergleiche Abbildung 5.4). Die Assoziation erzeugt keine neuen Informationen, sondern verarbeitet die vorhandenen IDs aus der XML-Datei und fügt Java Objektzeiger in die beiden verbundenen Objekte ein. Beispielsweise wird der Inhalt eines Funktionsbausteins, programmiert in S7-Graph, als Step, Transition und Branch Objekte abgebildet. Diese werden über Objekte der Klasse Connection verbunden, wie im folgenden Beispiel in Abbildung 5.4 dargestellt. Diese Objekte beinhalten anfangs nur die aus der XML stammenden IDs in dem Connection Objekt. Für die Analyse eines Funktionsbausteines müssen nun für jeden Step alle Connections durchsucht werden, bis dessen ID gefunden wurde. Anschließend wird die ID der verbundenen Transition verwendet, um alle Transitionen nach dieser ID zu durchsuchen. Daher sind immer, wenn eine solche Verbindung analysiert wird, zwei sukzessive Suchen notwendig. Der Aufwand dieser Suchen und damit auch die Dauer der Analyse steigt mit zunehmender Größe eines SPS-Codes überproportional an. Um dies zu verhindern, werden bei der Assoziationsergänzung die in Abbildung 5.4 hellblau angedeuteten Beziehungen als Objektzeiger in die Objekte eingefügt.

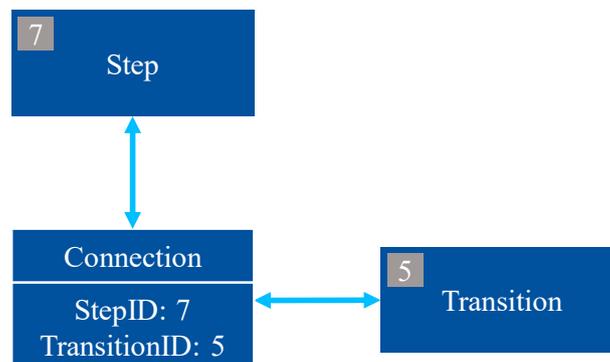


Abbildung 5.4: Beispiel für die Assoziationsergänzung (hellblau)

Abgesehen von dieser Konstellation mit drei Objekten, gibt es auch Anordnungen mit zwei involvierten Objekten (beispielsweise Tag mit Channel). In dieser Konstellation verfügt lediglich eine Seite über die Verbindungsinformation (vergleichbar mit Step und Connection). Auch diese Verbindungen resultieren in Suchaufwand und werden durch Zeiger bidirektional verknüpft. Diese expliziten Verbindungsinformationen im SPS-Code-Modell müssen einmalig durch die eben beschriebenen Suchvorgänge ergänzt, jedoch nicht bei jeder Verwendung während der Analyse neu durchgeführt werden.

5.2.2 Implementierung der regelbasierten Strukturanalyse

Die Analyse setzt auf dem SPS-Code-Modell nach Abbildung 4.3 auf und ist daher herstellerunabhängig implementiert. Das SPS-Code-Modell wird für die Analyse um sechs Klassen erweitert, welche einheitliche Regeln für die vergleichbaren Bausteine in verschiedenen SPS-Programmiersprachen definieren. Abbildung 5.5 bildet diese sechs zusätzlichen Klassen inklusive der Beziehung zu den Klassen des SPS-Code-Modells ab. Die Klassen AnalyzedBlock, AnalyzedCall, AnalyzedTag und AnalyzedNetwork beinhalten Verweise auf Instanzen der Klassen des SPS-Code-Modells, welche in Abbildung 5.5 am Rand in Blau ergänzend dargestellt sind. Diese Klassen dienen als Zwischenebene zum SPS-Code-Modell, sie definieren Analyseabläufe auf den Instanzen des SPS-Code-Modells und berechnen akkumulierte Informationen. Die Klassen AnalyzedBlock und AnalyzedNetwork verallgemeinern außerdem Instanzen verschiedener Programmiersprachen, indem sie, je nach zugrunde liegender Klasse, entsprechend unterschiedliche Methoden aufrufen. Der Hauptbaustein (MainOB / OB1) wird aus den zu analysierenden Blöcken ausgeklammert, da dieser zwangsläufig als Startpunkt des Steuerprogramms viele der anderen Blöcke aufruft und in der Analyse aller Blöcke eingeschlossen wird. Dies verfälscht die eigentliche Bindungsanalyse zwischen den funktionalen Blöcken. Die Klasse Component berechnet die Bindungsstärke zwischen den einzelnen verbleibenden Blöcken nach den in 4.3.2 definierten Regeln, speichert die berechneten Bewertungen und bildet die Grundlage für die abschließende Abgrenzung der SPS-Code Bausteine als funktionale Gruppen. Für die Berechnung der Bewertung werden, ausgehend vom analysierten Block, die darunterliegenden analysierten Komponenten (Blöcke, Netzwerke, Tags und Calls) aufgerufen.

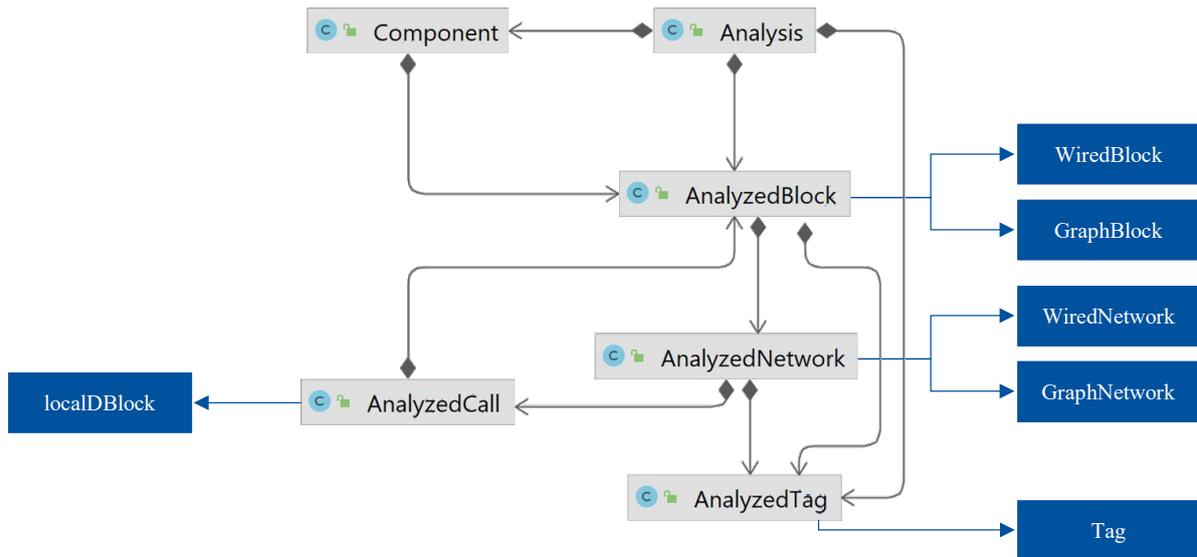


Abbildung 5.5: UML-Diagramm der Klassen mit Bezug zur Analyse des SPS-Code-Modells

Die Analyseergebnisse sind in Objekten der Klasse Component gespeichert und werden über die GUI visuell aufbereitet, sodass ein Benutzer diese Werte interpretieren und verwerten kann. Ein exemplarisches Analyseergebnis ist in Abbildung 5.6 dargestellt. Die Abgrenzbarkeit der Funktionsbausteine ist in Klammern nach den analysierten Blöcken (blaue Schrift) angegeben. Die Beziehungen zu den anderen Blöcken sind in der Box darunter aufgelistet, wobei die Beziehung über die Farbe qualitativ bewertet ist (grün=enge, gelb=leichte, grau=keine Bindung) und in Klammer die gemeinsamen Tags als qualitative Bewertung aufgelistet sind. Die qualitative Einordnung der Blöcke entspricht der finalen Einteilung der Blöcke in funktionale Gruppen und wird für die Generierung der Struktur des Digitalen Zwillings verwendet. Darüber hinaus können Ingenieure die Ergebnisse verwenden, um die Aufrufreihenfolge (Call Hierarchy), die Tagverwendung sowie die Bindung zwischen den Blöcken des SPS-Codes leichter zu verstehen, indem der Ablauf und die Zusammenhänge innerhalb des SPS-Codes visuell aufbereitet werden. Dies ermöglicht einem Ingenieur den SPS-Code stückweise zu betrachten und über die Aufteilung die Komplexität des Codes zu reduzieren. Außerdem kann über den Reiter *Expandable Model*, das SPS-Code-Modell betrachtet werden.

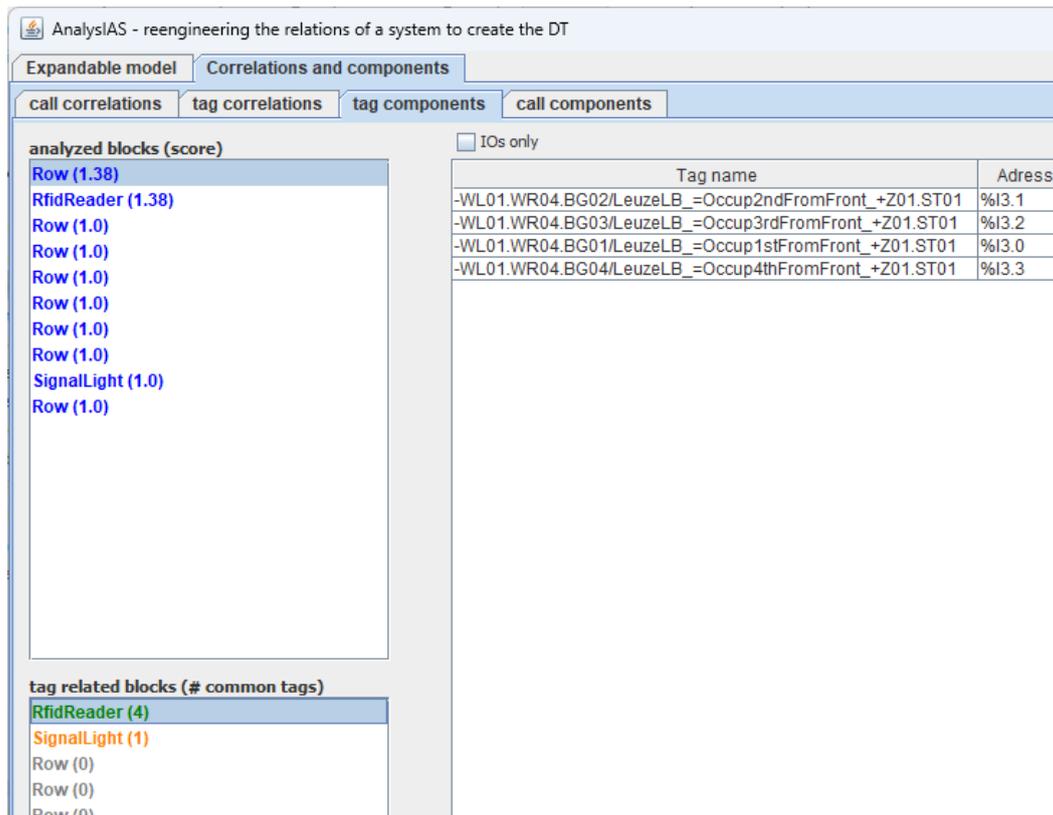


Abbildung 5.6: Visualisierung der Ergebnisse der SPS-Code-Analyse

In dieser Forschung werden die Vorteile für einen Menschen nicht weiter betrachtet und das gewonnene Wissen über die Struktur des Produktionssystems für die Erstellung der funktionalen Struktur des Digitalen Zwillings genutzt. Diese Erkenntnisse über die funktionale Struktur werden in den weiteren Modulen durch andere Perspektiven ergänzt und noch weiter strukturiert.

5.3 Analysemodul für Positions- und E/A-Daten

Das Softwaremodul zur Analyse der Positions- und Betriebsdaten ist die Realisierung der zweiten und dritten Methode. Diese Methoden greifen eng ineinander und werden für die Extraktion der physischen Gruppierung beide benötigt. Die beiden Methoden sind deshalb in einem gemeinsamen Modul in Python implementiert. Ausgangspunkt des Moduls ist der Zugriff auf eine Zeitreihendatenbank, deren Inhalt und Aufgabe im Folgenden beschrieben werden, gefolgt von der Implementierung der Analyse.

5.3.1 Zeitreihendatenbank zur Speicherung der Daten

Die Bereitstellung der Betriebs- und Positionsdaten ist mit einer frei verfügbaren Influx-Datenbank als Zeitreihendatenbank implementiert. Diese ist auf die Bedürfnisse abgestimmt, schnell einlaufende Datenströme, die über einen Zeitstempel indiziert werden sollen, optimiert abzuspeichern. Dabei können außerdem komplexe zusammengesetzte Indizes aus dem Zeitstempel und einem oder mehreren sogenannten Tags erstellt werden. Die zu speichernden

Daten werden als Fields bezeichnet und können ebenfalls aus mehreren Bestandteilen zusammengesetzt sein. Die Betriebsdaten besitzen einen zusammengesetzten Index bestehend aus dem Zeitstempel und dem Signalnamen sowie einem einfachen Field mit dem Signalwert. Die gespeicherten Positionen haben einen Index aus dem Zeitstempel und der RTLS Transponder ID sowie zusammengesetzte Fields mit den Positionswerten in X, Y und Z Richtung. Einigen der Positionsdaten ist darüber hinaus ein weiter Tag im Index für das Klassenlabel angefügt. Diese Labels sind für das Training des Klassifikationsmodells relevant.

Die Erfassung der Daten ist kein Kernelement der Methodik, spielt jedoch eine Rolle bei der Anwendung der Methodik und der Dauer der Ausführung, wenn für das Brownfield-Produktionssystem noch keine entsprechenden Aufzeichnungen vorliegen. Daher wurden im Rahmen dieser Forschung die Erfassungsmöglichkeiten für das Bestandssystem zur Evaluierung (siehe Kapitel 6.3.1) betrachtet und möglichst generische Erfassungssoftware verwendet beziehungsweise implementiert. Die Erfassung der Positionsdaten ist als Client implementiert, der die Positionen des RTLS über das standardisierte SLMP-Protokoll (ISO/IEC 24730-1) [185] erfasst und an die Influx-Zeitreihendatenbank übermittelt. Dabei kann zwischen einem automatischen Erfassungsmodus und einem halbautomatischen Labeling-Modus gewählt werden. Der Erfassungsmodus nimmt dabei eine Bereinigung der Positionsdaten vor, falls dies nicht direkt durch das RTLS durchgeführt wird. Dabei können unter anderem zeitlich aufeinander folgende identische Positionen zusammengeführt, der Positionsverlauf geglättet und nicht benötigte Informationen (z.B. der Akkustatus) entfernt werden. Im Labeling-Modus fügt das Erfassungsprogramm den erfassten Positionen zusätzlich Klassenlabels für das Training des Klassifikators als Tag in der InfluxDB an. Dazu muss im Programm die ID eines Positionstransponders und ein Label eingegeben werden. Dieses Label wird den erfassten Positionen des gewählten Transponders angefügt, bis das Labeling deaktiviert oder ein neues Label eingegeben wird. Die Erfassung der Betriebsdaten erfolgt über den OPC UA Server der SPS. Der OPC Standard als verbreiteter Industriestandard mit zusätzlicher Datenbeschreibung kann durch open-source Softwarepakete in eine InfluxDB geschrieben werden [186]. Die weiter steigende Verbreitung von OPC UA und die Forschung zur semantischen Anreicherung [175] führt dazu, dass dieser Erfassungsweg von Betriebsdaten sich weiter etabliert und als Basis dieser Analyse durchgängig zur Verfügung steht. Die Analyse benötigt die Daten aus beiden Erfassungswegen zum selben Zeitraum, weshalb in dieser Realisierung dieses Tool händisch gestartet und parallel zur Positionserfassung ausgeführt wird. In einer kommerziellen Implementierung kann das Tool zur Betriebsdatenerfassung nachgebaut und direkt in die Positionserfassung integriert werden, sodass beide Tools immer parallel laufen. Alternativ besteht bereits eine Erfassung dieser Daten und es muss eine Extraktion sowie Transformation der Daten stattfinden, sodass sie für die Analysen aufbereitet vorliegen.

5.3.2 Implementierung der datengetriebenen Analysen

Die Analyse selbst ist in Python 3.7.6 implementiert, da in Python alle Bibliotheken für die Datenaufbereitung, Analyse und Klassifikation vorhanden sind. Des Weiteren ist jeweils eine Bibliothek mit einem Treiber für die Anbindung der vorgelagerten Influx-Datenbank und der nachgelagerten Neo4j-Datenbank verfügbar.

Bevor die eigentliche Analysephase beginnt, wird geprüft, ob ein aktueller Klassifikator für die Positionen vorliegt. Wenn kein oder kein aktuelles Klassifikationsmodell vorliegt, werden alle Positionen mit Labels aus der InfluxDB abgefragt und in ein Pandas Dataframe eingelesen. Diese Daten werden als Beispieldatensatz für den k-nearest Neighbor Algorithmus mit dynamic time-warping (DTW) als Metrik der Sequentia Bibliothek geladen und als Klassifikator gespeichert. Dieser Klassifikator kann beliebig ausgetauscht werden, wenn andere Anforderungen vorliegen oder gegen einen Clustering-Algorithmus ersetzt werden, falls keine gelabelten Daten vorliegen. Dabei ist zu beachten, dass ein Clustering in einer weniger zutreffenden Gruppierung hinsichtlich der physischen Zusammengehörigkeit resultieren kann.

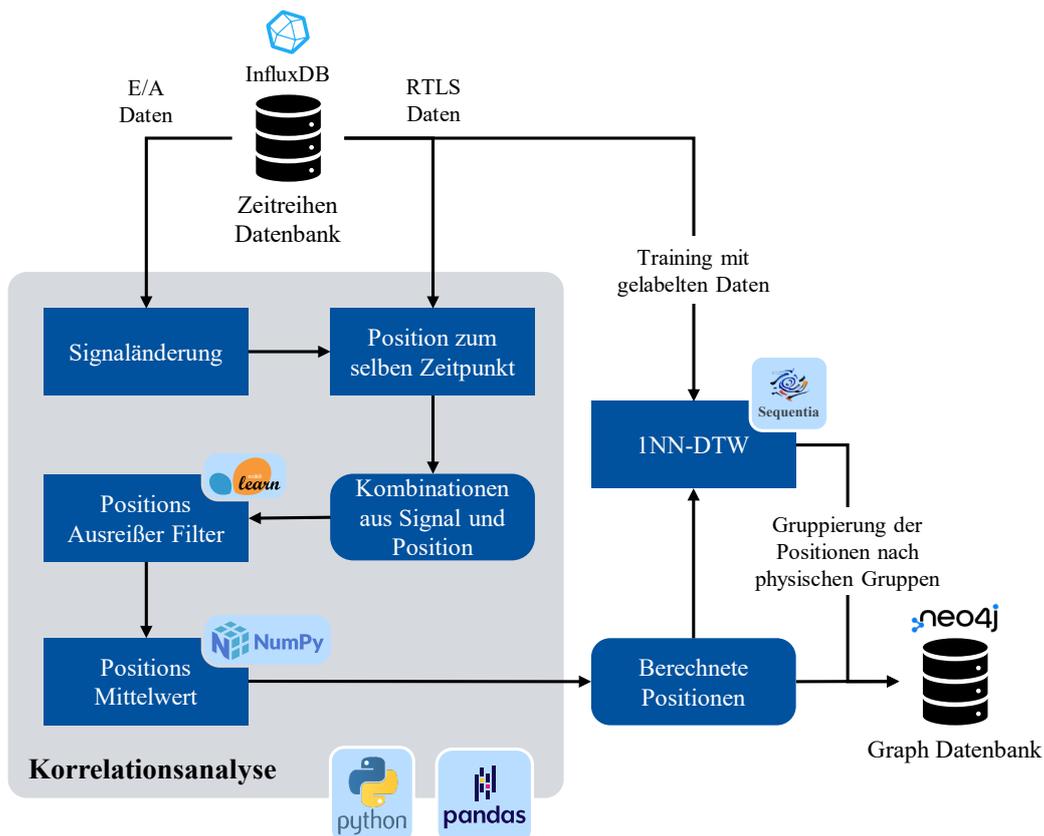


Abbildung 5.7: Submodule des EA/RTLS-Analyse-Moduls

Wenn ein Modell zur Einordnung der Positionen in physische Gruppen vorhanden ist, beginnt die eigentliche Korrelationsanalyse mit dem Einlesen der benötigten Informationen aus den beiden Datenbanken (Positionen/Betriebsdaten) der InfluxDB (siehe Abbildung 5.7). In diesen wird zuerst in den Sensor-/Aktorsignalen nach Änderungen des Signals zwischen zwei Einträgen gesucht. Diese Signalwechsel werden, sortiert nach dem zugehörigen Zeitstempel, gesammelt und

nachfolgend für diese Zeitpunkte nach Einträgen in den Positionen gesucht. Diese Informationen werden basierend auf deren Zeitstempel zusammengeführt und anschließend nach den Signalnamen sortiert, die als Tag den Betriebsdaten in der InfluxDB anhängt. Die gesammelten Einträge werden pro Signal mittels Funktionen zur Detektion von Ausreißern aus der Bibliothek „Scikit-learn“ und Funktionen zur Mittelwertberechnung aus der Numpy Bibliothek verarbeitet und zu einer Position pro Signal reduziert. Die berechnete Position eines Signals respektive der zugehörigen Peripherie wird im Nachgang durch den trainierten Klassifikator einer physischen Gruppe zugeordnet.

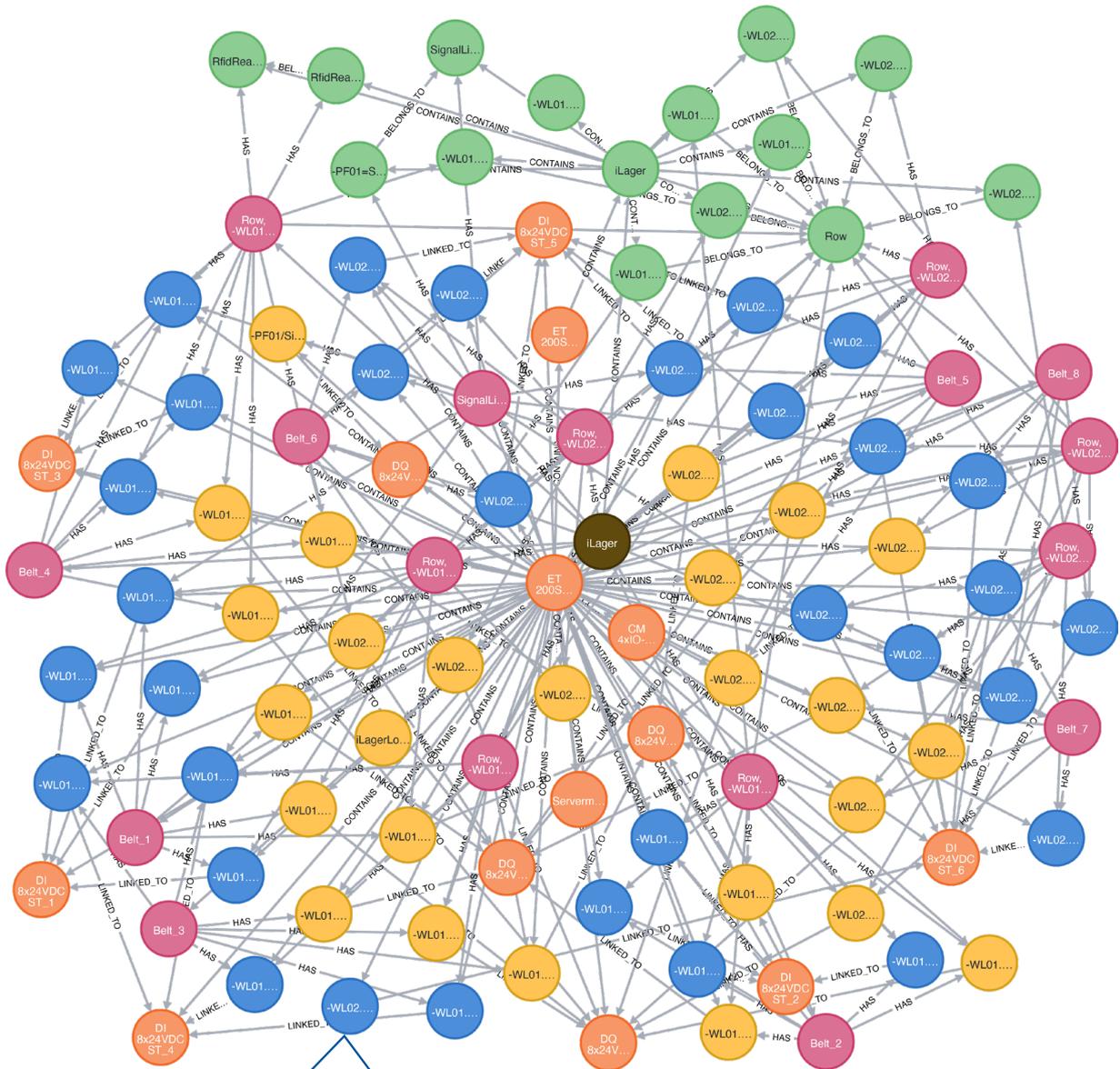
Die Informationen zur Position der Peripherie und deren Zugehörigkeit zu einer physischen Gruppe werden dem Wissensgraphen in der Neo4j-Graphdatenbank hinzugefügt. Das Softwaremodul erstellt dabei für jedes Peripheriegerät einen Knoten und führt diesen anhand des Signalnamens mit den bestehenden Knoten aus der SPS-Code-Analyse zusammen. Als Resultat beinhaltet der Graph in Neo4j die Knoten aus der SPS-Code-Analyse, die um die Eigenschaften zur Position ergänzt werden. Die physischen Gruppen werden als neue Knoten angelegt und alle Peripherie-Knoten durch eine Beziehung einem solchen Knoten zugeordnet. Dabei wird jede Peripherie nur einer physischen Gruppe zugeordnet, da physisch eine Peripherie auch nur an einem Ort verbaut sein kann.

5.4 Modul zur Wissensspeicherung und Extraktion in einem Graphen

Das extrahierte Wissen über die Relationen aus den Analysemodulen wird in einem Wissensgraphen gespeichert und zusammengeführt. Zu diesem Zweck sind im Assistenzsystem zwei Module implementiert. Das erste Modul ist die Graphdatenbank Neo4j, welche die in den Analysen generierten Knoten und Kanten speichert (Kapitel 5.4.1). Das zweite Modul implementiert die Extraktion von zusätzlichem, bereits implizit enthaltenem Wissen und wird in Kapitel 5.4.2 ausführlich erläutert.

5.4.1 Wissensspeicherung in einer Graphdatenbank

Das generierte Wissen wird in einer Neo4j Graphdatenbank gespeichert. Diese kann als separate Instanz der freiverfügbaren „Neo4j Desktop“-Anwendung ausgeführt werden. Falls das Assistenzsystem keine laufende Neo4j Graphdatenbank findet, hostet es automatisch eine Build-In-Instanz von Neo4j. Die Wahl von Neo4j als Datenbank bietet, abgesehen von der effizienten Speicherung und Abfrage von Informationen mit komplexen Beziehungen, nativ implementierte Algorithmen für Graphenanalytics. Diese Algorithmen sind für das Modul zur Wissensextraktion relevant. Darüber hinaus sind Datenbanktreiber als Bibliotheken für die beiden verwendeten Programmiersprachen Java und Python verfügbar, sodass keine zusätzliche Konvertierung vorgenommen werden muss.



Node properties

Sensor

dataType	Bool
fromFileName	PLC_Tags
id	16
isType	Sensor
logicalAddress	%5.0
name	-
	WL02.WR03.BG01/LeuzeLB_=Occup1
	stFromFront_+Z01.ST01
timesRWNA	5
timesRead	0
timesWrite	0

Legende

Node labels

- * (106) **System (1)** **Hardware (1)** **CpuDevice (1)**
- PeripheryDevice (12)** **PlcCode (1)** **FunctionBlock (3)**
- DataBlock (11)** **Sensor (33)** **Actuator (26)**
- FunctionGroup (9)** **LocationGroup (8)**

Relationship types

- * (315) **HAS (158)** **CONTAINS (89)**
- BELONGS_TO (10)** **LINKED_TO (58)**

Abbildung 5.8: Wissensgraph für das intelligente Lager in Neo4j Desktop

Über die Neo4j Desktop Anwendung kann außerdem für eine Verifikation durch einen Ingenieur der Graph visualisiert werden. Für das intelligente Lager, das in der Evaluierung verwendet wird (vgl. Kapitel 6.3.1), sieht der Wissensgraph wie in Abbildung 5.8 aus. Neo4j generiert automatisch eine Farbcodierung der Knoten anhand des Knotentyps, um den Graphen übersichtlicher zu gestalten. Bereits für überschaubare Systeme, wie das intelligente Lager, führt die Methodik zu einem Graphen mit 108 Knoten und über 300 Kanten. Trotz der Farbcodierung ist die Visualisierung der Knoten und Kanten unübersichtlich. Für den finalen, automatisierten Export des Graphen als grundlegende Struktur für den Digitalen Zwilling ist die unübersichtliche Visualisierung primär unproblematisch. Um einem Menschen die weitere Arbeit mit diesem Wissensgraphen zu ermöglichen, ist eine Segmentierung notwendig. Auch zur Kennzeichnung von Templates beziehungsweise der Klassen-Instanz-Relationen ist noch ein Zwischenschritt notwendig, der im folgenden Kapitel beschrieben wird.

5.4.2 Generierung von Templates durch die Identifikation von repetitiven Strukturen

Dieses Modul extrahiert aus dem Wissensgraphen implizit enthaltenes Wissen über sich wiederholende Strukturen im Wissensgraph. Diese Strukturen entstehen durch mehrfach verbaute, identische oder ähnliche Konstellationen von Sensoren und Aktoren im Brownfield-Produktionssystem. Dieses Modul implementiert die in Kapitel 4.6.4 beschriebene Funktionalität zur automatischen Identifikation und Abgrenzung von häufigen Teilgraphen als Templates unter der Verwendung von Graphalgorithmen.

GSpan Algorithmen sind für diese Aufgabe weitverbreitet, wie in Kapitel 4.6.4 gezeigt wurde, jedoch gibt es noch kaum Implementierungen für die Graphdatenbank Neo4j. Die firmeneigene Implementierung in der Neo4j Graph Data Science (GDS)-Bibliothek zur Identifikation häufiger Subgraphen ist bisher nur als (pre-)alpha Release verfügbar und noch fehlerbehaftet. Darüber hinaus sind die bestehenden Implementierungen nicht oder nur selten kompatibel mit Neo4j. Daher muss bei diesen Implementierungen immer ein Umweg über csv, json oder andere Austauschformate gemacht werden. Das Modul zur Identifikation von häufigen Teilgraphen ist über einen alternativen Weg mit Community Detektion Algorithmen implementiert, da die einzige öffentlich verfügbare GSpan Implementierung [187] für Neo4j noch nicht ausgereift ist und für keinen der erstellten Graphen von bestehenden Produktionssystemen konvergiert. Dabei konnte weder auf einem normalen Laptop noch auf speziellen, leistungsstarken Rechnern für maschinelles Lernen ein Ergebnis erzielt werden. Die Neo4j GDS Bibliothek beinhaltet für die Community Detektion verschiedene Algorithmen, die direkt auf einem Graphen in Neo4j ausgeführt werden können. Louvain und Label Propagation sind produktiv veröffentlichte Algorithmen, die für die Struktur des Graphen geeignet sind und auf einem Bürolaptop in Sekunden ausgeführt werden können und nicht wie die gSpan-Implementierung mehrere Tage benötigen.

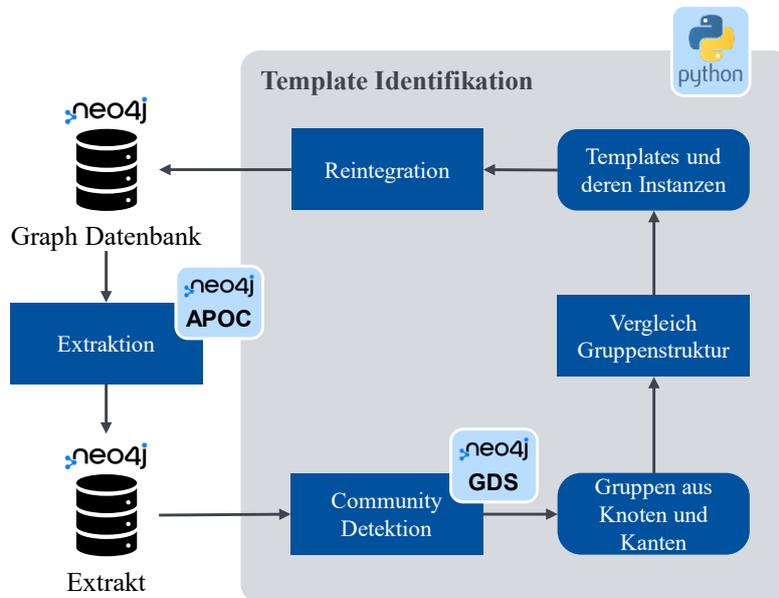


Abbildung 5.9: Submodule der Template-Identifikation

Das Submodul zur Template-Identifikation (siehe Abbildung 5.9) verwendet als ersten Schritt das APOC-Plugin von Neo4j, um ein Extrakt der Datenbank zu erzeugen, das keine Komponenten des Automatisierungssystems beinhaltet. Durch die hohe Anzahl der Verbindungen der Automatisierungshardware zur Peripherie in verschiedenen Teilen des Produktionssystems verzerren diese Elemente die Bildung von Gruppen im Wissensgraphen und erhöhen die Laufzeit der Analyse deutlich. Die Extraktion erleichtert durch die vorherige Größenreduktion die Identifikation der häufig vorhandenen Strukturen. Dazu werden nur Knoten und deren Kanten in eine separate Datenbank kopiert, welche nicht vom Typ PLC oder Device sind und zum Automatisierungssystem gehören (Abbildung 5.10). Anschließend gruppiert das Modul mittels des Community-Detektion-Algorithmus aus der Neo4j GDS Bibliothek die Knoten und Kanten. In Abbildung 5.10 sind die Knoten manuell räumlich in diesen Gruppen angeordnet.

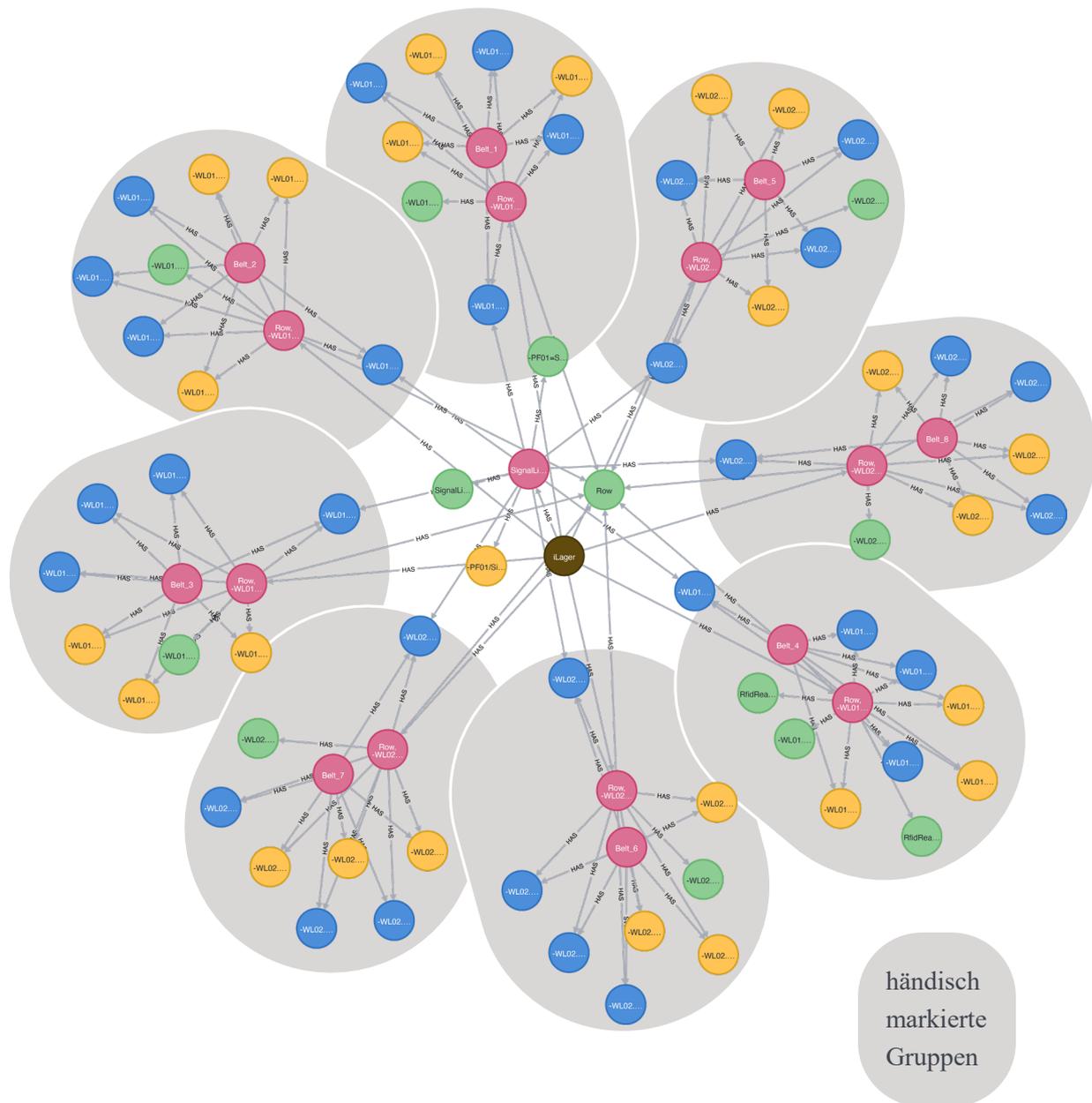


Abbildung 5.10: Manuell sortierter Extrakt des Wissensgraphen

Diese Gruppen (auch Communities genannt) aus eng verbundenen Knoten werden anschließend hinsichtlich ihrer Struktur und dem Typ der Knoten miteinander verglichen. Alle Gruppen, die eine identische Struktur und Knotentyp haben, werden als Templates zusammengefasst. Dadurch entstehen Templates, welche die Struktur und Labels, jedoch keine Werte für die Eigenschaften der Namen und IDs beinhalten. Die Elemente der Gruppen werden um einen Identifier, welcher auf das allgemeine Template verweist, und um eine Instanz-ID erweitert, welche die einzelnen Instanzen eines Templates voneinander abgrenzt. Der Reintegrationsschritt überträgt diese zusätzlichen Informationen abschließend anhand der Namen und IDs wieder in die vollständige Datenbank.

5.5 Exportmodul aus dem Wissensgraphen

Der Wissensgraph in der Neo4j Datenbank beinhaltet die funktionalen und räumlichen Relationen der Peripherie des bestehenden Produktionssystems, ergänzende Eigenschaften (zum Beispiel die Hardwareadresse, den Namen, etc.) jeder Peripherie und die Klassen-Instanz-Relationen in Form der extrahierten Templates. Das Wissen wird über dieses Exportmodul für die Erstellung Digitaler Zwillinge in anderen Softwaretools verfügbar gemacht. Die Bereitstellung ist über das standardisierte, XML-basierte Austauschformat AutomationML (AML) implementiert, das viele Entwurfstools inzwischen importieren können. Dabei unterscheidet sich die Benennung der Strukturelemente des AML-Formats abhängig vom Zielsystem. Für das Exportmodul ist daher eine Anpassung an das Zielsystem, in dem der Digitale Zwilling erstellt werden soll, erforderlich.

In diesem Assistenzsystem wurde exemplarisch das Exportmodul für den Siemens Automation Designer als Zielsystem realisiert. Der Export des Wissensgraphen verläuft dabei aus der Perspektive des Produktionssystems von grob nach fein (top-down) anhand der hinterlegten Ontologie unter dem Wissensgraphen. Der Export beginnt mit den Knoten, die dem Konzept „*MechatronicSystem*“ aus der Basisontologie zugeordnet sind (vergleiche Abbildung 4.11). Anschließend arbeitet der Export hierarchisch weiter nach unten über die Knoten des Konzepts „*Module*“ sowie deren untergeordneten Knoten des Konzepts „*Component*“.

```

1 <CAEXFile SchemaVersion="2.15" FileName="AD-AMLImport.aml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ims"
2 <AdditionalInformation>
3 <WriterHeader>
4 <WriterName>AutomationML Editor</WriterName>
5 <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
6 <WriterVendor>AutomationML e.V.</WriterVendor>
7 <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
8 <WriterVersion>5.3.3.0</WriterVersion>
9 <WriterRelease>5.3.3.0</WriterRelease>
10 <LastWritingDateTime>2022-11-04T12:31:36.554091</LastWritingDateTime>
11 <WriterProjectTitle>unspecified</WriterProjectTitle>
12 <WriterProjectID>unspecified</WriterProjectID>
13 </WriterHeader>
14 </AdditionalInformation>
15 <InstanceHierarchy Name="FunctionAspect">
16 <InternalElement Name="WD1" ID="UN027" RefBaseSystemUnitPath="ADNXSUCLib/Type Library/Unit">
17 <Description>Overwritten</Description>
18 <Attribute Name="Function">
19 <Attribute Name="Aspect Name set by">
20 <Value>User</Value>
21 </Attribute>
22 <Attribute Name="Designated">
23 <Value>True</Value>
24 </Attribute>
25 <Attribute Name="Designation">
26 <Value>WD1</Value>
27 </Attribute>
28 <Attribute Name="Multi-level Reference Designation">
29 <Value>WD1</Value>
30 </Attribute>
31 <Attribute Name="Name">
32 <Value>WD1</Value>
33 </Attribute>
34 <Attribute Name="Parent">
35 <Value />
36 </Attribute>
37 </Attribute>
38 <Attribute Name="Location">
39 </Attribute>
40 <Attribute Name="Product">
41 </Attribute>
42 <Attribute Name="Aspect Name set by">
43 </Attribute>
44 <Attribute Name="Designated">
45 <Value>True</Value>
46 </Attribute>
47 <Attribute Name="Designation">
48 </Attribute>

```

Abbildung 5.11: Export des Wissens als AML-Datei in „pretty print“ Strukturierung

Dieses Vorgehen erzeugt die klassische Baumstruktur von CAx-Entwicklungstools und kann optional auf die TBox erweitert werden, um eine feingliedrigere Auflösung der Baumstruktur zu erhalten. Das Exportmodul wurde in Python 3.11.3 unter der Verwendung der xml.etree.ElementTree-Bibliothek implementiert. Im Vergleich zum EA/RTLS-Analysemodul ist dieses Modul auf eine neue Python Version portiert, die über eine Funktion verfügt, um den AML-Export optisch zweckmäßig eingerückt abzuspeichern (engl. Pretty print), wie in Abbildung 5.11 exemplarisch dargestellt ist. Diese AML-Datei wird im Zielsystem, in diesem Fall dem Siemens Automation Designer, importiert und die Relationen des Digitalen Zwillings dadurch automatisiert erstellt. Abbildung 5.12 zeigt das Ergebnis mit zusätzlich manuell ergänztem CAD-Modell als optische Visualisierung.

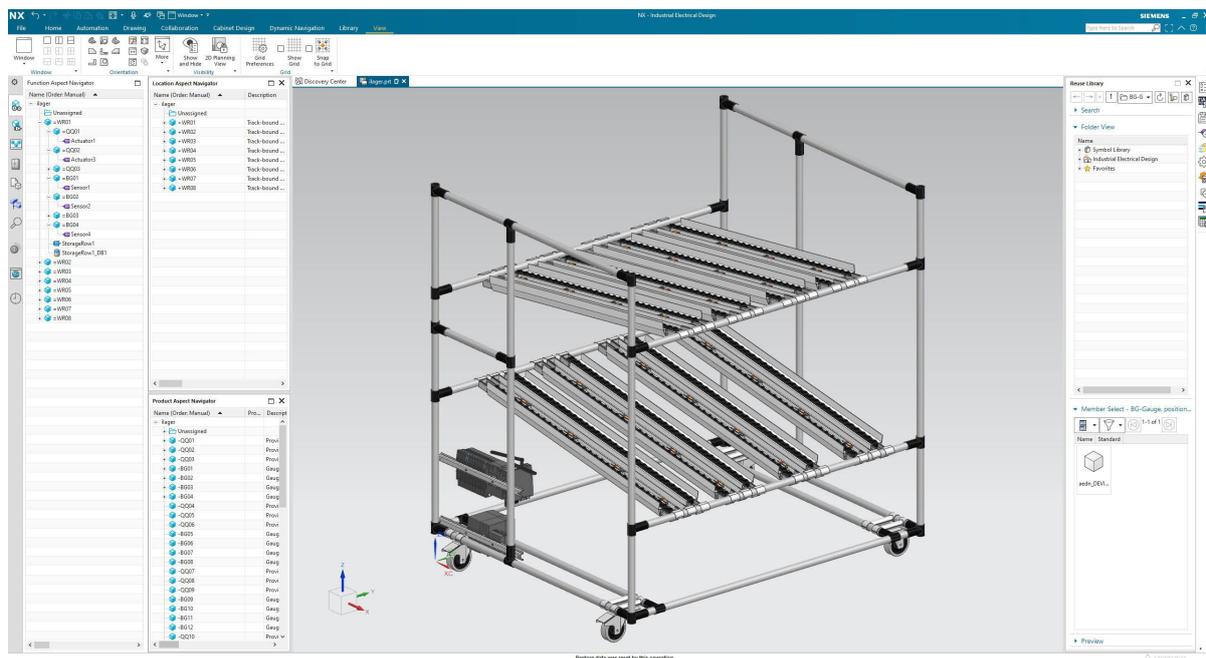


Abbildung 5.12: Resultierender Digitaler Zwilling im Automation Designer mit manuell ergänztem CAD-Modell

In Kapitel 5 wurde die Implementierung der entwickelten Methodik als Software-Assistenzsystem und dessen einzelne Softwaremodule erläutert. Im folgenden Kapitel wird die Evaluierung der Methodik vorgestellt. Dabei wird unter anderem die Anwendung des Assistenzsystems auf ein Brownfield-Produktionssystem mit dem bestehenden manuellen Vorgehen verglichen, um die Mehrwerte aufzuzeigen.

6 Evaluierung

Die entwickelte Methodik wird in diesem Kapitel anhand des implementierten und im letzten Kapitel beschriebenen Software-Assistenzsystems evaluiert. Dazu wird im vorliegenden Kapitel zuerst ein Überblick über das Vorgehen der Evaluierung als Evaluierungskonzept beschrieben. Anschließend werden in den nachfolgenden drei Unterkapiteln die Schritte der Evaluierung und deren Ergebnisse detailliert beschrieben und eingeordnet.

6.1 Evaluierungskonzept

Die Evaluierung der vorliegenden Forschung und die darin entwickelte Methodik erfolgt anhand von drei Evaluierungsaspekten, wie in Abbildung 6.1 dargestellt. Als erster Evaluierungsaspekt wird die Zielerfüllung überprüft. Dazu werden die Elemente der Methodik beziehungsweise deren Implementierung im Software-Assistenzsystem den initialen Anforderungen an das Konzept zugeordnet sowie deren Erfüllung beurteilt. In einem zweiten Schritt werden die Mehrwerte der Methodik gegenüber einer manuellen Erstellung der Relationen eines Digitalen Zwillings evaluiert. Die Erfassung des aktuellen Ist-Zustands erfolgt durch Experteninterviews und durch eine manuelle Erstellung der Relationen des Digitalen Zwillings eines Brownfield-Produktionssystems. Kapitel 6.3.1 beinhaltet die detaillierte Beschreibung dieses Systems. Dieser Ist-Zustand wird mit der automatisierten Erstellung der Relationen durch das Software-Assistenzsystem als Implementierung der Methodik für das gleiche Brownfield-Produktionssystem verglichen. Mit Bezug zu den dabei erzielten Ergebnissen werden die quantitativen und qualitativen Mehrwerte hinsichtlich einer Zeiteinsparung und einer Vermeidung von Fehlerquellen diskutiert.

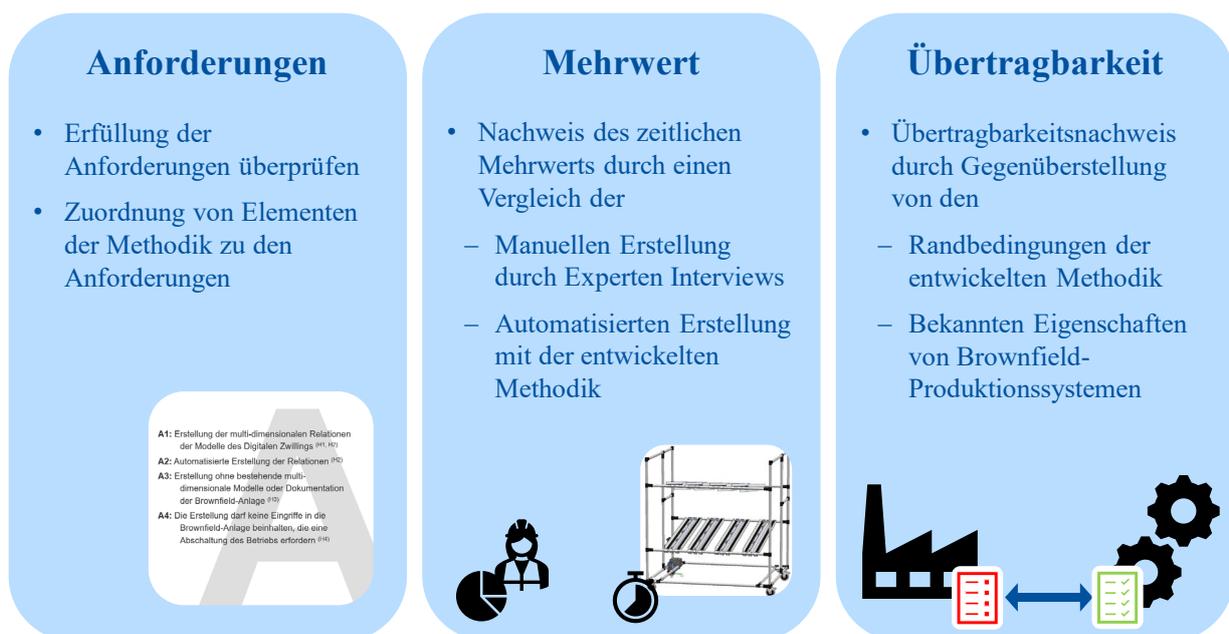


Abbildung 6.1: Die drei Aspekte zur Evaluierung der entwickelten Methodik

Die Evaluierung des zeitlichen Mehrwerts wird anhand des intelligenten Lagers als Demonstrationssystem durchgeführt. Als letzten Schritt wird die Anwendbarkeit der Methodik auf andere Systeme thematisiert, um eine Generalisierbarkeit der Methodik zu überprüfen. Dies erfolgt durch eine Gegenüberstellung der Voraussetzungen der Methodik mit den bekannten Eigenschaften von Brownfield-Produktionssystemen.

6.2 Erfüllung der Anforderungen durch die Methodik

Die Methodik wurde entwickelt, um die aufgezeigte Forschungslücke zu schließen und infolgedessen eine Annäherung des aktuellen Ist-Zustandes an den angestrebten Soll-Zustand hinsichtlich der Erstellung Digitaler Zwillinge für Brownfield-Produktionssysteme zu erreichen. In Kapitel 1.5 wurden hierzu (Forschungs-)Anforderungen ermittelt und definiert, die von der Methodik zu erfüllen sind, um die bestehenden Herausforderungen bewältigen und das initiale Problem lösen zu können. In diesem Unterkapitel werden daher die bei der Anwendung der Methodik abzudeckenden Anforderungen überprüft. Hierfür werden die Elemente des implementierten Software-Assistenzsystems und die durch diese erzeugten anforderungsbezogenen Informationen aufgelistet. Dazu werden die zugrunde liegenden Mechanismen der Methodik und ihre Implementierung den Anforderungen zugeordnet.

Zur Erfüllung der Anforderung (A1) [Automatisierte Erstellung zur Reduzierung von Fehlern und der benötigten Zeit] wurde die Methodik als Software-Assistenzsystem implementiert, das lediglich eine manuelle Auswahl der Informationsquellen benötigt. In der aktuellen Implementierung können nach jedem Schritt die Ergebnisse betrachtet und anschließend die Bearbeitung des nächsten Schritts gestartet werden. Dabei muss der Bediener keine eigene Expertise einbringen, sodass alle Schritte auch vollautomatisch ablaufen könnten, und der Ablauf als automatisiert angesehen werden kann.

Die Analysemethoden der Methodik (Kapitel 4.3 – 4.5) extrahieren Informationen über die Systemrelationen, die im Wissensgraph (Kapitel 4.6) münden. Die erzeugten Relationen bilden den dominanten Teil des Wissensgraphen und bilden die Beziehungen der Peripherie des Produktionssystems ab. Der Wissensgraph kann über die graphische Oberfläche der Graphdatenbank eingesehen werden. Diese Bausteine sind für die Erfüllung der Anforderung (A2) [Erstellung der Relationen automatisierter, mechatronischer Systeme] verantwortlich. Die im Wissensgraph abgespeicherten Relationen werden im Exportmodul verwendet, um die AML-Datei zu erzeugen. Der Import der AML-Datei in das Zielsystem zur Modellierung des Digitalen Zwillinges ermöglicht die weitere Arbeit mit den erstellten Relationen zwischen den Modellen des Digitalen Zwillinges.

Die Methodik baut auf dem SPS-Code, Betriebsdaten und Positionsdaten auf und erfordert keinen weiteren produktionssystemspezifischen Input. Die Analyse des SPS-Codes ist durch den SPS-Adapter vom Hersteller der SPS entkoppelt und auf einem allgemeinen SPS-Code-Modell

realisiert. Die Betriebs- und Positionsdaten müssen lediglich gewisse Informationen pro Eintrag besitzen, wie den Namen des SPS-Signals oder die ID des Positionstransponders. Der Analysevorgang bedarf kein manuelles Mitwirken eines Menschen, der zusätzliche Informationen hinzufügen oder im Analyseprozess manuelle Entscheidungen treffen muss. Der Ablauf der Methodik sollte dennoch durch einen Experten begleitet werden, der die Ergebnisse verifiziert und eine zusätzliche Kontrollinstanz bildet. Optional kann die Basisontologie des ODPs durch eine systemspezifische TBox ergänzt werden, um die Aussagekraft der Ontologie-Konzepte zu erhöhen. Dieser freiwillige Schritt benötigt zusätzliches Expertenwissen über das Brownfield-Produktionssystem. Insgesamt ist die Methodik ohne (Experten-)Wissen über das bestehende Produktionssystem verwendbar und erfüllt daher die Anforderung (A3) [Erstellung der Relationen des Digitalen Zwillings ohne (semi-)formales Expertenwissen].

Die Erfüllung der Anforderung (A4) [Erstellung des Digitalen Zwillings ohne Beeinträchtigung des produktiven Betriebs] ist anhand der verwendeten Informationsquellen der Methodik nachweisbar. Die Informationsquellen sind die Schnittstelle zum Brownfield-Produktionssystem und geben Auskunft über dieses. Die Software, in der der Digitale Zwilling implementiert werden soll, ist unabhängig vom Produktionssystem und die Erstellung des Digitalen Zwillings in diesem hat keinen direkten Einfluss auf den Betrieb. Der SPS-Code als Informationsquelle kann dabei entweder über einen Backup-Server bezogen, wie die Autoren in [97] vorschlagen, oder direkt von der SPS geladen werden. Die Betriebs- und Positionsdaten sind oft bereits vorhanden, da solche Daten im Zuge der Digitalisierung und dem fortschreitenden Einsatz von KI einen steigenden Wert besitzen. Alternativ ist der Abgriff der Betriebsdaten von der SPS im Betrieb mit geringem Aufwand möglich, zum Beispiel über einen integrierten OPC UA oder Webserver in der SPS. Die Positionsdaten sind unabhängig vom Betrieb über ein autarkes Lokalisierungssystem erfassbar und haben keinen Einfluss auf den bestehenden Betrieb.

Die entwickelte Methodik erfüllt damit die Anforderungen aus Kapitel 1.5. Dadurch werden die Herausforderungen in Kapitel 1.3 adressiert, der Erstellungsprozess des Digitalen Zwillings eines Brownfield-Produktionssystems unterstützt und die Erreichung des Soll-Zustands erleichtert. Im nachfolgenden Kapitel wird der Mehrwert der Methodik evaluiert und daran die Erleichterung quantifiziert.

6.3 Evaluierung der Mehrwerte der Methodik

Die Evaluierung der Methodik in diesem Unterkapitel geht über die vorangehende Evaluierung hinsichtlich der Erfüllung der Anforderungen hinaus und zeigt, dass die Nutzung der Methodik einen Mehrwert gegenüber der Ist-Situation bietet [188]. Hierzu wird die Methodik auf das intelligente Lager (iLager) des FlexCell¹¹ Projektes in der ARENA2036¹² angewandt und die Arbeitsschritte sowie deren benötigte Dauer erhoben. Das Projekt und das enthaltene iLager als Brownfieldsystem sind im nachfolgenden Unterkapitel beschrieben. Dem wird der manuelle Erstellungsprozess der Relationen des Digitalen Zwillings gegenübergestellt, der den aktuellen Ist-Zustand abbildet.

Die Dauer des manuellen Vorgehens wird über eine manuelle Erstellung der Relationen des Digitalen Zwillings am intelligenten Lager erhoben und durch die Abschätzung von Experten in Interviews bestätigt und untermauert (Kapitel 6.3.3). Das automatisierte Vorgehen und die Feststellung des manuellen Prozesses werden im Kapitel 6.3.2 erläutert. Abschließend wird eine Gegenüberstellung der beiden Vorgehensweisen vorgenommen und die Evaluierungsergebnisse bewertet.

6.3.1 Brownfield-Produktionssystem zur Demonstration und Evaluierung der Methodik

Das Brownfield-Produktionssystem „FlexCell“ aus dem gleichnamigen Projekt in der ARENA2036 wird für die Demonstration der Methodik und deren Evaluierung herangezogen. Konkret wird die Evaluierung am intelligenten Lager der FlexCell vorgenommen. Das FlexCell-Projekt verkörpert eine Gemeinschaftsforschung der Firmen Siemens, Kuka und Trumpf sowie dem IAS. Das Projekt hat das Ziel, eine durchgängige, digitale Entwicklungstoolkette für Produktionssysteme zu schaffen und deren Einsparpotenzial zu demonstrieren. Diese durchgängige digitale Toolkette soll den Entwurf flexibler Produktionssysteme, eine adaptive Produktionsplanung und eine problemlose Rekonfiguration des Systems mit allen Vorteilen einer modellbasierten Entwicklung ermöglichen. In der vorliegenden Arbeit zur Erstellung der Relationen als Grundlage eines Digitalen Zwillings werden die Ergebnisse der digitalen Entwicklung aus dem Projekt vernachlässigt und nur das Produktionssystem genutzt. Darüber hinaus sind die involvierten Projektpartner und Tool-Experten für die Erfassung des Ist-Zustandes hinsichtlich der manuellen Erstellung eines Digitalen Zwillings qualifiziert und werden, wie später beschrieben, in die Evaluierung eingebunden sein.

¹¹ Kurzform für flexible Produktionszelle

¹² ARENA2036 (Active Research Environment for the Next generation of Automobiles 2036) ist ein Forschungscampus der Universität Stuttgart, in dem verschiedene Institute und Unternehmen aus der Industrie zusammenarbeiten. Die Zielsetzung ist die Erforschung von Themen um das Automobil und dessen Produktion in der Zukunft, konkret im Jahr 2036, dem 150-jährigen Jubiläum des Automobils. Der Forschungscampus besteht aus mehr als 40 Industriepartner, die unter der Führung der Universität Stuttgart eine Vernetzung von Wissenschaft und Wirtschaft schaffen und durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert werden.

Das FlexCell-System besteht aus drei Teilsystemen: dem intelligenten Lager, einem mobilen Roboter und einer Laserschweißanlage. Zusätzlich besitzt das System eine übergeordnete Steuerung zur Koordination der Teilsysteme und des gesamten Produktionsablaufs. Die Interaktion mit dem Bedienpersonal findet über ein Bedienpanel als Mensch-Maschine-Schnittstelle¹³ statt. Jedes Teilsystem besitzt eine eigene SPS und führt die Aufträge der übergeordneten Steuerung selbständig aus. Abbildung 6.2 zeigt ein Bild der FlexCell mit dem intelligenten Lager im Vordergrund. Im Hintergrund ist links die übergeordnete Steuerung (Integra 4) in doppelter Ausführung sichtbar, einmal für die Anlagensteuerung und eine weitere als Testsystem von parallelen Entwicklungsversuchen. Rechts daneben ist im Hintergrund die Laserschweißmaschine TrueLaser Cell3000 der Firma Trumpf zu sehen. Die Laserschweißanlage wird im Projekt zum Verschweißen und Beschriften von Metallteilen verwendet. Rechts hinter dem Lager, leicht verdeckt, ist der mobile Roboter KMR iiwa¹⁴ der Firma Kuka erkennbar. Dieser besteht aus einer mobilen, autonom fahrenden Plattform und einem darauf montierten Kuka LBR iiwa¹⁵ Roboterarm. Im Projekt wird dieser mobile Roboter für den Transport des Materials und den Fügeprozess der Einzelteile verwendet.



Abbildung 6.2: Blick auf das intelligente Lager mit dem teilweise verdeckten mobilen Roboter, der Laserbearbeitungsmaschine und der übergeordneten Steuerung im Hintergrund

¹³ Besser bekannt unter dem englischen Begriff: Human-Machine-Interface (HMI)

¹⁴ Kuka Mobile Robotik (KMR) intelligent industrial work assistant (iiwa)

¹⁵ Leichtbauroboter (LBR) intelligent industrial work assistant (iiwa)

In Abbildung 6.3 ist das CAD-Modell der FlexCell mit allen Teilsystemen und einer nachträglichen Beschriftung zu sehen.

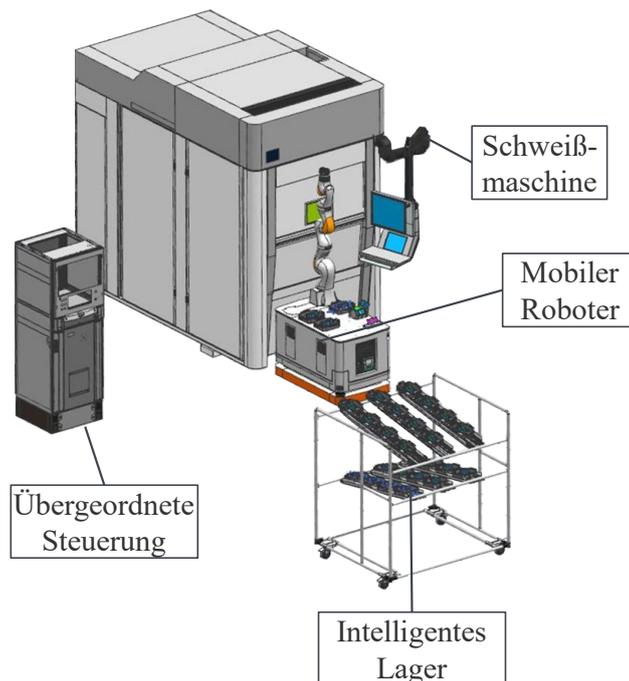


Abbildung 6.3: CAD-Modell der FlexCell mit nachträglicher Beschriftung

Die FlexCell produziert kleine Modellautos aus vorgefertigten Blechteilen, die in sogenannten Werkstückträgern, Metallwannen mit einer Haltevorrichtung für die Rohteile, lagern. Dabei werden die Bauteile in vier Baugruppen zusammengefasst, welche jeweils in einem Werkstückträger aufbewahrt und durch das intelligente Lager bereitgestellt werden. Hierfür besitzt das intelligente Lager zwei Lagerebenen mit jeweils vier Lagerreihen. Die untere Lagerebene repräsentiert die Ausgabe und stellt befüllte Werkstückträger zur Verfügung. Die obere Lagerebene ist für die Rücknahme zuständig und nimmt entleerte Werkstückträger ohne Bauteile wieder entgegen. Die Lagerebenen sind entgegengesetzt geneigt, sodass ein Transport der leeren und vollen Werkstückträger in unterschiedliche Richtungen stattfinden kann. Der mobile Roboter entnimmt die Werkstückträger mit Material aus dem intelligenten Lager und fügt die darin enthaltenen Teile in Fügevorrichtungen zusammen. Anschließend stellt er leere Werkstückträger in der Rücknahme des Lagers wieder ab. Der Roboter fährt mit dem gefügten Modellauto zur Laserschweißmaschine und lässt dieses dort verschweißen oder am Ende kundenindividuell lasergravieren. Die Schritte vom Beschaffen des Materials über den Füge- und Schweißvorgang sowie dem Zurückstellen der leeren Werkstückträger wird mehrfach wiederholt, bis das komplette Auto aus den vier verschiedenen Bauteilgruppen fertig ist. Der Produktionsablauf wird dabei durch die übergeordnete Steuerung koordiniert und beispielsweise die Materialverfügbarkeit im intelligenten Lager geprüft, der Entnahme- und Fügevorgang durch den Roboter beauftragt oder der Schweißauftrag gestartet.

Das intelligente Lager selbst besteht aus 24 Aktoren zur Steuerung des Materialflusses in den Lagerreihen und zur Separierung der Werkstückträger, einer Signalleuchte zur Benachrichtigung von Werkern zum Nachfüllen von Werkstückträgern und 35 Sensoren zur Detektion der Werkstückträger sowie deren Identifikation über RFID-Tags. Diese Sensoren und Aktoren sind an einer Siemens ET200sp SPS angeschlossen und werden durch diese autark ausgelesen und gesteuert. Die SPS stellt die Informationen über den offenen Industrial-Ethernet-Standard „Profinet“ mit Hilfe des eingebauten industriellen WLAN-Clients der übergeordneten Steuerung und dem mobilen Roboter zur Verfügung. Außerdem ist ein Akku für eine unterbrechungsfreie Stromversorgung verbaut, sodass das Lager frei auf seinen Rädern durch die Halle beweglich ist ohne jegliche Kabelbindung.

Tabelle 3: Komponenten des intelligenten Lagers

Gruppe	Komponente	Anzahl
Sensoren	Lichttaster mit Hintergrundausblendung HAT3C.BXL/4P-200-M12	32
	RFID-Reader RF210R	3
Aktoren	Hubmagnet IST-LS 2924B	24
	Signaldauerlichtelement rot	1
Steuerung	SPS ET200SP: CPU 1512SP F-1 PN	1
	Input-Modul DI8x24V DC ST	6
	Output-Modul DQ8x24V/0,5A ST	4
	IO-Link-Modul CM 4x ST	1
Stromversorgung	Netzteil SITOP PSU100S 24V/10A	1
	USV SITOP UPS1600 24V/10A	1
	Batteriemodul SITOP UPS1100 24V/12Ah	1
	Selektivitätsmodul SITOP PSE200U 10A	1
Kommunikation	Busadapter BA 2xRJ45	1
	iWLAN Client SCALANCE W734-1	1

Die komplette Fläche des Produktionssystems wird von einem Indoor-Lokalisationssystem abgedeckt. Dieses ist nicht in die Steuerung des intelligenten Lagers eingebunden, sondern agiert autark, verarbeitet die Signale der Positionstransponder und stellt deren Position strukturiert bereit. Diese Informationen werden für die übergeordnete Kontrolle und Steuerung des Materialtransports benötigt und sind für die Methodik eine relevante Eingangsgröße. Das System besteht aus den in Tabelle 4 gelisteten Komponenten.

Tabelle 4: Komponenten des Indoor-Lokalisationssystems

Komponente	Anzahl
Stromversorgung SITOP PSU100L 24V/10A	1
Industrie PC IPC427E	1
Empfänger SIMATIC RTLS Gateway RTLS4030G	5
Sender SIMATIC RTLS Transponder RTLS4030T	4
Sender SIMATIC RTLS Transponder RTLS4040T	3

6.3.2 Erstellung des Digitalen Zwillings für das intelligente Lager mithilfe der entwickelten Methodik

Die Erstellung des Digitalen Zwillings wurde mit dem Software-Assistenzsystem als Realisierung der Methodik durchgeführt. Das intelligente Lager wurde dazu physikalisch in Betrieb genommen und das Lokalisierungssystem für die Erfassung der Positionsdaten gestartet. Diese zwei Systeme wurden anschließend beide über Ethernet mit einem handelsüblichen Laptop verbunden, das die aktuellen Werte abgreift und in die Datenbank speichert. Zusätzlich wurde zum einen das Programm Influx-OPCUA-logger gestartet, welches die Betriebsdaten vom OPC UA Server der SPS abfragt, zum anderen das Skript zur Erfassung der Positionsdaten aufgerufen (vgl. Kapitel 5.3.1). Die Erfassung der Positionsdaten wurde zu Beginn im halbautomatischen Modus ausgeführt, in dem Labels manuell eingestellt werden. Dieser Erfassungsmodus wurde als Grundlage für die Klassifikation der Peripheriepositionen ungefähr 45 min beibehalten bevor in den vollautomatischen Modus gewechselt wurde. In diesem wurde für weitere vier Stunden das Lager betrieben und Positions- sowie Betriebsdaten erfasst. Inklusiv aller anfänglicher Schwierigkeiten mit der Inbetriebnahme und Verbindung aller Systeme konnte in unter acht Stunden der benötigte Datensatz erfasst werden. Daran anschließend wurden die Daten manuell analysiert und bereinigt, um Übertragungs- und Erfassungsfehler zu entfernen. Dazu wurden die Zeitreihen visualisiert und beispielsweise Datenpunkte entfernt, die bei der Umstellung des Labels im halbautomatischen Betrieb ohne ein solches abgespeichert wurden oder Positionen, die außerhalb des Bereichs des Lagers sind und durch die manuelle Materialrückführung entstanden sind. Die Datenbereinigung konnte in unter einem halben Tag abgeschlossen werden. Im Bereich der SPS-Code-Analyse wurde das Metamodell an den SPS-Code des intelligenten Lagers angepasst, da das entwickelte Metamodell auf dem SPS-Code des modularen Produktionssystems (MPS) am IAS basierte. Dieser ist in TIA V14.1 implementiert und das korrespondierende Metamodell daher nur eingeschränkt für den SPS-Code des iLagers nutzbar, der in TIA V15.1 implementiert ist. Die Anpassung erfolgte über einen manuell angestoßenen XML-Export unter Zuhilfenahme von TIA Openness und der folgenden automatisierten Erstellung einer XSD mittels der Software LiquidXML Studio aus den XML-Dateien des iLagers und des MPS. Der Anpassungsvorgang dauerte ungefähr einen halben Tag.

Nach dieser Vorbereitung wurden in der Konfiguration des Assistenzsystems die aktuellen IP-Adressen und Zugangsdaten der Zeitreihendatenbank und der Graphdatenbank ergänzt und anschließend das Assistenzsystem gestartet. Der eigentliche Ablauf der Methodik, implementiert als Assistenzsystem, nahm weniger als eine halbe Stunde in Anspruch. Dabei wurden bereits die Ergebnisse der Zwischenschritte betrachtet und auf Plausibilität geprüft. Der reine Analyseprozess ohne weitere Verifikation ist in etwa 15 Minuten durchführbar.

6.3.3 Erfassung des Ist-Zustands der manuellen Erstellung des Digitalen Zwillings durch eine Expertenbefragung

Der aktuelle Ist-Zustand hinsichtlich der manuellen Erstellung der Relationen des Digitalen Zwillings im Automation Designer wurde mithilfe von zehn Expertenbefragungen erhoben. Hierfür wurden unter anderem Projektpartner, Mitarbeiter des Toolherstellers und Experten aus anderen Industriefirmen sowie ein wissenschaftlicher Mitarbeiter befragt¹⁶. Um sicherzustellen, dass die Experten fundierte Aussagen über die manuelle Erstellung machen können, wurden nur solche ausgewählt, die Erfahrung mit dem Automation Designer hatten. Die größte Expertengruppe mit vier Personen kam dabei aus dem Umfeld der Entwicklung, Support und Vertrieb des Automation Designer. Bei diesen Experten ist davon auszugehen, dass sie die Dauer der Erstellung aufgrund ihrer Erfahrung mit dem Tool eher kürzer ansetzen, möglicherweise auch, um die Fähigkeit ihrer Software nicht herabzustufen. Ungefähr die Hälfte der befragten Personen kannten das intelligente Lager schon aus vorheriger Zusammenarbeit oder aus Workshops im FlexCell-Projekt. Den anderen Interviewpartnern wurde das intelligente Lager über eine ca. fünfminütige Vorstellung anhand der Bilder und einer verbalen Erklärung des Aufbaus und seiner Funktionen sowie durch die Aufgabenbeschreibung der FlexCell erläutert. Die Befragung wurde mit einem Fragebogen mit festen Antwortmöglichkeiten sowie der Möglichkeit zur Ergänzung von Freitextantworten durchgeführt. Des Weiteren wurde die manuelle Erstellung des Digitalen Zwillings mit dem Automation Designer im Rahmen dieser Forschung auch durchgeführt und die Dauer der einzelnen Schritte ermittelt. Diese Zeiten wurden bei der Auswertung der Dauer der Schritte zu den Zeiten der Expertenbefragungen hinzugefügt. Für die Auswertung der Zeiten standen damit elf Werte (zehn Befragungen und eine eigene, manuelle Durchführung) zur Verfügung. Diese wurden gemittelt, um Fehleinschätzungen durch mangelndes Fachwissen einzelner Experten für einzelne Schritte der Erstellung der Relationen eines Digitalen Zwillings im Automation Designer zu eliminieren. Die Befragten gaben die Dauer in den meisten Antworten in Stunden an, in einzelnen Fällen jedoch auch in Tagen oder Wochen. In diesem Fall wurde nachgefragt und klargestellt, dass ein Tag mit acht Stunden und eine Woche mit fünf Tagen respektive 40 Stunden berechnet wird. Diese Umrechnung liegt auch den weiteren Diagrammen zugrunde. Für die Berechnung der durchschnittlichen Dauer der manuellen Erstellung wurde für

¹⁶ Die Experten arbeiten bei den folgenden Unternehmen und Instituten (alphabetische Reihenfolge): ABB (1), IAS (1), Kuka Systems (2), Lapp Service GmbH (1), Siemens AG (4), Siemens Healthcare GmbH (1),

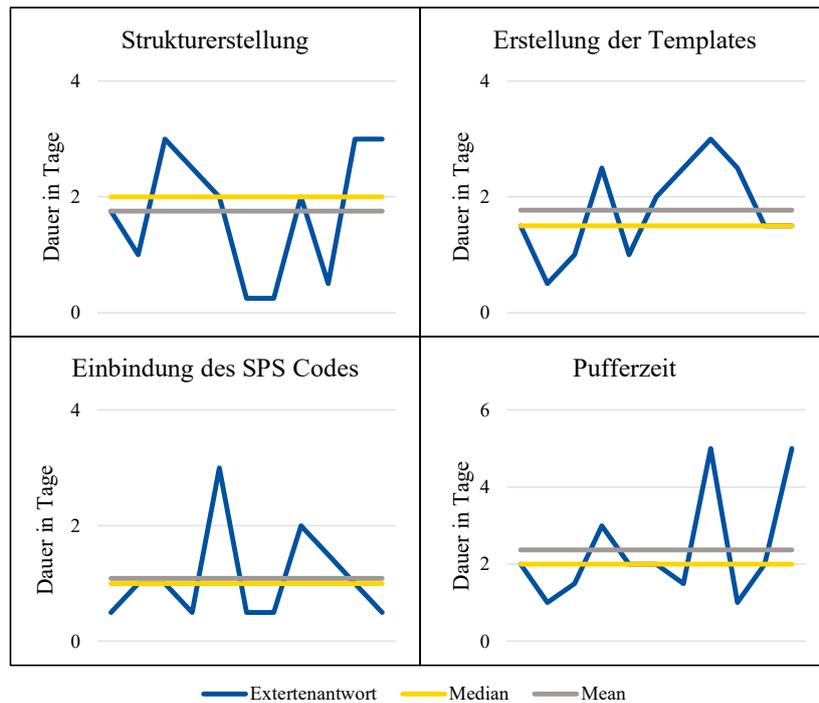


Abbildung 6.5: Vergleich des Mean (grau) und Median (gelb) im Vergleich zu den Expertenantworten (blau) für die Fragen zwei bis fünf

Der manuelle Prozess beginnt mit der Untersuchung des technischen Systems, in diesem Fall des intelligenten Lagers. Der komplette Prozess ist in Abbildung 6.6, grafisch aufgearbeitet, dargestellt. Die meisten Experten haben rund zwei Stunden bis etwa einen Tag für die Untersuchung eingeplant. Nur eine Antwort liegt mit 15 Tagen außerhalb des üblichen Antwortbereichs. Dieser Ausreißer wird durch die Wahl des Median für die Berechnung der durchschnittlichen Dauer von einem halben Tag ausgeschlossen. Die meisten Experten überprüfen dabei die Software, Sensoren und Aktoren. Gleichermaßen kontrollieren die Experten die Mechanik, finden diese Information aber etwas weniger wichtig. Zwei Drittel der Experten betrachten die Verdrahtung zwischen den Komponenten und der SPS. Weniger als die Hälfte analysiert die Feldbusanbindung und das auch nur mit der niedrigsten Priorität. An die Untersuchung des intelligenten Lagers schließt sich der Schritt der Erstellung der Systemstruktur im Automation Designer an. Dies ist mit zwei Tagen der größte Teil des manuellen Prozesses und wird durch den Analyseteil der automatisierten Methodik abgedeckt. Anschließend wird die (Baum-)Struktur des Produktionssystems in den drei verschiedenen Betrachtungsaspekten erstellt. Diese Struktur beinhaltet „anonyme“ Knoten der Komponenten des Systems und die Relationen zwischen diesen Komponenten. Dabei werden jedoch keine Modelle der einzelnen Komponenten in den Fachbereichen (CAD-Modell, ECAD, etc.) erstellt. Im Ist-Ablauf folgt anschließend der Import des SPS-Codes im Automation Designer und dessen Einbindung in die Struktur. Die Funktions- und Datenbausteine werden dazu in die Struktur eingebettet sowie deren Eingangsparameter mit den Signalen der Peripherie verbunden. Aus dieser Struktur werden abschließend Templates der sich wiederholenden Anordnungen erstellt, die instanziiert und wiederverwendet werden können. Durch diese Wiederverwendung wird im Ist-Ablauf weiterer

Projekte die meiste Zeit eingespart. Dabei ist jedoch genau die Erstellung der Templates nach Meinung der Experten die Hauptfehlerquelle (70% Erstnennung) und mit einem hohen zeitlichen Aufwand verbunden. Dabei wird die Identifikation geeigneter Templates in den großen Strukturen von Produktionssystemen als Herausforderung angeführt. Weiterhin sehen die Experten die Erstellung des funktionalen Aspekts sowie die richtige Einbindung des SPS-Codes mit jeweils 40% Nennung als Fehlerquelle, da die Zuordnung der SPS-Bausteine zur Peripherie und der Peripherie in die funktionale Struktur nicht visuell am Produktionssystem erkennbar sein muss. Entsprechend wurden diese Schritte in gleicher Weise als Quellen für den zeitlichen Aufwand eingestuft (Nennung Templates: 70%, SPS: 80%, funktionaler Aspekt: 30%).

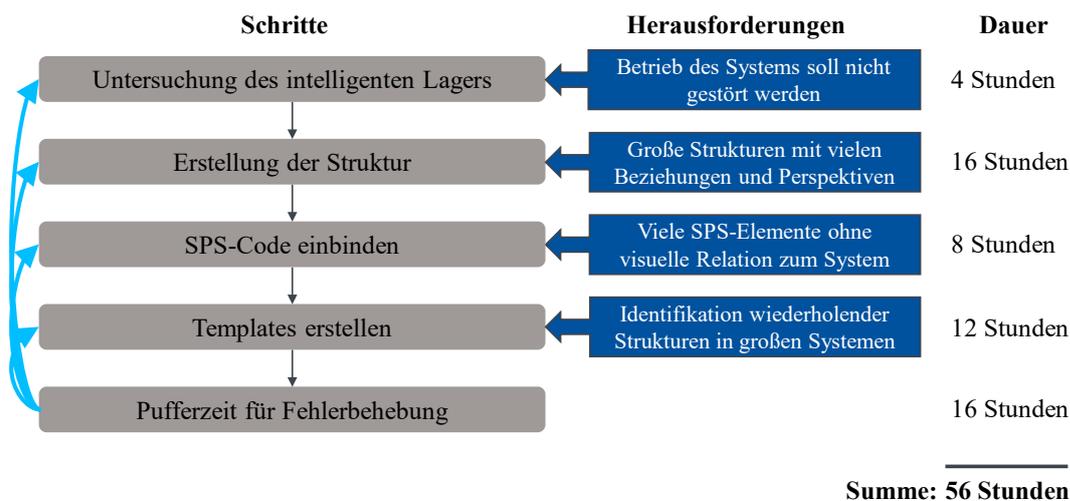


Abbildung 6.6: Manuelle Schritte des Ist-Ablaufs, deren Dauer und Herausforderungen

In den Experteninterviews wurden über die Erhebung der Dauer hinaus durch die Experten Einschätzungen zu Problemquellen geäußert und Begründungen für ihre veranschlagten Zeiten gegeben. Eine explizite Abfrage hat nicht stattgefunden, sie wurde jedoch dokumentiert, falls die Experten sich dazu geäußert haben. Dabei sind die nicht sichtbaren Beziehungen einer funktionalen Struktur sowie die insgesamt große Struktur moderner Produktionssysteme als Herausforderungen genannt worden. Diese bereiten Menschen Schwierigkeiten, einerseits bei der Erfassung, andererseits auch dabei, den Überblick zu behalten. Die Templates sind dabei sowohl Lösung als auch Herausforderung, da diese die Strukturgröße reduzieren und die Erstellung beschleunigen. Dazu müssen initial Templates mit einem hohen Wiederverwendungspotential identifiziert werden, um in der weiteren Verwendung der Templates einen Mehrwert zu liefern.

6.3.4 Diskussion und Bewertung der Evaluierungsergebnisse

Aus dem Vergleich der automatisierten Erstellung mithilfe der Methodik und der manuellen Erstellung hinsichtlich ihrer zeitlichen Dauer ergibt sich aus den letzten beiden Kapiteln das Diagramm in Abbildung 6.7. Es zeigt sich, dass die automatisierte Durchführung eine deutliche Zeiteinsparung erzielen kann. Ein Vergleich der Gesamtdauern ergibt, dass eine Einsparung um etwas mehr als 70% gegenüber dem aktuellen Vorgehen möglich ist. Dabei ist die Pufferzeit voll

einberechnet in der Annahme, dass diese in vielen Fällen benötigt wird. Tritt der beste Fall ein und die Erstellung wird fehlerfrei durchgeführt, ist diese Pufferzeit zu vernachlässigen. Der manuelle Prozess verkürzt sich folglich von sieben auf fünf Tage und dadurch reduziert sich auch der prozentuale zeitliche Mehrwert auf 60% Zeitersparnis. Zudem variiert die Dauer der manuellen Erstellung mit der Erfahrung des Ingenieurs. Die Verwendung der automatisierten Methodik führt zu einer gleichbleibenden Dauer für den Erstellungsprozess. Die Dauer der automatisierten Methodik kann für bestehende Produktionssysteme ebenfalls positiv beeinflusst werden, wenn bereits Daten erfasst wurden oder eine kompatible Version und Hersteller des SPS-Codes verwendet werden. In diesem Fall reduziert sich die Dauer mit der Methodik weiter.

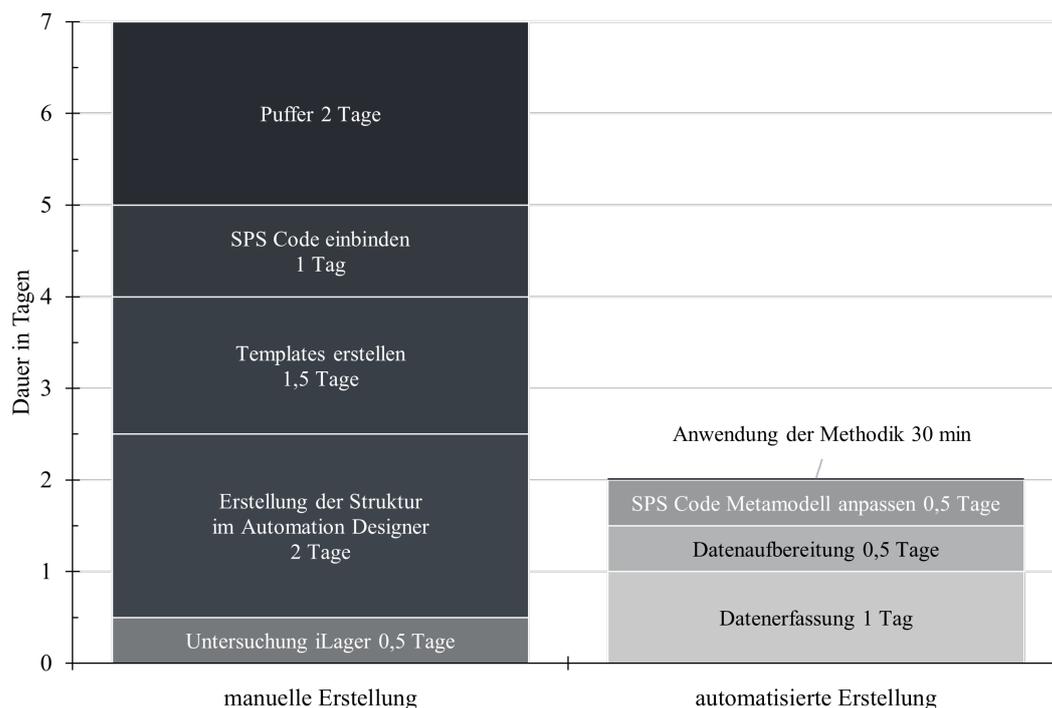


Abbildung 6.7: Gegenüberstellung der Dauer des manuellen und des automatisierten Prozesses zur Erstellung der Relationen des Digitalen Zwillings für das intelligente Lager

Bei der Erstellung des Digitalen Zwillings und der Erhebung der Zeiten wurden die spezifischen Modelle nicht berücksichtigt, sondern auf die Struktur der Relationen Wert gelegt. Dies wurde bei den Interviews auch entsprechend kommuniziert und darauf hingewiesen, dass die zugrunde liegenden CAD-Modelle, Schaltpläne, Simulationsmodelle, SPS-Codes und ähnliche nicht erstellt werden müssen oder bereits existieren. Daher werden diese Elemente nicht in den geschätzten Zeiten berücksichtigt. Die Dauer für deren Erstellung muss zum Erstellungsprozess für einen vollständigen Digitalen Zwilling von Brownfield-Produktionssystemen auf beide Vorgehensweisen hinzuaddiert werden, um eine Einschätzung des kompletten Prozesses zu bekommen. Einige dieser Elemente, wie der SPS-Code, existieren für jedes bestehende System oder können von der Steuerung heruntergeladen werden. CAD-Modelle existieren in manchen Fällen bereits oder sind in Teilen oder vollständig durch automatisierte Methoden wie Laserscans erstellbar. Daher variiert die Dauer für die Erstellung eines vollständigen Digitalen Zwillings für jedes Brownfield-Produktionssystem abhängig von den vorhandenen Gegebenheiten und dem

Einsatz weiterer Methoden. Darüber hinaus ist die Erstellungsdauer ebenfalls abhängig von der geplanten Anwendung, die auf dem Digitalen Zwilling als Basis zum Einsatz kommen soll und der dazu benötigten Modellgranularität. Weitere Einflussfaktoren sind die Größe der Anlage sowie die Dichte der verbauten Komponenten, für die ein Digitaler Zwilling erstellt werden soll. Diese Faktoren werden im Rahmen der Übertragbarkeit in Kapitel 6.4.4 diskutiert.

Insgesamt ist festzustellen, dass die entwickelte Methodik einen zeitlichen Gewinn gegenüber der bisherigen manuellen Erstellung bietet. Die automatisierte Erstellung verringert außerdem die Gelegenheit für Flüchtigkeitsfehler bei der Erstellung der großen Strukturen, sie ist unabhängig von der jeweiligen Erfahrung und dem Wissen des Ingenieurs und generiert dadurch aus der fixen Menge zu betrachtender Informationen objektive, gleichbleibende, generellere Templates. Diese bieten eine breitere Wiederverwendbarkeit für andere Produktionssysteme und erhöhen bei der skalierten Anwendung des Automation Designers auf weitere Anlagen die Zeiteinsparung.

6.4 Übertragbarkeit der Methodik auf andere Brownfield-Produktionssysteme

Als Nachweis, dass diese Methodik die Forschungslücke bei Brownfield-Produktionssysteme schließen kann und nicht nur eine spezialisierte Methodik für das iLager als Evaluierungssystem ist, wird in diesem Kapitel die Übertragbarkeit der Methodik diskutiert. Da die Methodik am modularen Produktionssystem (MPS) am IAS konzipiert und implementiert wurde, bildet der durchgeführte Übertrag der Methodik auf das iLager zur Evaluierung des Mehrwerts einen beispielhaften Übertrag. Dieser wird eingangs vorgestellt und die notwendigen Modifikationen am Assistenzsystem besprochen. Darüber hinaus werden anschließend die Randbedingungen der Methodik betrachtet und den gängigen Eigenschaften von Brownfield gegenübergestellt. Daraus wird eine verallgemeinerte Aussage über die Übertragbarkeit anhand der Voraussetzungen für die Methodik getroffen und bewertet. In dieser Diskussion wird auch eine Einschätzung der Skalierbarkeit der Methodik auf größere Systeme gegeben.

6.4.1 Beispielhafter Übertrag auf das intelligente Lager

Die ursprüngliche Entwicklung der Methodik wurde an einem modularen Produktionssystem von Festo am IAS entwickelt. Dieses System besteht aus vier Förderbändern mit jeweils zwei Bearbeitungsstationen. Die Bearbeitungsstationen und Förderbänder werden über eine zentrale Siemens S7-1500 SPS gesteuert, die in TIA v14.1 programmiert ist, und produzieren kleine Plastikzylinder mit Metalleinsätzen. Im Unterschied dazu ist das iLager durch eine ET200sp mit TIA v15.1 gesteuert und Teil eines flexiblen Produktionssystems, das kleine Modellautos aus Blech fertigt. Abgesehen von der Steuerung unterscheidet sich auch die Größe des MPS mit einer Abmessung von ca. 1m x 1m vom iLager mit ca. 2m x 2m Grundfläche.

Dadurch entsteht vor allem für die Analyse des SPS-Codes ein Unterschied. Das entwickelte Metamodell des SPS-Codes für das MPS ist aufgrund der neueren Version des SPS-Codes nicht direkt auf das iLager übertragbar. Hierzu muss der SPS-Code des iLagers über TIA Openness in XML exportiert, automatisiert durch LiquidXML Studio die XSD-Datei erstellt und aus dieser Datei das Metamodell, bestehend aus den 114 Java Klassen, erweitert werden. Der ganze Prozess kann komplett automatisiert ablaufen. Es muss dabei geprüft werden, dass die Namespaces der XSD-Dateien mit den existierenden XSD-Dateien identisch sind. Der Vorgang zur Anpassung des Metamodells wurde bei der Auswertung des zeitlichen Mehrwertes berücksichtigt und auf die Dauer des automatisierten Prozesses mittels der Methodik aufaddiert. In einer kommerziellen Umsetzung der Methodik ist es möglich, auch diesen Schritt zu entfernen, indem mit den SPS-Herstellern ein Vertrag zu den generalisierten XSD-Dateien abgeschlossen und diese eingebunden werden. Die Hersteller verfügen über diese Dateien, da sie für deren Funktionen in den XML-Exporttools des SPS-Codes hinterlegt sein müssen.

Zusätzlich mussten für das iLager eigene Positionsdaten und Sensor-/Aktorsignale erfasst werden. Dieser Vorgang wurde mit einer Software zur Speicherung von Signalen des OPC UA Servers der SPS in einer InfluxDB Zeitreihendatenbank teilautomatisiert vorgenommen. Parallel wurden über ein eigenes Automatisierungsskript die erfassten Positionsdaten mit Labels versehen, sodass ein gelabelter Datensatz für die Klassifikation durch den INN mit DTW vorlag. Hierfür wird im Skript eingestellt, welcher Bereich (Klasse) nun erfasst wird und anschließend Material in diesen Bereich zugeführt beziehungsweise die Aufträge in diesem Bereich gestartet.

Abgesehen von der Erfassung musste die TBox für das neue System erstellt werden, um eine detaillierte semantische Beschreibung der Wissens Elemente im Wissensgraphen zu erreichen. Dieser Schritt ist durch die generalisierte Ontologie Design Patterns optional und kann übersprungen werden, wenn auf eine detaillierte Beschreibung verzichtet werden kann. In Anwendungsfällen, wie der Nutzung für kontextbasierte Entscheidungsfindungen, bringt dies einen funktionalen Mehrwert [189]. Der Mehraufwand für die Erstellung einer TBox ist jedoch durch die nachfolgende Anwendung getrieben und keine zwingende Notwendigkeit der Methodik.

Abschließend muss bei einem Übertrag das Exportmodul an das Zielsystem und dessen Benennung der AML-Tags angepasst werden. Im betrachteten Übertrag war das Zielsystem mit dem Automation Designer für beide Anwendungen der Methodik identisch und hat keine Anpassung nach sich gezogen.

6.4.2 Randbedingungen der Methodik

Zur Verallgemeinerung der Übertragbarkeit über das aufgeführte Beispiel hinaus sind in diesem Kapitel die Randbedingungen der Methodik aufgelistet und die daraus resultierenden Bedingungen für die Anwendbarkeit der Methodik auf Brownfield-Produktionssysteme herausgearbeitet.

Die Randbedingungen, die primär durch das Brownfield-Produktionssystem erfüllt werden müssen, betreffen die Datenquellen. Der SPS-Code (*RB1.0*) muss strukturelle Bedingungen erfüllen, sodass er durch die Analyse verarbeitet werden kann. Die wichtigste Grundbedingung ist, dass der SPS-Code in einer der fünf standardisierten Programmiersprachen der Norm DIN EN/IEC 61131-3 [142] implementiert wurde, da die Metamodellierung nur die Struktur dieser Sprachen abbildet (*RB1.1*). Eine andere Struktur kann daher nicht durch die Analyse untersucht werden. Prinzipiell ist es denkbar, das Metamodell und die Analyse auf weitere Programmiersprachen zu übertragen und anzupassen. Dies ist jedoch nicht ohne deutlichen Mehraufwand für eine Weiterentwicklung der SPS-Code-Analyse möglich. Für den Import des SPS-Codes ist es außerdem wichtig, dass ein Export des nativen SPS-Code Formats eines Herstellers als XML-Datei möglich ist (*RB1.2*). Optimalerweise ist auch die dazugehörige XSD-Datei verfügbar, jedoch ist dies keine zwingende Bedingung, da diese Datei mit Tools wie LiquidXML Studio oder ähnlichen Softwaretools auch automatisiert aus den XML-Dateien erzeugt werden kann.

Die Betriebsdaten der SPS (*RB2.0*) müssen nur wenige Anforderungen erfüllen. Sie müssen die aktuellen Werte der Sensoren und Aktoren beinhalten und der Signalname, der auch im SPS-Code verwendet wird, muss dem Wert zugeordnet sein (*RB2.1*). Die neuen Werte der Betriebsdaten müssen dabei in ausreichend hoher Frequenz aktualisiert werden, sodass keine Signaländerungen verloren gehen, die durch den technischen Prozess ausgelöst werden (*RB2.2*). Die Methodik selbst ist auf die Verwendung einer Zeitreihendatenbank angepasst und baut auf die Zeitstempel dieser auf. Es ist jedoch auch denkbar, mit Performanceabstrichen eine relationale Datenbank zu verwenden und jedem Eintrag aktiv einen Zeitstempel (*RB2.3*) hinzuzufügen. Der Übertragungsweg ist dabei unabhängig und unterliegt keiner Randbedingung durch die Methodik, da diese erst bei der Datenhaltung ansetzt. Die Datenbasis sollte dabei keine veralteten Daten beinhalten, die vor einer Rekonfiguration der Anlage aufgenommen wurden oder nur marginal wenige im Vergleich zu aktuellen Daten, sodass der Einfluss durch die aktuellen Daten deutlich überwiegt und so eine fehlerhafte Positionierung der Peripherie vermieden wird (*RB2.4*).

Die Positionsdaten unterliegen ähnlichen Restriktionen, da die Methodik auch hier erst ab der Datenbasis greift (*RB3.0*). Sie ist daher entkoppelt vom Erfassungssystem und dem Übertragungsweg. Wichtig ist, vergleichbar mit den Betriebsdaten, eine aktuelle Datenbasis mit Zeitstempeln (*RB3.1*) mit ausreichend hoher Erfassungsfrequenz, die an den technischen Prozess angepasst ist und mindestens so schnell ist wie der Prozess (*RB3.2*). Abgesehen davon sollten mindestens die letzten zehn Produktionsabläufe über alle Sensoren und Aktoren mit Klassenlabel erfasst werden, um eine Verwendung der Positionsdatenanalyse gewährleisten zu können. Die Auflösungsgenauigkeit des Lokalisierungssystems sollte kleiner als der mittlere Abstand der Peripheriekomponenten sein (*RB3.3*), da es sonst zu regelmäßigen Fehlklassifikationen kommt. Die Unterschreitung des Abstandes durch einzelne wenige Komponenten ist vertretbar, dies kann durch die Glättung der Position über einen beziehungsweise mehrere zeitliche Verläufe

kompensiert werden. Übergreifend für die Positionsdaten und die Betriebsdaten muss der Zeitstempel synchronisiert sein, sodass diese einander zuordenbar sind (*RB3.4*).

Die weiteren Anforderungen betreffen nicht das Brownfield-Produktionssystem, sondern richten sich an die Anwendung des Assistenzsystems, dessen Modifikation und die benötigte Hardware. Diese Anforderungen werden nicht im folgenden Kapitel adressiert, sondern in Kapitel 6.4.4 bewertet. Die Graphanalyse benötigt in ihrer aktuellen Implementierung die Kennzeichnung der Knoten über die Konzepte der Ontologie Design Patterns. Daher muss weiteres Wissen, beispielsweise aus externen Quellen, oder eine modifizierte TBox auf die entworfenen ODPs aufbauen beziehungsweise die Konzepte referenzieren (*RB4.0*). Die Durchführung aller Analysen benötigt gewisse Rechenressourcen. Dabei sind die CPU und der RAM relevante Größen, da die aktuell verfügbaren Bibliotheken noch keine Berechnungen auf der GPU unterstützen. In Tests hat sich ein Laptop mit 16GB RAM und einem Intel i5-8350U Prozessor für die Ausführung des Assistenzsystems als ausreichend herausgestellt. Die Verwendung von weniger als 8GB RAM hat die Analysen deutlich verlangsamt und die Identifikation von Templates nicht erlaubt. (*RB5.0*) Der Export des Wissens ist exemplarisch für diese Evaluierung implementiert worden und muss an das jeweilige Zielsystem für den Digitalen Zwilling angepasst werden. Daher wird auch keine Randbedingung für diese Funktionalität gestellt.

6.4.3 Abgleich mit den Eigenschaften von Brownfield-Produktionssystemen

Den Randbedingungen der Methodik aus dem letzten Kapitel werden in diesem Abschnitt die Eigenschaften von Brownfield-Produktionssystemen gegenübergestellt und die Erfüllung der Randbedingungen aufgezeigt.

Aktuelle Produktionssysteme wurden und werden nach übereinstimmenden Aussagen verschiedener Autoren für die Steuerung eingesetzt und sind aufgrund der Anlagenlebensdauer von mehreren Jahrzehnten auch noch länger präsent [54, 55, 190, 191]. Daher ist für diese Systeme in den meisten Fällen davon auszugehen, dass ein SPS-Code vorhanden ist (adressiert *RB1.0*). Die SPS-Hersteller setzen die IEC Norm 61131-3, die der Standard für die Programmierung jeder SPS ist [38, 41, 55], in ihren Entwicklungsumgebungen um und befolgen deren Struktur. Es ist jedoch nicht genau definiert, wann eine Software normgerecht ist [38], daraus resultieren syntaktische Varianten, beispielsweise bei der Verwendung von Führungszeichen [192]. Als Folge sind die Programme nicht zwangsläufig mit den Entwicklungsumgebungen anderer Hersteller kompatibel, jedoch sind die grundlegenden Strukturelemente dieselben und können daher in die Metamodellstruktur übertragen werden (adressiert *RB1.1*). Die Exportschnittstelle in den herstellereigenen Entwicklungsumgebungen zum Export des SPS-Codes als XML-Dateien, beispielsweise TIA Openness von Siemens, reduzieren die Varianz individueller Elemente im Code. Die IEC Norm 61131-10 [145]

definiert einen Standard für den XML-basierten Austausch von SPS-Code nach der IEC 61131-3 und erweitert dazu das PLCopen XML-Format, das auch im AutomationML Format eingesetzt wird. Aus diesem Grund ist eine weitere Annäherung des XML-Exports an den IEC-Standard und langfristig verfügbare XML-Exportoptionen zu erwarten (adressiert *RB1.2*).

Basierend auf dem technischen Fortschritt und der daraus steigenden Verbreitung von künstlicher Intelligenz werden vermehrt Daten durch die Anlagenbetreiber erfasst [193]. Diese erfassten Daten können eine erste Grundlage für die Datenbasis bilden oder sogar bereits ausreichend sein. Darüber hinaus ist eine Erfassung der benötigten Datenbasis aufgrund der geringen Größe in kurzer Zeit möglich und kann durch weitere Software automatisiert werden. Das Labeln der Positionsdaten kann durch Software nicht komplett automatisiert werden und bedarf einer manuellen Auswahl der Label beim Erfassungsprozess. Die Datenbasen der Betriebsdaten und Positionsdaten können daher kurzfristig erstellt werden oder sind eventuell schon vorhanden beziehungsweise zukünftig mit zunehmender Wahrscheinlichkeit vorhanden (adressiert *RB2.0 & RB 3.0*). Eine Einschränkung bildet die Randbedingung, dass die Betriebsdaten die Namen der SPS-Signale und einen Zeitstempel beinhalten müssen. Bestehende Datenbasen können prinzipiell diese Bedingung vernachlässigen, jedoch sind Zeitreihendaten der am weitesten verbreitete Datentyp in der Automatisierungsindustrie [149]. Zeitreihendaten beschreiben einen zeitlichen Verlauf eines Wertes und verfügen daher immer über einen Zeitstempel. Aufgrund der Verbreitung und der Relevanz für viele Anwendungen, wie die Prognose oder Analyse eines Verbrauchs [194], ist zu erwarten, dass Daten mit Zeitstempel verfügbar sind. Die Big Data-Charakteristik der Betriebsdaten hinsichtlich der Datenmenge und -geschwindigkeit macht den Einsatz von noSQL Datenbanken, wie Zeitreihendatenbanken, die automatisch einen Zeitstempel anfügen, für eine effiziente Datenspeicherung wahrscheinlich (*RB2.3* und *RB3.1* adressiert). Die Bereitstellung der Daten durch die SPS über OPC UA, wie im vorliegenden Anwendungsbeispiel oder über den Webserver der SPS an ein IT-System liefert die Signale mit demselben Namen wie im SPS-Code und erfüllt daher *RB2.1*. Eine Bereitstellung über andere Kommunikationskanäle, wie ein zusätzliches Edge-Gerät, bedarf einer Kontrolle der Benennung der Signale. Die Erfassung der Betriebsdaten erfolgt durch die SPS und wird bei der Planung angepasst an die Prozessgeschwindigkeit ausgelegt, da diese den Ablauf des technischen Prozesses steuert (adressiert *RB2.2*). Die Erfassungsgeschwindigkeit des Lokalisierungssystems hängt nicht zwangsläufig an der Geschwindigkeit des technischen Prozesses und wird vielmehr durch die Anwendung bestimmt, welche die Positionen benötigt. Dabei werden für viele Anwendungsfälle eines Lokalisierungssystems Intervalle von einer Sekunde und kürzer verwendet [138]. UBW-Systeme, die mit der TDOA-Methode arbeiten, können kostengünstig Erfassungsintervalle von unter 100 ms realisieren [15]. Dies ergibt bei einer maximalen Positionstoleranz von 15 cm zwischen der letzten erfassten Position und der Bewegung des Materials innerhalb eines Intervalls eine zulässige Materialhöchstgeschwindigkeit von 1,5 m/s. Für die Lokalisierungssysteme muss, abhängig von der Toleranz in der Positionsgenauigkeit, geprüft werden, ob das Erfassungsintervall auf Grund der Materialgeschwindigkeit verkürzt werden muss. Da das

Material zur Bearbeitung angehalten wird und mehrere Positionspunkte erfasst werden, ist die mittlere Positionsgenauigkeit höher als das eben berechnete Beispiel (adressiert *RB3.2*). Die kontinuierliche Erfassung der Betriebsdaten für Dokumentationszwecke, Big Data-Analysen und ähnlichen Applikationen erzeugt eine große Datenbasis mit Zeitbezug, der durch den, für die Methodik obligatorischen, Zeitstempel explizit festgehalten ist. Anhand des Zeitstempels kann das Training auf die neusten Datenpunkte beschränkt werden. Des Weiteren werden in dispositiven Datensystemen zur Historisierung der Daten teilweise zusätzliche Gültigkeitsfelder ergänzt, die eine Auswahl der verwendbaren Daten erleichtern. Durch eine Filterung anhand dieser Eigenschaften und über die Begrenzung der zeitlich sortierten Datenmenge kann das Verhältnis von aktuellen Daten zu unrelevanten, vergangenen Daten optimiert werden (adressiert *RB2.4*). Die Positionsdaten müssen im gleichen Zeitraum verfügbar sein und allen voran eine Auflösung besitzen, die eine korrekte Zuordnung der Komponenten ermöglicht. Die verfügbaren Lokalisierungssysteme beanspruchen eine Genauigkeit von zehn cm [195] oder in 85% der Fälle sogar eine Genauigkeit von unter 8 cm [15]. Die Lokalisierung durch die 5G Technologie kommt derzeit noch nicht in diese Auflösungsbereiche, jedoch ist für diese neue Technologie eine deutliche Verbesserung der Auflösung durch aktuelle Forschungen feststellbar [196]. Damit ist die Erfassung von Brownfield-Produktionssystemen bis zu dieser Strukturgröße möglich (adressiert *RB3.3*). Zuletzt bleibt die Synchronisation der Zeitstempel der Betriebs- und Positionsdaten, die abhängig vom Erfassungsweg der Daten ist. Für komplett autarke Systeme, die unabhängig voneinander Daten erfassen, muss dies beachtet werden. Sobald die Erfassung, wie in dieser Realisierung, in einer geteilten Zeitreihendatenbank abläuft, kann der Zeitstempel durch die Datenbank vergeben werden und ist dadurch automatisch synchron (adressiert *RB3.4*).

6.4.4 Bewertung der Übertragbarkeit

Die Randbedingungen (*RB1.0-RB3.3*) der Methodik decken sich mit den Eigenschaften von Brownfield-Produktionssystemen. Sie können daher in der Regel als erfüllt betrachtet werden. Darüber hinaus sind die vom Produktionssystem unabhängigen Randbedingungen (*RB4.0 und RB5.0*) noch offen, die bei der Anwendung der Methodik beziehungsweise bei der Nutzung des Assistenzsystems beachtet werden müssen. Diese betreffen die Hardware zur Ausführung des Assistenzsystems und mögliche Anpassungen am Wissensgraphen sowie der zugrundeliegenden Ontologie.

Die Anforderungen an die Hardware sind durch die deutlich überwiegende Anzahl der gängigen Computer und Laptops erfüllt, wie Statistiken zeigen [197–199]. Das Assistenzsystem wurde bei der Evaluierung auf einem Bürolaptop ohne besondere Hardwareausstattung und schwächeren Spezifikationen als der Durchschnitt der Geräte in den Statistiken ausgeführt. Die steigende Rechenleistung neuer Prozessoren und die sinkenden Preise für Arbeitsspeicher führen dazu, dass das Assistenzsystem zukünftig uneingeschränkt vom Gerät in tragbarer Dauer ausgeführt werden kann (adressiert *RB5.0*).

Die Randbedingung an weitere Quellen betrifft die genutzten ODPs, die bei der Einbringung zukünftiger Quellen verwendet werden müssen. Ebenso müssen diese bei der Erstellung einer neuen TBox für ein anderes System als Basis benutzt und referenziert werden. Diese hier beschriebenen Aufgaben betreffen die Erweiterung des Assistenzsystems beziehungsweise die optionale TBox. Das Assistenzsystem kann prinzipiell auch ohne diese Anpassungen betrieben werden, weshalb diese Randbedingung nicht essenziell für die Übertragbarkeit der Methodik ist.

Die Übertragung der Methodik auf größere Systeme hat des Weiteren Auswirkungen auf die Laufzeit des Assistenzsystems. Dabei muss eine Betrachtung pro Softwaremodul vorgenommen werden, da die Laufzeit jedes Moduls unterschiedlich skaliert:

- Die Laufzeit der SPS-Code-Analyse skaliert linear mit der Anzahl der SPS-Code-Elemente (FBs, DBs und IOs).
- Die Analyselaufzeit der RTLS und Betriebsdaten skaliert ebenfalls linear, da nicht für jeden Analyseschritt alle Datenpunkte herangezogen und in Bezug gesetzt, sondern jeweils feste Zeitfenster in der Zeitreihendatenbank betrachtet werden.
- Die Dauer der Graphanalyse zur Identifikation der sich wiederholenden Substrukturen als Templates skaliert deutlich nicht linear, sondern exponentiell mit der Anzahl der Knoten und Kanten im Wissensgraph.

Zur Verbesserung der Skalierbarkeit der Methodik beziehungsweise der Laufzeit des Software-Assistenzsystems bei größeren Produktionssystemen kann das Assistenzsystem auf einer leistungsstärkeren Hardware ausgeführt werden. Dabei wird zukünftig die Verwendung einer Grafikkarte zur Beschleunigung der Graphanalyse einen deutlichen Mehrwert liefern können, sobald entsprechende Implementierungen der Algorithmen verfügbar sind, welche die Grafikkarte(n) mit einbeziehen. Darüber hinaus kann das Assistenzsystem mit einer leistungsstärkeren Hardware in der Hinsicht noch optimiert werden, sodass die drei Analysen parallelisiert und damit Laufzeit eingespart werden kann. Die SPS-Code-Analyse ist komplett unabhängig von den Betriebs- und Positionsdatenanalysen. Diese beiden Analysen sind ebenfalls weitestgehend unabhängig bis auf die Nutzung des mit den Positionsdaten trainierten Klassifikator, um die Ergebnisse der Betriebsdatenanalyse zu gruppieren. Abgesehen davon ist zu überlegen, ob die Erstellung eines einzigen Digitalen Zwillings für ein großes System angestrebt werden muss oder ob dieses System eher in zwei oder mehr Digitale Zwillinge unterteilt werden sollte. Da die Methodik auf den SPS-Code aufbaut, ist die Unterteilung anhand der verbauten SPS in den verschiedenen Stationen eines Produktionssystems eine zweckmäßige, natürliche Untergliederung in mehrere Digitale Zwillinge. Zusammenfassend lässt sich festhalten, dass die Bedingungen für die Anwendung der Methodik auf weitere Brownfield-Produktionssysteme erfüllt sind.

7 Schlussbetrachtung

Dieses Kapitel fasst die wichtigsten Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf mögliche Erweiterungen der Forschungsergebnisse.

7.1 Zusammenfassung der Ergebnisse

In dieser Arbeit wurde die Erweiterung bestehender Produktionssysteme um einen Digitalen Zwilling betrachtet, sodass diese sogenannten Brownfield-Produktionssysteme unter den sich wandelnden Rahmenbedingungen der Industrie zukünftig effizient betrieben werden können. Dabei wurde das Problem identifiziert, dass Digitale Zwillinge für Brownfield-Produktionssysteme in der Industrie bisher noch in einem manuellen, zeitaufwendigen und fehleranfälligen Vorgang erstellt werden. Eine Betrachtung des gegenwärtigen Stands der Wissenschaft und Technik zeigte, dass wenige Ansätze zur automatisierten Erstellung von digitalen Modellen existierender Produktionssysteme vorhanden sind. Darunter sind vor allem viele Ansätze zur Erstellung dreidimensionaler Modelle (CAD-Modell). Jedoch werden die Relationen zwischen den Modellen eines Digitalen Zwillings bei den vorhandenen Ansätzen kaum beziehungsweise nicht berücksichtigt oder es werden umfangreiche Vorbedingungen gestellt, sodass diese nicht für die nachträgliche Erstellung der Relationen eines Digitalen Zwillings für ein Brownfield-Produktionssystem herangezogen werden können. Es fehlt zum gegenwärtigen Zeitpunkt eine Methodik, die automatisiert einen Digitalen Zwilling inklusive der Relationen zwischen den unterschiedlichen Modellen generiert, ohne auf formales Wissen über das Produktionssystem zurückzugreifen und den produktiven Betrieb während der Erstellung des Digitalen Zwillings einschränken zu müssen. Aus diesem Grund wurde in dieser Arbeit eine Methodik zur Erstellung der Relationen eines Digitalen Zwillings von Bestandssystemen in der Produktion entwickelt und als prototypisches Software-Assistenzsystem implementiert.

Die entwickelte Methodik erstellt, im Gegensatz zu den vorhandenen Ansätzen, automatisiert die Relationen zwischen den Modellen des Digitalen Zwillings, ohne weitreichende Vorbedingungen an das Brownfield-Produktionssystem und dessen Software zu stellen. Dies wurde am exemplarischen Brownfield-Produktionssystem des intelligenten Lagers in der ARENA2036 verdeutlicht. Dazu verwendet die Methodik drei unterschiedliche Quellen des bestehenden Brownfield-Produktionssystems, extrahiert durch Analysemethoden implizites Wissen über die Relationen zwischen den Modellen und führt dieses in einem Wissensgraph als Basis für die Erstellung des Digitalen Zwillings zusammen. Dieser Wissensgraph bildet die Relationen zwischen den Modellen des bestehenden Produktionssystems ab und wird als Grundlage für einen Digitalen Zwilling herangezogen. Zur Generierung des Wissens wurden drei Analysemethoden entwickelt. Jede dieser Analysen verarbeitet eine oder mehrere der Informationsquellen, um automatisiert die implizit enthaltenen Relationen zu extrahieren. Bei der *ersten Analyse* wird der SPS-Code in einer regelbasierten Strukturanalyse verarbeitet und die funktionalen

Abhängigkeiten und elektrischen Verbindungen des Produktionssystems herausgearbeitet. Diese bilden Eltern-Kind-Relationen und Instanz-Instanz-Relationen des Digitalen Zwillinges und werden in den Wissensgraphen gespeichert. Dazu wird der SPS-Code über ein Metamodell als Zwischenschritt in ein objektorientiertes, herstellerunabhängiges SPS-Code-Modell überführt, sodass die Analyse unabhängig vom nativen SPS-Code-Format ist. Dabei wird der Bezug der funktionalen Gruppen zu den SPS-Signalen und der dahinterstehenden Peripherie bewahrt. Die Signalnamen sind für die Verknüpfung des extrahierten Wissens mit den folgenden Analysen relevant. Die *zweite Analyse* ist datenbasiert und verarbeitet die Materialpositionen im Produktionssystem. Durch ein Clustering der Positionen werden zwei Ergebnisse erzielt. Zum einen können die Abmessungen der physischen Gruppen abgeschätzt und zum anderen kann der trainierte Algorithmus für die dritte Analyse verwendet werden. Als *dritte Analyse* wurde eine Korrelationsanalyse der Kombination der Materialpositionen und der Betriebsdaten entwickelt. Durch die Untersuchung der zeitlichen Korrelation von Signaländerungen in den Betriebsdaten mit der Materialposition werden die Positionen der Peripheriekomponenten abgeschätzt. Diese Positionen werden im Zuge der Analyse durch den Vorhersagealgorithmus der zweiten Analyse den physischen Gruppen zugeteilt. Die Informationen zur ungefähren Position werden zusammen mit der Gruppierung als Eltern-Kind-Relationen aus der geometrischen Perspektive in den Wissensgraphen gespeichert. Die Gesamtheit dieser drei Analysen kann ohne eine Unterbrechung des Produktionsbetriebs und ohne eine bestehende Wissensbasis automatisiert durchgeführt werden.

Die generierten Informationen werden basierend auf den Signalnamen zusammengeführt, mit Ontologien semantisch angereichert und im Wissensgraph abgespeichert. Dadurch wird das alleinstehende Wissen, das in den Analysen generiert wurde, in Bezug zueinander gebracht und für die Nutzung im Digitalen Zwilling bereitgestellt. Durch die allgemeinen Basisontologien, die auf Normen zu mechatronischen und cyber-physischen Systemen, der formalen Prozessbeschreibung und Digitalen Zwillingen basieren, ist die semantische Beschreibung auf andere Produktionssysteme übertragbar und das Wissen maschinell verarbeitbar. Der hierarchische Aufbau der Ontologie erlaubt es, bei Bedarf eine systemspezifische TBox für eine detailliertere Beschreibung zu ergänzen. Auf dem Wissensgraphen wurde eine weitere Methode zur Extraktion der häufig verwendeten Substrukturen implementiert, um wiederverwendbare digitale Engineering-Templates zu erstellen. Diese sind für Menschen aufgrund der Strukturgröße und Komplexität nur schwer oder fehlerhaft zu erkennen, wodurch die Fehleranfälligkeit des Prozesses gesenkt wurde. Darüber hinaus kann so die Wiederverwendbarkeit in zukünftigen Projekten und die Skalierbarkeit auf größere Systeme unterstützt werden, da eine maschinelle Identifikation der Templates zu objektiveren Ergebnissen kommt als durch die eingeschränkte menschliche Perspektive. Diese Templates sowie das dabei generierte Wissen bilden die Klassen-Instanz-Relationen ab.

Der Wissensgraph beinhaltet nach diesen drei Analysen und der Anreicherung im Graph das benötigte Wissen über die Relationen zwischen den Modellen des Digitalen Zwillings. Der Graph kann entweder abgefragt und visualisiert werden, sodass ein Ingenieur damit arbeiten kann, oder für die weitere Nutzung in ein Zielsystem überführt werden. In dieser Arbeit wurde ein Export im AutomationML-Format vorgesehen, das in vielen Softwaretools importiert werden kann. Alternativ kann dieses Modell des Assistenzsystems durch ein beliebiges Modul ausgetauscht oder ergänzt werden, um eine spezifische Schnittstelle zu bedienen oder um eine Datei in einem anderen Austauschformat zu erzeugen.

Die Evaluierung der Methodik hat gezeigt, dass das Forschungsziel, die Relationen eines Digitalen Zwillings für ein bestehendes Produktionssystem zu erstellen, erreicht wurde und dabei die Anforderungen erfüllt werden konnten. Die Erstellung der Relationen als integraler Bestandteil eines Digitalen Zwillings für bestehende Produktionssysteme konnte durch die Methodik automatisiert werden. Bei einer Gegenüberstellung des bisherigen manuellen Prozesses und der Anwendung der entwickelten Methodik ließ sich eine deutliche Zeiteinsparung von rund 60% beziehungsweise 70% gegenüber einem manuellen Prozess inklusive des geplanten Puffers für die Fehlerbeseitigung nachweisen. Durch die Methodik werden im Vergleich zum manuellen Vorgehen die Wiederverwendbarkeit der Templates verbessert, Fehlerquellen reduziert und dadurch die Hemmschwelle zur nachträglichen Erstellung eines Digitalen Zwillings für bestehende Produktionssysteme gesenkt. Eine Betrachtung der Randbedingungen der Methodik und ein Abgleich mit typischen Eigenschaften von Brownfield-Produktionssystemen untermauert den beispielhaften Übertrag vom Entwicklungssystem auf das Evaluierungssystem. Dadurch wurden die Anwendbarkeit und die Übertragbarkeit der Methodik gezeigt. Anlagenbetreiber profitieren von dieser Methodik neben der Zeitersparnis und Fehlerreduktion bei der Erstellung auch langfristig durch die erzielbaren Mehrwerte bei der Nutzung des generierten Digitalen Zwillings, beispielsweise durch Zeiteinsparungen aus der virtuellen Rekonfigurationsplanung oder der virtuellen Inbetriebnahme im Vergleich zum manuellen Vorgehen ohne Digitalen Zwillings. Dazu müssen über die Relationen hinaus ebenfalls die Modelle des Digitalen Zwillings erstellt werden, wozu bestehende Ansätze eingesetzt werden können. Dies ist jedoch kein Teil dieser Arbeit und wird im Ausblick thematisiert.

7.2 Ausblick

Die entwickelte Methodik wurde speziell für bestehende Produktionssysteme der Fertigungsindustrie entwickelt und fokussiert sich auf die bisher unbehandelten Relationen. Um den Digitalen Zwillings für konkrete Anwendungsfälle nutzbar zu machen, werden zusätzlich zu den Relationen weitere Komponenten, wie die verschiedenen Modelle aus den unterschiedlichen Fachbereichen oder historisierte Betriebsdaten benötigt (Kapitel 2.2.2 beinhaltet eine Übersicht der Komponenten). Für die Erstellung der Modelle können die Ansätze verwendet werden, die bereits in der wissenschaftlichen Literatur publiziert worden sind. Beispielsweise sind CAD-

Modelle des mechanischen Aufbaus des bestehenden Produktionssystems eine häufig benötigte Information, die durch Laserscans erstellt werden können. Diese Modelle können entweder separat abgelegt und im Wissensgraphen als Referenz ergänzt werden oder direkt im finalen Zielsystem in den Digitalen Zwilling integriert werden. Bei der Verwendung von AutomationML als Austauschformat zum Übertragen des Wissens aus dem Wissensgraphen in das Zielsystem, können die referenzierten Modelle ebenfalls direkt eingebunden und übertragen werden.

Durch die Integration weiterer Quellen und die Kombination der Modelle mit den in dieser Arbeit erstellten Relationen entsteht ein vollwertiger Digitaler Zwilling. Dessen Mehrwerte lassen sich in verschiedenen Anwendungen feststellen, die den Digitalen Zwilling als Grundlage verwenden, beispielsweise einem darauf aufbauenden flexiblen Rekonfigurationsmanagement. Des Weiteren kann dieser Digitale Zwilling durch zusätzliche Services erweitert werden, wie etwa durch eine dynamische Adaption der Modellgranularität, um anwendungsfallbezogen automatisch die passenden Modelle zu referenzieren.

Abgesehen vom weiteren Einsatz in Anwendungen und der Erweiterung des gewonnenen Wissens kann die Methodik selbst noch ergänzt werden. Eine mögliche Erweiterung ist die Integration einer zusätzlichen Methode, die vor oder während des Exports des Wissens in das Zielsystem für alle Signalnamen die Namenskonvention der Ankerpunktmethodik anwendet. Dazu muss zum einen die Struktur des Wissensgraphen, die IEC 81364 und die Regeln der Ankerpunktmethodik in die Benennung einfließen. Zum anderen muss weiteres Wissen über die Art der Peripherie, den Hersteller und den Nutzen im Produktionssystem aus einer externen Quelle (ein menschlicher Experte oder formalisiertes Wissen) eingebracht werden. Vielversprechend könnte dabei der Einsatz hybrider KI-Methoden sein, die wissens- und datenbasierte Ansätze kombinieren, um sowohl das formale Wissen als auch die Varianz durch die Freiheitsgrade der Benennung und Strukturierung eines Produktionssystems adressieren zu können. Durch die Berücksichtigung der Namenskonvention bei der Erstellung des Digitalen Zwillings kann anschließend die Ankerpunktmethodik genutzt werden, um zukünftig dessen Synchronität mit dem realen Produktionssystem zu gewährleisten.

Darüber hinaus ist eine Erweiterung der Analyse des SPS-Codes auf die textbasierten SPS-Sprachen oder Hochsprachen denkbar. Dadurch lässt sich die Anwendbarkeit der Methodik auf noch mehr Produktionssysteme ausweiten, um später auch verteilte Systeme ohne SPS analysieren zu können. Dabei können Deep-Learning-Architekturen wie Transformer voraussichtlich eine entscheidende Rolle spielen, wie der aktuelle Erfolg von GPT in der Softwareentwicklung zeigt.

Literaturverzeichnis

- [1] S. M. Kaplan: *Wiley electrical and electronics engineering dictionary*. Hoboken, N.J: John Wiley & Sons, 2004, DOI: 10.1109/9780470547151.
- [2] DIN SPEC 91345: *Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*, Apr. 2016.
- [3] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA), Hrsg., “Industrie 4.0: Begriffe”, VDI-Statusreport, 2022, [Online]. Verfügbar unter: <https://www.vdi.de/ueber-uns/presse/publikationen/details/industrie-40-begriffe-terms-and-definitions> (zuletzt geprüft am 26.03.2024).
- [4] B. Ashtari Talkhestani, T. Jung, B. Lindemann, N. Sahlab, N. Jazdi, W. Schloegl, und M. Weyrich, “An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System”, *at - Automatisierungstechnik*, Bd. 67, Nr. 9, S. 762–782, 2019, DOI: 10.1515/auto-2019-0039.
- [5] M. Weyrich, F. Steden, J. Wolf, und M. Scharf, “Identification of mechatronic units based on an example of a flexible customized multi lathe machine tool” in *16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Toulouse, France, 2011, S. 1–4, DOI: 10.1109/ETFA.2011.6059102.
- [6] M. Rauscher, “Agentenbasierte Konsistenzprüfung heterogener Modelle in der Automatisierungstechnik” Dissertation, Institut für Automatisierungstechnik und Softwaresysteme, Universität Stuttgart, Stuttgart, 2015.
- [7] VDI 3633: *Simulation von Logistik-, Materialfluss- und Produktionssystemen: Begriffe*, Mai. 2018.
- [8] K. Alexopoulos und G. Chryssolouris, “Process” in *CIRP encyclopedia of production engineering*, L. Laperrière, G. Reinhart, S. Chatti, und T. Tolio, Hrsg., 1st ed., Berlin, Heidelberg: Springer, 2014, S. 973–975, DOI: 10.1007/978-3-642-20617-7_6567.
- [9] J. Matevska: *Rekonfiguration komponentenbasierter Softwaresysteme zur Laufzeit*. Wiesbaden: Vieweg+Teubner, 2010, DOI: 10.1007/978-3-8348-9780-0.
- [10] C. Diedrich, T. Hadlich, und M. Thron, “Semantik durch Merkmale für Industrie 4.0” in *Handbuch Industrie 4.0 Bd.2: Automatisierung*, B. Vogel-Heuser, T. Bauernhansl, und M. ten Hompel, Hrsg., 2nd ed., Berlin: Springer Vieweg, 2017, S. 417–432, DOI: 10.1007/978-3-662-53248-5_69.
- [11] S. S. Kamble, A. Gunasekaran, H. Parekh, V. Mani, A. Belhadi, und R. Sharma, “Digital twin for sustainable manufacturing supply chains: Current trends, future perspectives, and an implementation framework”, *Technological Forecasting and Social Change*, Bd. 176, S. 121448, 2022, DOI: 10.1016/j.techfore.2021.121448.
- [12] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, und M. Hoffmann, “Industrie 4.0”, *Wirtschaftsinf*, Bd. 56, Nr. 4, S. 261–264, 2014, DOI: 10.1007/s11576-014-0424-4.
- [13] S. Biffel, D. Gerhard, und A. Lüder, “Introduction to the Multi-Disciplinary Engineering for Cyber-Physical Production Systems” in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, S. Biffel, A. Lüder, und D. Gerhard, Hrsg., Cham: Springer International Publishing, 2017, S. 1–24, DOI: 10.1007/978-3-319-56345-9_1.
- [14] F. Mhenni, J.-Y. Choley, O. Penas, R. Plateaux, und M. Hammadi, “A SysML-based methodology for mechatronic systems architectural design”, *Advanced Engineering Informatics*, Bd. 28, Nr. 3, S. 218–231, 2014, DOI: 10.1016/j.aei.2014.03.006.
- [15] A. Frankó, G. Vida, und P. Varga, “Reliable Identification Schemes for Asset and Production Tracking in Industry 4.0”, *Sensors (Basel, Switzerland)*, Bd. 20, Nr. 13, 2020, DOI: 10.3390/s20133709.
- [16] M. Weyrich, M. Klein, J.-P. Schmidt, N. Jazdi, K. D. Bettenhausen, F. Buschmann, C. Rubner, M. Pirker, und K. Wurm, “Evaluation Model for Assessment of Cyber-Physical Production Systems” in *Springer Series in Wireless Technology, Industrial Internet of Things: Cybermanufacturing Systems*, S. Jeschke, C. Brecher, H. Song, und D. B. Rawat, Hrsg., Cham: Springer International Publishing, 2017, S. 169–199, DOI: 10.1007/978-3-319-42559-7_7.
- [17] J. Ladiges, “Automatisierte Bestimmung von Eigenschaften industrieller Produktionssysteme unter Einfluss evolutionärer Änderungen” Dissertation, Institut für Automatisierungstechnik, Helmut-Schmidt-Universität, Hamburg, 2018.

- [18] E. Westkämper und C. Löffler: *Strategien der Produktion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, DOI: 10.1007/978-3-662-48914-7.
- [19] R. Drath, C. Mosch, S. Hoppe, A. Faath, E. Barnstedt, B. Fiebigler, und W. Schlögl, “Diskussionspapier – Interoperabilität mit der Verwaltungsschale, OPC UA und AutomationML: Zielbild und Handlungsempfehlungen für industrielle Interoperabilität”, AutomationML e.V., IDTA, OPC Foundation, VDMA, Apr. 2023, [Online]. Verfügbar unter: <https://www.vdma.org/documents/34570/78243663/Diskussionspapier-Zielbild-und-Handlungsempfehlungen-fur-industrielle-Interoperabilitat-5.3-protected.pdf> (zuletzt geprüft am 11.09.2023).
- [20] H. Kagermann, W. Wahlster, und J. Helbig, “Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Deutschlands Zukunft als Produktionsstandort sichern ; Abschlussbericht des Arbeitskreises Industrie 4.0”, [Online]. Verfügbar unter: <https://www.acatech.de/publikation/umsetzungsempfehlungen-fuer-das-zukunftsprojekt-industrie-4-0-abschlussbericht-des-arbeitskreises-industrie-4-0/download-pdf> (zuletzt geprüft am 14.08.2023).
- [21] M. Schamp, S. Hoedt, A. Claeys, E. H. Aghezzaf, und J. Cottyn, “Impact of a virtual twin on commissioning time and quality”, *IFAC-PapersOnLine*, Bd. 51, Nr. 11, S. 1047–1052, 2018, DOI: 10.1016/j.ifacol.2018.08.469.
- [22] W. Kritzingler, M. Karner, G. Traar, J. Henjes, und W. Sihn, “Digital Twin in manufacturing: A categorical literature review and classification”, *IFAC-PapersOnLine*, Bd. 51, Nr. 11, S. 1016–1022, 2018, DOI: 10.1016/j.ifacol.2018.08.474.
- [23] S. Boschert und R. Rosen, “Digital Twin—The Simulation Aspect” in *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers*, P. Hehenberger und D. Bradley, Hrsg., Switzerland: Springer, 2016, S. 59–74, DOI: 10.1007/978-3-319-32156-1_5.
- [24] E. Negri, L. Fumagalli, und M. Macchi, “A Review of the Roles of Digital Twin in CPS-based Production Systems”, *Procedia Manufacturing*, Bd. 11, S. 939–948, 2017, DOI: 10.1016/j.promfg.2017.07.198.
- [25] A. Chiarini, V. Belvedere, und A. Grando, “Industry 4.0 strategies and technological developments. An exploratory research from Italian manufacturing companies”, *Production Planning & Control*, Bd. 31, Nr. 16, S. 1385–1398, 2020, DOI: 10.1080/09537287.2019.1710304.
- [26] M. Bajzek, J. Fritz, H. Hick, M. Maletz, C. Faustmann, und G. Stieglbauer, “Model Based Systems Engineering Concepts” in *Powertrain, Systems Engineering for Automotive Powertrain Development*, H. Hick, K. Küpper, und H. Sorger, Hrsg., 1st ed., Cham: Springer International Publishing, 2020, S. 195–234, DOI: 10.1007/978-3-319-99629-5_8.
- [27] S. Alker, V. Joy, P. Roberts, und N. Smith, “The Definition of Brownfield”, *Journal of Environmental Planning and Management*, Bd. 43, Nr. 1, S. 49–69, 2000, DOI: 10.1080/09640560010766.
- [28] A. Vaughn, P. Fernandes, und J. T. Shields, “Manufacturing system design framework manual”, S. 1–64, 2002, [Online]. Verfügbar unter: <http://hdl.handle.net/1721.1/81902>.
- [29] P. O’Donovan, K. Leahy, K. Bruton, und D. T. J. O’Sullivan, “An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities”, *Journal of Big Data*, Bd. 2, Nr. 1, S. 1–26, 2015, DOI: 10.1186/s40537-015-0034-z.
- [30] T. Lehtonen, J. Pakkanen, M. Lanz, J. Järvenpää, und R. Tuokko, “Brownfield process for developing of product families” in *Proceedings of the 18th International Conference on Engineering Design*, Copenhagen, Denmark, 2011, S. 248–257.
- [31] S. R. Bader, C. Wolff, M. Vössing, und J.-P. Schmidt, “Towards Enabling Cyber-Physical Systems in Brownfield Environments” in *Lecture Notes in Business Information Processing*, Bd. 331, *Exploring Service Science (IESS): 9th International Conference*, G. Satzger, L. Patricio, M. Zaki, N. Kühl, und P. Hottum, Hrsg., Cham: Springer International Publishing, 2018, S. 165–176, DOI: 10.1007/978-3-030-00713-3_13.
- [32] M. Strube, “Modellgestützte Modernisierungsplanung industrieller Automatisierungslösungen” Dissertation, Institut für Automatisierungstechnik, Helmut-Schmidt-Universität, Hamburg, 2014.
- [33] A. Strahilov und H. Hämmerle, “Engineering Workflow and Software Tool Chains of Automated Production Systems” in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, S. Biffl, A. Lüder, und D. Gerhard, Hrsg., Cham: Springer International Publishing, 2017, S. 207–234.

- [34] B. Drescher, P. Stich, J. Kiefer, A. Strahilov, T. Bär, und G. Reinhart, “Physikbasierte Simulation im Anlagentstehungsprozess: Einsatzpotenziale bei der Entwicklung automatisierter Montageanlagen im Automobilbau” in *15. ASIM Fachtagung Simulation in Produktion und Logistik: Entscheidungsunterstützung von der Planung bis zur Steuerung*, Paderborn, Germany, 2013.
- [35] S. Biffl, F. Ekaputra, A. Luder, J. Pauly, F. Rinker, L. Waltersdorfer, und D. Winkler, “Technical Debt Analysis in Parallel Multi-Disciplinary Systems Engineering” in *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Kallithea-Chalkidiki, Greece, 2019, S. 342–346, DOI: 10.1109/SEAA.2019.00059.
- [36] S. Biffl, E. Maetzler, M. Wimmer, A. Lueder, und N. Schmidt, “Linking and versioning support for AutomationML: A model-driven engineering perspective” in *13th IEEE International Conference on Industrial Informatics (INDIN)*, Cambridge, United Kingdom, 2015, S. 499–506, DOI: 10.1109/INDIN.2015.7281784.
- [37] K. Paetzold, “Product and Systems Engineering/CA* Tool Chains” in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, S. Biffl, A. Lüder, und D. Gerhard, Hrsg., Cham: Springer International Publishing, 2017, S. 27–62, DOI: 10.1007/978-3-319-56345-9_2.
- [38] J. Kiefer, “Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau” Dissertation, Lehrstuhl für Fertigungstechnik/CAM, Universität des Saarlandes, Saarbrücken, 2007.
- [39] P. Marks, “Softwaregestütztes Assistenzkonzept zur Modernisierung von Automatisierungssystemen” Dissertation, Institut für Automatisierungstechnik und Softwaresysteme, Universität Stuttgart, Stuttgart, 2020.
- [40] M. Bajzek, J. Fritz, und H. Hick, “Systems Engineering Principles” in *Powertrain, Systems Engineering for Automotive Powertrain Development*, H. Hick, K. Küpper, und H. Sorger, Hrsg., 1st ed., Cham: Springer International Publishing, 2020, S. 149–194, DOI: 10.1007/978-3-319-99629-5_7.
- [41] B. Vogel-Heuser, A. Fay, I. Schaefer, und M. Tichy, “Evolution of software in automated production systems: Challenges and research directions”, *Journal of Systems and Software*, Bd. 110, S. 54–84, 2015, DOI: 10.1016/j.jss.2015.08.026.
- [42] M. Bellgran: *Production Development: Design and Operation of Production Systems*. London: Springer London, 2010, DOI: 10.1007/978-1-84882-495-9.
- [43] M. Borth, J. Verriet, und G. Muller, “Digital Twin Strategies for SoS 4 Challenges and 4 Architecture Setups for Digital Twins of SoS” in *14th Annual Conference System of Systems Engineering (SoSE)*, Anchorage, AK, USA, 2019, S. 164–169, DOI: 10.1109/SYBOSE.2019.8753860.
- [44] M. Foehr, J. Vollmar, A. Calà, P. Leitão, S. Karnouskos, und A. W. Colombo, “Engineering of Next Generation Cyber-Physical Automation System Architectures” in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, S. Biffl, A. Lüder, und D. Gerhard, Hrsg., Cham: Springer International Publishing, 2017, S. 185–206, DOI: 10.1007/978-3-319-56345-9_8.
- [45] A. R. Hevner, S. T. March, J. Park, und S. Ram, “Design Science in Information Systems Research”, *MIS Quarterly*, Bd. 28, Nr. 1, S. 75–105, 2004, DOI: 10.2307/25148625.
- [46] M. Steinbuch, “Mechatronics Disrupted” in *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers*, P. Hehenberger und D. Bradley, Hrsg., Switzerland: Springer, 2016, S. 17–24, DOI: 10.1007/978-3-319-32156-1_2.
- [47] P. Hehenberger, T. J. Howard, und J. Torry-Smith, “From Mechatronic Systems to Cyber-Physical Systems: Demands for a New Design Methodology?” in *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers*, P. Hehenberger und D. Bradley, Hrsg., Switzerland: Springer, 2016, S. 147–163, DOI: 10.1007/978-3-319-32156-1_10.
- [48] M. Weyrich: *Industrielle Automatisierungs- und Informationstechnik: IT-Architekturen, Kommunikation und Software zur Systemgestaltung*, 1. Aufl. Berlin, Heidelberg: Springer Vieweg, 2023, DOI: 10.1007/978-3-662-56355-7.
- [49] VDI/VDE-Richtlinie 2206: *Entwicklung mechatronischer und cyber-physischer Systeme*, Nov. 2021.
- [50] T. Müller, B. Lindemann, T. Jung, N. Jazdi, und M. Weyrich, “Enhancing an Intelligent Digital Twin with a Self-organized Reconfiguration Management based on Adaptive Process Models”, *Procedia CIRP*, Bd. 104, S. 786–791, 2021, DOI: 10.1016/j.procir.2021.11.132.

- [51] A. Scholz, “Unterstützung des Engineerings von fertigungstechnischen Produktionssystemen mit Hilfe von Maschinenfunktionen: Methode, Modell und Beschreibungsmittel für ein funktionsorientiertes Planen” Dissertation, Institut für Automatisierungstechnik, Helmut-Schmidt-Universität, Hamburg, 2018.
- [52] D. Behnen, H. Mersch, C. Quix, D. Schmitz, M. Zhang, K. Fayzullin, C. Brecher, U. Epple, und M. Jarke, “Gemeinsamkeiten und Unterschiede in der Modellierung von prozesstechnischen und diskreten Produktionsanlagen”, *EKA*, S. 1–10, 2010.
- [53] K. D. Listmann, P. Wenzelburger, und F. Allgöwer, “Industrie 4.0 – (R)evolution ohne Regelungstechnik?”, *at - Automatisierungstechnik*, Bd. 64, Nr. 7, 2016, DOI: 10.1515/auto-2016-0039.
- [54] F. Biesinger und M. Weyrich, “The Facets of Digital Twins in Production and the Automotive Industry” in *23rd International Conference on Mechatronics Technology (ICMT)*, Salerno, Italy, 2019, S. 1–6, DOI: 10.1109/ICMECT.2019.8932101.
- [55] B. Vogel-Heuser, J. Fischer, S. Feldmann, S. Ulewicz, und S. Rösch, “Modularity and architecture of PLC-based software for automated production Systems: An analysis in industrial companies”, *Journal of Systems and Software*, Bd. 131, S. 35–62, 2017, DOI: 10.1016/j.jss.2017.05.051.
- [56] K. Thramboulidis, “IEC 61499 as an Enabler of Distributed and Intelligent Automation: A State-of-the-Art Review—A Different View”, *Journal of Engineering*, Bd. 2013, S. 1–9, 2013, DOI: 10.1155/2013/638521.
- [57] K. D. Bettenhausen und S. Kowalewski, “Cyber-physical systems: Chancen und Nutzen aus Sicht der Automation”, VDI/VDE-GMA, 2013, [Online]. Verfügbar unter: <https://www.vdi.de/ueber-uns/presse/publikationen/details/cyber-physical-systems-chancen-und-nutzen-aus-sicht-der-automation> (zuletzt geprüft am 26.03.2024).
- [58] S. Jeschke, C. Brecher, H. Song, und D. B. Rawat, Hrsg.: *Industrial Internet of Things: Cybermanufacturing Systems*. Cham: Springer International Publishing, 2017.
- [59] D. Pantförder, F. Mayer, C. Diedrich, P. Göhner, M. Weyrich, und B. Vogel-Heuser, “Agentenbasierte dynamische Rekonfiguration von vernetzten intelligenten Produktionsanlagen – Evolution statt Revolution” in *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien, Migration*, T. Bauernhansl, M. ten Hompel, und B. Vogel-Heuser, Hrsg., Wiesbaden: Springer Vieweg, 2014, S. 145–158, DOI: 10.1007/978-3-658-04682-8_7.
- [60] R. Isermann: *Mechatronische Systeme: Grundlagen*, 2. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, DOI: 10.1007/978-3-540-32512-3.
- [61] B. Vogel-Heuser, M. Böhm, F. Brodeck, K. Kugler, S. Maasen, D. Pantförder, M. Zou, J. Buchholz, H. Bauer, F. Brandl, und U. Lindemann, “Interdisciplinary engineering of cyber-physical production systems: highlighting the benefits of a combined interdisciplinary modelling approach on the basis of an industrial case”, *Design Science*, Nr. 6, 2020, DOI: 10.1017/dsj.2020.2.
- [62] R. Haberfellner, O. L. de Weck, und E. Fricke: *Systems engineering: Fundamentals and applications*. Cham: Springer International Publishing, 2019, DOI: 10.1007/978-3-030-13431-0.
- [63] S. Sierla, M. Azangoo, K. Rainio, N. Papakonstantinou, A. Fay, P. Honkamaa, und V. Vyatkin, “Roadmap to semi-automatic generation of digital twins for brownfield process plants”, *Journal of Industrial Information Integration*, Bd. 27, Nr. 100282, 2022, DOI: 10.1016/j.jii.2021.100282.
- [64] J. Kiefer, S. Allegretti, und T. Breckle, “Quality- and Lifecycle-oriented Production Engineering in Automotive Industry”, *Procedia CIRP*, Bd. 62, S. 446–451, 2017, DOI: 10.1016/j.procir.2016.06.086.
- [65] M. W. Grieves, “Product lifecycle management: the new paradigm for enterprises”, *International Journal of Product Development*, Bd. 2, 1/2, S. 71–84, 2005, DOI: 10.1504/IJPD.2005.006669.
- [66] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, und L. Wang, “Modeling, Simulation, Information Technology & Processing Roadmap: Technology Area 11”, Washington, [Online]. Verfügbar unter: https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf (zuletzt geprüft am 28.12.2018).
- [67] M. Grieves: *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*, 2015.
- [68] M. Singh, E. Fuenmayor, E. Hinchy, Y. Qiao, N. Murray, und D. Devine, “Digital Twin: Origin to Future”, *Applied System Innovation*, Bd. 4, Nr. 2, S. 36, 2021, DOI: 10.3390/asi4020036.

- [69] M. Sjarov, T. Lechler, J. Fuchs, M. Brossog, A. Selmaier, F. Faltus, T. Donhauser, und J. Franke, “The Digital Twin Concept in Industry: A Review and Systematization” in *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, 2020, S. 1789–1796, DOI: 10.1109/ETFA46521.2020.9212089.
- [70] G. N. Schroeder, C. Steinmetz, C. E. Pereira, und D. B. Espindola, “Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange”, *IFAC-PapersOnLine*, Bd. 49, Nr. 30, S. 12–17, 2016, DOI: 10.1016/j.ifacol.2016.11.115.
- [71] T. Müller, N. Sahlab, S. Kamm, C. Köhler, D. Braun, N. Jazdi, und M. Weyrich, “Context-enriched modeling using Knowledge Graphs for intelligent Digital Twins of Production Systems” in *27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Stuttgart, Germany, 2022, S. 1–8, DOI: 10.1109/ETFA52439.2022.9921615.
- [72] T. Kuhn, “Digitaler Zwilling”, *Informatik-Spektrum*, Bd. 40, Nr. 5, S. 440–444, 2017, DOI: 10.1007/s00287-017-1061-2.
- [73] R. Rosen, G. von Wichert, G. Lo, und K. D. Bettenhausen, “About The Importance of Autonomy and Digital Twins for the Future of Manufacturing”, *IFAC-PapersOnLine*, Bd. 48, Nr. 3, S. 567–572, 2015, DOI: 10.1016/j.ifacol.2015.06.141.
- [74] F. Tao, H. Zhang, A. Liu, und A. Y. C. Nee, “Digital Twin in Industry: State-of-the-Art”, *IEEE Transactions on Industrial Informatics*, Bd. 15, Nr. 4, S. 2405–2415, 2019, DOI: 10.1109/TII.2018.2873186.
- [75] L. Lattanzi, R. Raffaelli, M. Peruzzini, und M. Pellicciari, “Digital twin for smart manufacturing: a review of concepts towards a practical industrial implementation”, *International Journal of Computer Integrated Manufacturing*, Bd. 34, Nr. 6, S. 567–597, 2021, DOI: 10.1080/0951192X.2021.1911003.
- [76] D. Jones, C. Snider, A. Nassehi, J. Yon, und B. Hicks, “Characterising the Digital Twin: A systematic literature review”, *CIRP Journal of Manufacturing Science and Technology*, Bd. 29, S. 36–52, 2020, DOI: 10.1016/j.cirpj.2020.02.002.
- [77] S. Malakuti und S. Grüner, “Architectural aspects of digital twins in IIoT systems” in *Proceedings of the 12th European Conference on Software Architecture: Companion proceedings*, Madrid, Spain, 2018, S. 1–2, DOI: 10.1145/3241403.3241417.
- [78] D. Dittler, D. Braun, T. Müller, V. Stegmaier, N. Jazdi, und M. Weyrich, “A procedure for the derivation of project-specific intelligent Digital Twin implementations in industrial automation” in *17. Fachtagung - Entwurf Komplexer Automatisierungssysteme (EKA): Beschreibungsmittel, Methoden, Werkzeuge und Anwendungen*, 2022.
- [79] E. Järvenpää, N. Siltala, und M. Lanz, “Formal resource and capability descriptions supporting rapid reconfiguration of assembly systems” in *IEEE International Symposium on Assembly and Manufacturing (ISAM)*, Fort Worth, TX, 2016, S. 120–125, DOI: 10.1109/ISAM.2016.7750724.
- [80] T. Müller, B. Caesar, M. Weiß, S. Ferhat, N. Sahlab, A. Fay, R. Oger, N. Jazdi, und M. Weyrich, “Reconfiguration management in manufacturing”, *at - Automatisierungstechnik*, Bd. 71, Nr. 5, S. 330–350, 2023, DOI: 10.1515/auto-2022-0139.
- [81] B. Lindemann, N. Jazdi, und M. Weyrich, “Anomaly detection and prediction in discrete manufacturing based on cooperative LSTM networks” in *16th IEEE International Conference on Automation Science and Engineering (CASE)*, Hong Kong, 2020, S. 1003–1010, DOI: 10.1109/CASE48305.2020.9216855.
- [82] D. Dittler, P. Lierhammer, D. Braun, T. Müller, N. Jazdi, und M. Weyrich, “A Novel Model Adaption Approach for intelligent Digital Twins of Modular Production Systems” in *28th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sinaia, Romania, 2023.
- [83] F. Jaensch, A. Csiszar, C. Scheifele, und A. Verl, “Digital Twins of Manufacturing Systems as a Base for Machine Learning” in *Proceedings of the 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Stuttgart, 2018, S. 1–6, DOI: 10.1109/M2VIP.2018.8600844.
- [84] DIN 62714-1: *Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme - Automation markup language: Teil 1: Architektur und allgemeine Festlegungen*, Feb. 2019.
- [85] R. Drath und M. Schleipen, “Grundarchitektur: das Objektmodell” in *VDI-Buch, Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*, R. Drath, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 45–94, DOI: 10.1007/978-3-642-04674-2_2.

- [86] A. Fuller, Z. Fan, C. Day, und C. Barlow, “Digital twin: Enabling technologies, challenges and open research”, *IEEE Access*, Nr. 8, S. 108952–108971, 2020, DOI: 10.1109/ACCESS.2020.2998358.
- [87] R. van Dinter, B. Tekinerdogan, und C. Catal, “Predictive maintenance using digital twins: A systematic literature review”, *Information and Software Technology*, Bd. 151, S. 107008, 2022, DOI: 10.1016/j.infsof.2022.107008.
- [88] VDI/VDE-Richtlinie 3693 Blatt 2: *Virtuelle Inbetriebnahme: Einführung der virtuellen Inbetriebnahme in Unternehmen*, Dez. 2018.
- [89] VDI/VDE-Richtlinie 3693 Blatt 1: *Virtuelle Inbetriebnahme: Modellarten und Glossar*, Aug. 2016.
- [90] G. E. Modoni, E. G. Caldarola, M. Sacco, und W. Terkaj, “Synchronizing physical and digital factory: benefits and technical challenges”, *Procedia CIRP*, Bd. 79, S. 472–477, 2019, DOI: 10.1016/j.procir.2019.02.125.
- [91] J. Michael, J. Pfeiffer, B. Rumpe, und A. Wortmann, “Integration Challenges for Digital Twin Systems-of-Systems” in *Proceedings of the 10th IEEE/ACM International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems (SESoS)*, Pittsburgh Pennsylvania, 2022, S. 9–12, DOI: 10.1145/3528229.3529384.
- [92] P. Bibow, M. Dalibor, C. Hopmann, B. Mainz, B. Rumpe, D. Schmalzing, M. Schmitz, und A. Wortmann, “Model-Driven Development of a Digital Twin for Injection Molding” in *Proceedings CAiSE: International Conference on Advanced Information Systems Engineering*, Bd. 12127, *Advanced Information Systems Engineering: 32nd International Conference, CAiSE 2020, Proceedings*, S. Dustdar, E. Yu, C. Salinesi, D. Rieu, und V. Pant, Hrsg., 1st ed., Cham: Springer International Publishing; Imprint Springer, 2020, S. 85–100, DOI: 10.1007/978-3-030-49435-3_6.
- [93] S. Sierla, L. Sorsamäki, M. Azangoo, A. Villberg, E. Hytönen, und V. Vyatkin, “Towards Semi-Automatic Generation of a Steady State Digital Twin of a Brownfield Process Plant”, *Applied Sciences*, Bd. 10, Nr. 19, S. 6959, 2020, DOI: 10.3390/app10196959.
- [94] M. Azangoo, L. Sorsamaki, S. A. Sierla, T. Matasniemi, M. Rantala, K. Rainio, und V. Vyatkin, “A Methodology for Generating a Digital Twin for Process Industry: A Case Study of a Fiber Processing Pilot Plant”, *IEEE Access*, Bd. 10, S. 58787–58810, 2022, DOI: 10.1109/ACCESS.2022.3178424.
- [95] D. Guerra-Zubiaga, V. Kuts, K. Mahmood, A. Bondar, N. Nasajpour-Esfahani, und T. Otto, “An approach to develop a digital twin for industry 4.0 systems: manufacturing automation case studies”, *International Journal of Computer Integrated Manufacturing*, Bd. 34, Nr. 9, S. 933–949, 2021, DOI: 10.1080/0951192X.2021.1946857.
- [96] C. Assawaarayakul, W. Srisawat, S. D. N. Ayuthaya, und S. Wattanasirichaigoon, “Integrate Digital Twin to Exist Production System for Industry 4.0” in *4th IEEE Technology Innovation Management and Engineering Science International Conferenc (TIMES-iCON)*, Bangkok, Thailand, 2019, S. 1–5, DOI: 10.1109/TIMES-iCON47539.2019.9024430.
- [97] F. Biesinger, D. Meike, B. Kraß, und M. Weyrich, “A digital twin for production planning based on cyber-physical systems: A Case Study for a Cyber-Physical System-Based Creation of a Digital Twin”, *Procedia CIRP*, Bd. 79, S. 355–360, 2019, DOI: 10.1016/j.procir.2019.02.087.
- [98] M. Sommer, J. Stjepandić, S. Stobrawa, und M. von Soden, “Automated Generation of a Digital Twin of a Manufacturing System by Using Scan and Convolutional Neural Networks” in *Advances in Transdisciplinary Engineering*, Bd. 12, *Transdisciplinary Engineering for Complex Socio-Technical Systems - Real-life Applications: Proceedings of the 27th ISTE International Conference on Transdisciplinary Engineering*, J. Pokojski, M. Gil, L. Newnes, J. Stjepandic, und N. Wognum, Hrsg.: IOS Press Incorporated, 2020, S. 363–372, DOI: 10.3233/ATDE200095.
- [99] F. Buonamici, M. Carfagni, R. Furferi, L. Governi, A. Lapini, und Y. Volpe, “Reverse engineering modeling methods and tools: a survey”, *Computer-Aided Design and Applications*, Bd. 15, Nr. 3, S. 443–464, 2018, DOI: 10.1080/16864360.2017.1397894.
- [100] Z. M. Bi und L. Wang, “Advances in 3D data acquisition and processing for industrial applications”, *Robotics and Computer-Integrated Manufacturing*, Bd. 26, Nr. 5, S. 403–413, 2010, DOI: 10.1016/j.rcim.2010.03.003.
- [101] E. Lindskog, J. Vallhagen, und B. Johansson, “Production system redesign using realistic visualisation”, *International Journal of Production Research*, Bd. 55, Nr. 3, S. 858–869, 2017, DOI: 10.1080/00207543.2016.1218085.

- [102] J. Li, J. Berglund, F. Auris, A. Hanna, J. Vallhagen, and K. Akesson, “Evaluation of Photogrammetry for Use in Industrial Production Systems” in *14th IEEE International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, 2018, S. 414–420, DOI: 10.1109/COASE.2018.8560496.
- [103] A. Petruccioli, F. Gherardini, and F. Leali, “Assessment of close-range photogrammetry for the low cost development of 3D models of car bodywork components”, *International Journal on Interactive Design and Manufacturing (IJIDeM)*, Bd. 16, Nr. 2, S. 703–713, 2022, DOI: 10.1007/s12008-022-00865-6.
- [104] G. Erdős, T. Nakano, and J. Váncza, “Adapting CAD models of complex engineering objects to measured point cloud data”, *CIRP Annals*, Bd. 63, Nr. 1, S. 157–160, 2014, DOI: 10.1016/j.cirp.2014.03.090.
- [105] G. Erdős, T. Nakano, G. Horváth, Y. Nonaka, and J. Váncza, “Recognition of complex engineering objects from large-scale point clouds”, *CIRP Annals*, Bd. 64, Nr. 1, S. 165–168, 2015, DOI: 10.1016/j.cirp.2015.04.026.
- [106] R. Stark, H. Grosser, and P. Müller, “Product analysis automation for digital MRO based on intelligent 3D data acquisition”, *CIRP Annals*, Bd. 62, Nr. 1, S. 123–126, 2013, DOI: 10.1016/j.cirp.2013.03.079.
- [107] H. Son, C. Kim, and C. Kim, “3D reconstruction of as-built industrial instrumentation models from laser-scan data and a 3D CAD database based on prior knowledge”, *Automation in Construction*, Bd. 49, S. 193–200, 2015, DOI: 10.1016/j.autcon.2014.08.007.
- [108] G. Koltun, M. Kolter, and B. Vogel-Heuser, “Automated Generation of Modular PLC Control Software from P&ID Diagrams in Process Industry” in *4th IEEE International Symposium on Systems Engineering (ISSE)*, Rome, 2018, S. 1–8, DOI: 10.1109/SysEng.2018.8544401.
- [109] G. D. Koltun, S. Feldmann, D. Schutz, and B. Vogel-Heuser, “Model-document coupling in aPS engineering: Challenges and requirements engineering use case” in *18th IEEE International Conference on Industrial Technology (ICIT)*, Toronto, ON, 2017, S. 1177–1182, DOI: 10.1109/ICIT.2017.7915529.
- [110] G. Koltun, F. Maurer, A. Knoll, E. Trunzer, and B. Vogel-Heuser, “Information Retrieval from Redlined Circuit Diagrams and its Model-Based Representation for Automated Engineering” in *Proceedings IECON 2018: 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington D.C., USA, 2018, S. 3114–3119, DOI: 10.1109/IECON.2018.8591443.
- [111] G. Koltun, M. R. Basirati, M. Subhan Hammeed, M. Bohm, H. Krcmar, and B. Vogel-Heuser, “Reverse Engineering on changed Functional Specification Documents for Model-Based Requirements Engineering” in *2nd IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, Taipei, Taiwan, 2019, S. 687–692, DOI: 10.1109/ICPHYS.2019.8780310.
- [112] A. Khan, M. Dahl, P. Falkman, and M. Fabian, “Digital Twin for Legacy Systems: Simulation Model Testing and Validation” in *14th IEEE International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, 2018, S. 421–426, DOI: 10.1109/COASE.2018.8560338.
- [113] F. Biesinger, D. Meike, B. Kraß, and M. Weyrich, “A Case Study for a Digital Twin of Body-in-White Production Systems General Concept for Automated Updating of Planning Projects in the Digital Factory” in *23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, Italien, 2018, S. 19–26, DOI: 10.1109/ETFA.2018.8502467.
- [114] F. Biesinger, D. Meike, B. Kraß, and M. Weyrich, “Methode zum automatischen Abgleich eines Digitalen Zwillings von Automatisierungskomponenten im Feld und deren digitalen Planungsständen”, *Automation*, Bd. 3, Nr. 04.07, 2018.
- [115] H. Zipper, F. Auris, A. Strahilov, and M. Paul, “Keeping the digital twin up-to-date — Process monitoring to identify changes in a plant” in *Proceedings of IEEE International Conference on Industrial Technology (ICIT)*, Lyon, France, 2018, S. 1592–1597, DOI: 10.1109/ICIT.2018.8352419.
- [116] S. Bougouffa, Q. H. Dong, S. Diehm, F. Gemein, and B. Vogel-Heuser, “Technical Debt indication in PLC Code for automated Production Systems: Introducing a Domain Specific Static Code Analysis Tool”, *IFAC-PapersOnLine*, Bd. 51, Nr. 10, S. 70–75, 2018, DOI: 10.1016/j.ifacol.2018.06.239.
- [117] M. B. Younis and G. Frey, “UML-based Approach for the Re-Engineering of PLC Programs” in *Proceedings IECON 2006: 32nd Annual Conference of the IEEE Industrial Electronics Society*, Paris, France, 2006, S. 3691–3696, DOI: 10.1109/IECON.2006.347292.
- [118] M. Obermeier, S. Braun, and B. Vogel-Heuser, “A Model-Driven Approach on Object-Oriented PLC Programming for Manufacturing Systems with Regard to Usability”, *IEEE Transactions on Industrial Informatics*, Bd. 11, Nr. 3, S. 790–800, 2015, DOI: 10.1109/TII.2014.2346133.

- [119] B. Ashtari Talkhestani, “Methodik zur Synchronisierung der Modelle des Digitalen Zwillings automatisierter Systeme” Dissertation, Institut für Automatisierungstechnik und Softwaresysteme, Universität Stuttgart, Stuttgart, 2020.
- [120] B. Ashtari Talkhestani, N. Jazdi, W. Schlögl, und M. Weyrich, “A concept in synchronization of virtual production system with real factory based on anchor-point method”, *Procedia CIRP*, Bd. 67, S. 13–17, 2018, DOI: 10.1016/j.procir.2017.12.168.
- [121] G. S. Martinez, S. Sierla, T. Karhela, und V. Vyatkin, “Automatic Generation of a Simulation-Based Digital Twin of an Industrial Process Plant” in *Proceedings IECON 2018: 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington D.C., USA, 2018, S. 3084–3089, DOI: 10.1109/IECON.2018.8591464.
- [122] O. Mathias, W. Gerrit, D. Oliver, L. Benjamin, S. Markus, und U. Leon, “Automatic Model Generation for Virtual Commissioning based on Plant Engineering Data”, *IFAC Proceedings Volumes*, Bd. 47, Nr. 3, S. 11635–11640, 2014, DOI: 10.3182/20140824-6-ZA-1003.01512.
- [123] S. Thongnuch, A. Fay, und R. Drath, “Semi-automatic generation of a virtual representation of a production cell”, *at - Automatisierungstechnik*, Bd. 66, Nr. 5, S. 372–384, 2018, DOI: 10.1515/auto-2017-0108.
- [124] D. Braun, F. Biesinger, N. Jazdi, und M. Weyrich, “A concept for the automated layout generation of an existing production line within the digital twin”, *Procedia CIRP*, Bd. 97, S. 302–307, 2020, DOI: 10.1016/j.procir.2020.05.242.
- [125] E. Westkämper, T. E. Baudisch, W. Schlögl, und G. Frank, “Automatic Model Generation for Virtual Commissioning of Specialized Production Machines”, *Softwaretechnik-Trends*, Bd. 32, Nr. 2, S. 82–83, 2012, DOI: 10.1007/BF03323491.
- [126] M. Sommer, J. Stjepandić, S. Stobrawa, und M. von Soden, “Improvement of Factory Planning by Automated Generation of a Digital Twin” in *Advances in Transdisciplinary Engineering*, Bd. 12, *Transdisciplinary Engineering for Complex Socio-Technical Systems - Real-life Applications: Proceedings of the 27th ISTE International Conference on Transdisciplinary Engineering*, J. Pokojski, M. Gil, L. Newnes, J. Stjepandic, und N. Wognum, Hrsg.: IOS Press Incorporated, 2020, DOI: 10.3233/ATDE200105.
- [127] IEC 62424: *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*, Jul. 2016.
- [128] L. Berardinelli, R. Drath, E. Maetzler, und M. Wimmer, “On the evolution of CAEX: A language engineering perspective” in *21st IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, 2016, S. 1–8, DOI: 10.1109/ETFA.2016.7733497.
- [129] 63278-1: *Verwaltungsschale für industrielle Anwendungen: Teil 1: Struktur der Verwaltungsschale*, Jul. 2022.
- [130] Bundesministerium für Wirtschaft und Energie (BMWi), Hrsg., “Verwaltungsschale in der Praxis: Wie definiere ich Teilmodelle, beispielhafte Teilmodelle und Interaktion zwischen Verwaltungsschalen” Diskussionspapier, Jul. 2020, [Online]. Verfügbar unter: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/2020-verwaltungsschale-in-der-praxis.pdf?__blob=publicationFile&v=1 (zuletzt geprüft am 11.09.2023).
- [131] UGS und PLM Components, Hrsg., “Open product lifecycle data sharing using XML: white paper”, [Online]. Verfügbar unter: https://www.plm.automation.siemens.com/legacy/products/open/plmxml/docs/wp_plm_xml_14.pdf (zuletzt geprüft am 08.09.2023).
- [132] J. Prinz, E. Yemenicioglu, und R. Drath, “AutomationML Export and Import Data Interfaces” in *AutomationML: The industrial Cookbook*, R. Drath, Hrsg.: De Gruyter, 2021, S. 41–47, DOI: 10.1515/9783110745979-004.
- [133] D. Braun, B. Ashtari, und M. Weyrich, “Integration of data and software into the Digital Twin via AML” in *AutomationML: The industrial Cookbook*, R. Drath, Hrsg.: De Gruyter, 2021, S. 521–538, DOI: 10.1515/9783110745979-031.
- [134] J. Berglund, E. Lindskog, J. Vallhagen, Z. Wang, C. Berlin, und B. Johansson, “Production System Geometry Assurance Using 3D Imaging”, *Procedia CIRP*, Bd. 44, S. 132–137, 2016, DOI: 10.1016/j.procir.2016.02.138.
- [135] DIN 81346-1: *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte - Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte: Teil 1: Allgemeine Regeln*, Mai. 2010.

- [136] D. Gutewort, A. Semm, und C. Erfurth, “Raising Awareness of SMEs for Real-Time Locating Systems (RTLS) in Intralogistics” in *Communications in Computer and Information Science*, Bd. 1404, *Innovations for Community Services*, U. R. Krieger, G. Eichler, C. Erfurth, und G. Fahrnberger, Hrsg., Cham: Springer International Publishing, 2021, S. 223–236, DOI: 10.1007/978-3-030-75004-6_15.
- [137] C. Küpper, J. Rösch, und H. Winkler, “Empirical findings for the usage of 5G as a basis for real time locating systems (RTLS) in the automotive industry”, *Procedia CIRP*, Bd. 107, S. 1287–1292, 2022, DOI: 10.1016/j.procir.2022.05.146.
- [138] C. Küpper, J. Rösch, und H. Winkler, “Use of real time localization systems (RTLS) in the automotive production and the prospects of 5G – A literature review”, *Production & Manufacturing Research*, Bd. 10, Nr. 1, S. 840–874, 2022, DOI: 10.1080/21693277.2022.2144522.
- [139] R. L. Ackoff, “From Data to Wisdom”, *Journal of applied systems analysis*, Bd. 16, S. 3–4, 1989.
- [140] D. Braun, M. Riedhammer, N. Jazdi, W. Schloegl, und M. Weyrich, “A methodology for the detection of functional relations of mechatronic components and assemblies in brownfield systems”, *Procedia CIRP*, Bd. 107, S. 119–124, 2022, DOI: 10.1016/j.procir.2022.04.020.
- [141] DIN 61131-1: *Speicherprogrammierbare Steuerungen: Teil 1: Allgemeine Informationen*, Mrz. 2004.
- [142] DIN 61131-3: *Speicherprogrammierbare Steuerungen: Teil 3: Programmiersprachen*, Jun. 2014.
- [143] DIN 61131-5: *Speicherprogrammierbare Steuerungen: Teil 5: Kommunikation*, Jan. 2001.
- [144] DIN 61499-1: *Funktionsbausteine für industrielle Leitsysteme: Teil 1: Architektur*, Sep. 2014.
- [145] DIN 61131-10: *Speicherprogrammierbare Steuerungen: Teil 10: XML-basiertes Austauschformat für Programme nach IEC 61131-3*, Nov. 2020.
- [146] DIN 81346-2: *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte - Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte: Teil 2: Klassifizierung von Objekten und Kennbuchstaben für Klassen*, Aug. 2017.
- [147] J. Rowley, “The wisdom hierarchy: representations of the DIKW hierarchy”, *Journal of Information Science*, Bd. 33, Nr. 2, S. 163–180, 2007, DOI: 10.1177/0165551506070706.
- [148] A. A. Akanmu, C. J. Anumba, und J. I. Messner, “An RTLS-Based Approach to Cyber-Physical Systems Integration in Design and Construction”, *International Journal of Distributed Sensor Networks*, Bd. 8, Nr. 12, S. 596845, 2012, DOI: 10.1155/2012/596845.
- [149] S. Kamm, S. S. Veekati, T. Müller, N. Jazdi, und M. Weyrich, “A survey on machine learning based analysis of heterogeneous data in industrial automation”, *Computers in Industry*, Bd. 149, S. 103930, 2023, DOI: 10.1016/j.compind.2023.103930.
- [150] Z. Xing, J. Pei, und E. Keogh, “A brief survey on sequence classification”, *ACM SIGKDD Explorations Newsletter*, Bd. 12, Nr. 1, S. 40–48, 2010, DOI: 10.1145/1882471.1882478.
- [151] B. H. Lindemann, “Datengetriebene Kompensation anomaler Prozessdynamiken in Automatisierungssystemen auf Basis adaptiver LSTM-Netze” Dissertation, Institut für Automatisierungstechnik und Softwaresysteme, Universität Stuttgart, Stuttgart, 2021.
- [152] F. Karim, S. Majumdar, und H. Darabi, “Insights Into LSTM Fully Convolutional Networks for Time Series Classification”, *IEEE Access*, Bd. 7, S. 67718–67725, 2019, DOI: 10.1109/ACCESS.2019.2916828.
- [153] C.-L. Liu, W.-H. Hsaio, und Y.-C. Tu, “Time Series Classification With Multivariate Convolutional Neural Network”, *IEEE Transactions on Industrial Electronics*, Bd. 66, Nr. 6, S. 4788–4797, 2019, DOI: 10.1109/TIE.2018.2864702.
- [154] F. Karim, S. Majumdar, H. Darabi, und S. Harford, “Multivariate LSTM-FCNs for time series classification”, *Neural networks : the official journal of the International Neural Network Society*, Bd. 116, S. 237–245, 2019, DOI: 10.1016/j.neunet.2019.04.014.
- [155] Y.-H. Lee, C.-P. Wei, T.-H. Cheng, und C.-T. Yang, “Nearest-neighbor-based approach to time-series classification”, *Decision Support Systems*, Bd. 53, Nr. 1, S. 207–217, 2012, DOI: 10.1016/j.dss.2011.12.014.
- [156] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, und P.-A. Muller, “Deep learning for time series classification: a review”, *Data Mining and Knowledge Discovery*, Bd. 33, Nr. 4, S. 917–963, 2019, DOI: 10.1007/s10618-019-00619-1.

- [157] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances”, *Data Mining and Knowledge Discovery*, Bd. 31, Nr. 3, S. 606–660, 2017, DOI: 10.1007/s10618-016-0483-9.
- [158] D. Braun, W. Schloegl, and M. Weyrich, “Automated data-driven creation of the Digital Twin of a brownfield plant” in *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, 2021, S. 1–7, DOI: 10.1109/ETFA45728.2021.9613676.
- [159] H. Baars und H.-G. Kemper, “Integration von Big Data-Komponenten in die Business Intelligence”, *Controlling*, Bd. 27, 4-5, S. 222–228, 2015, DOI: 10.15358/0935-0381-2015-4-5-222.
- [160] L. Stanescu, “A Comparison between a Relational and a Graph Database in the Context of a Recommendation System” in *Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems (ACSIS)*, 2021, S. 133–139, DOI: 10.15439/2021F33.
- [161] R. J. Sholichah, M. Imrona, and A. Alamsyah, “Performance Analysis of Neo4j and MySQL Databases using Public Policies Decision Making Data” in *7th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE): Proceedings*, Semarang, Indonesia, 2020, S. 152–157, DOI: 10.1109/ICITACEE50144.2020.9239206.
- [162] A. Hogan, E. Blomqvist, M. Cochez, C. D’amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, und A. Zimmermann, “Knowledge Graphs”, *ACM Computing Surveys*, Bd. 54, Nr. 4, S. 1–37, 2022, DOI: 10.1145/3447772.
- [163] N. Sahlab, S. Kamm, T. Muller, N. Jazdi, und M. Weyrich, “Knowledge Graphs as Enhancers of Intelligent Digital Twins” in *4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, Victoria, BC, Canada, 2021, S. 19–24, DOI: 10.1109/ICPS49255.2021.9468219.
- [164] A. J. Donkers, Djuan Yang, und N. H. Baken, “Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs” in *Linked Data in Architecture and Construction (LDAC)*, 2020, S. 23–36.
- [165] N. F. Noy, D. L. McGuinness, and others: *Ontology development 101: A guide to creating your first ontology*: Stanford knowledge systems laboratory technical report KSL-01-05 and ...
- [166] C. Hildebrandt, S. Törsleff, B. Caesar, und A. Fay, “Ontology Building for Cyber-Physical Systems: A domain expert-centric approach” in *14th IEEE International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, 2018, S. 1079–1086, DOI: 10.1109/COASE.2018.8560465.
- [167] N. Sahlab, D. Braun, C. Köhler, N. Jazdi, und M. Weyrich, “Extending the Intelligent Digital Twin with a context modeling service: A decision support use case”, *Procedia CIRP*, Bd. 107, S. 463–468, 2022, DOI: 10.1016/j.procir.2022.05.009.
- [168] C. Hildebrandt, A. Köcher, C. Küstner, C.-M. Lopez-Enriquez, A. W. Müller, B. Caesar, C. S. Gundlach, und A. Fay, “Ontology Building for Cyber-Physical Systems: Application in the Manufacturing Domain”, *IEEE Transactions on Automation Science and Engineering*, Bd. 17, Nr. 3, S. 1266–1282, 2020, DOI: 10.1109/TASE.2020.2991777.
- [169] VDI/VDE-Richtlinie 3682 Blatt 1: *Formalisierte Prozessbeschreibungen: Konzept und grafische Darstellung*, Mai. 2015.
- [170] ISO 23247 Part 1: *Automation systems and integration — Digital twin framework for manufacturing: Overview and general principles*, Okt. 2021.
- [171] ISO 23247 Part 2: *Automation systems and integration — Digital twin framework for manufacturing: Reference architecture*, Okt. 2021.
- [172] ISO 23247 Part 3: *Automation systems and integration — Digital twin framework for manufacturing: Digital representation of manufacturing elements*, Okt. 2021.
- [173] ISO 23247 Part 4: *Automation systems and integration — Digital twin framework for manufacturing: Information exchange*, Okt. 2021.
- [174] C. Huang, H. Cai, L. Xu, B. Xu, Y. Gu, und L. Jiang, “Data-driven ontology generation and evolution towards intelligent service in manufacturing systems”, *Future Generation Computer Systems*, Bd. 101, S. 197–207, 2019, DOI: 10.1016/j.future.2019.05.075.
- [175] R. Schiekofe, S. Grimm, M. M. Brandt, und M. Weyrich, “A formal mapping between OPC UA and the Semantic Web” in *17th IEEE International Conference on Industrial Informatics (INDIN)*, Helsinki, Finland, 2019, S. 33–40, DOI: 10.1109/INDIN41052.2019.8972102.

- [176] D. Braun, T. Müller, N. Sahlab, N. Jazdi, W. Schlögl, und M. Weyrich, “A graph-based knowledge representation and pattern mining supporting the Digital Twin creation of existing manufacturing systems” in *27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Stuttgart, Germany, 2022, S. 1–4, DOI: 10.1109/ETFA52439.2022.9921707.
- [177] M. Wörlein, T. Meinel, I. Fischer, und M. Philippsen, “A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston” in *Lecture notes in computer science Lecture notes in artificial intelligence*, Bd. 3721, *Knowledge discovery in databases: PKDD: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, A. Jorge, L. Torgo, P. Brazdil, R. Camacho, und J. Gama, Hrsg., Berlin, Heidelberg: Springer, 2005, S. 392–403, DOI: 10.1007/11564126_39.
- [178] X. Yan und J. Han, “gSpan: graph-based substructure pattern mining” in *Proceedings of IEEE International Conference on Data Mining (ICDM)*, Maebashi City, Japan, 2002, S. 721–724, DOI: 10.1109/ICDM.2002.1184038.
- [179] A. Faci, M.-J. Lesot, und C. Laudy, “cgSpan: Pattern Mining in Conceptual Graphs” in *Lecture Notes in Artificial Intelligence*, Bd. 12855, *Artificial Intelligence and Soft Computing: 20th International Conference ICAISC*, L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, und J. M. Zurada, Hrsg., Cham: Springer International Publishing, 2021, S. 149–158, DOI: 10.1007/978-3-030-87897-9_14.
- [180] C. Zhou, R. Hu, H. Zhang, Y. Lu, und L. Zou, “Mining Frequent Induced Subtree Patterns with Subtree-Constraint” in *6th IEEE International Conference on Data Mining Workshops (ICDMW)*, Hong Kong, China, 2006, S. 3–7, DOI: 10.1109/ICDMW.2006.112.
- [181] B. Ashtari Talkhestani und M. Weyrich, “Digital Twin of manufacturing systems: a case study on increasing the efficiency of reconfiguration”, *at - Automatisierungstechnik*, Bd. 68, Nr. 6, S. 435–444, 2020, DOI: 10.1515/auto-2020-0003.
- [182] N. Schmidt und A. Lüder, “AutomationML in a Nutshell”, AutomationML consortium, AutomationML: The Glue for Seamless Automation Engineering, Nov. 2015, [Online]. Verfügbar unter: https://www.automationml.org/wp-content/uploads/2021/06/AutomationML-in-a-Nutshell_151104.pdf (zuletzt geprüft am 11.07.2023).
- [183] R. Drath, Hrsg.: *AutomationML: The industrial Cookbook*: De Gruyter, 2021.
- [184] O. Musch: *Design patterns mit Java: Eine Einführung*. Wiesbaden, Heidelberg: Springer Vieweg, 2021, DOI: 10.1007/978-3-658-35492-3.
- [185] ISO 24730 Part 1: *Information technology — Real-time locating systems (RTLS): Application programming interface (API)*, Feb. 2014.
- [186] *Influx-OPCUA-logger*, Jeroen Coussement: Factory, 2016, MIT, [Online]. Verfügbar unter: <https://github.com/coussej/node-opcu-logger>.
- [187] *ssm_neo4j*, fangjiecao und illool, 2020, MIT, [Online]. Verfügbar unter: https://github.com/fangjiecao/ssm_neo4j.
- [188] D. Braun, N. Jazdi, W. Schlögl, und M. Weyrich, “Qualitative and quantitative evaluation of a methodology for the Digital Twin creation of brownfield production systems” in *28th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sinaia, Romania, 2023, S. 1–8, DOI: 10.1109/ETFA54631.2023.10275451.
- [189] N. Sahlab, D. Braun, T. Jung, N. Jazdi, und M. Weyrich, “A Tier-based Model for Realizing Context-Awareness of Digital Twins” in *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, 2021, S. 1–4, DOI: 10.1109/ETFA45728.2021.9613408.
- [190] A. Lüder, N. Schmidt, K. Hell, H. Röpke, und J. Zawisza, “Fundamentals of Artifact Reuse in CPPS” in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, S. Biffel, A. Lüder, und D. Gerhard, Hrsg., Cham: Springer International Publishing, 2017, S. 113–138, DOI: 10.1007/978-3-319-56345-9_5.
- [191] E. R. Alphonsus und M. O. Abdullah, “A review on the applications of programmable logic controllers (PLCs)”, *Renewable and Sustainable Energy Reviews*, Bd. 60, S. 1185–1205, 2016, DOI: 10.1016/j.rser.2016.01.025.
- [192] N. Gohert, “Automatische SPS-Codegenerierung für Syntheseverfahren der Supervisory Control Theory” Masterthesis, Institut für Informations- und Elektrotechnik, Hochschule für angewandte Wissenschaften Hamburg, Hamburg, 2014.

- [193] T. Cerquitelli, F. Ventura, D. Apiletti, E. Baralis, E. Macii, und M. Poncino, “Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes”, *Expert Systems with Applications*, Bd. 182, S. 115269, 2021, DOI: 10.1016/j.eswa.2021.115269.
- [194] H. Baars und H.-G. Kemper: *Business Intelligence & Analytics - Grundlagen und praktische Anwendungen: Ansätze der IT-basierten Entscheidungsunterstützung*, 4. Aufl. Wiesbaden, Heidelberg: Springer Vieweg, 2021, DOI: 10.1007/978-3-8348-2344-1.
- [195] Siemens: *SIMATIC RTLS: Lokalisierungssysteme SIMATIC RTLS4040T*. Betriebsanleitung, [Online]. Verfügbar unter: https://cache.industry.siemens.com/dl/files/504/109822504/att_1147298/v1/BA_RTLS4040T_0.pdf (zuletzt geprüft am 13.08.2023).
- [196] H. Vietz, M. Hirth, S. Baum, und M. Weyrich, “Synthetic Data Generation for improving Deep Learning-based 5G Indoor Positioning” in *28th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sinaia, Romania, 2023, S. 1–7, DOI: 10.1109/ETFA54631.2023.10275437.
- [197] Steam: *Anteil der User auf Steam nach Anzahl der vorhandenen Prozessor-Kerne im Juni 2023*. Statista, [Online]. Verfügbar unter: <https://de.statista.com/statistik/daten/studie/38755/umfrage/nutzungsanteil-von-pcs-auf-der-plattform-steam-nach-anzahl-der-cpus/> (zuletzt geprüft am 15.09.2023).
- [198] Steam: *Anteil der Steam-User mit Intel-CPU nach Prozessorgeschwindigkeit im Juni 2023*. Statista, [Online]. Verfügbar unter: <https://de.statista.com/statistik/daten/studie/38772/umfrage/nutzungsanteil-von-intel-cpus-bei-den-usern-der-plattform-steam-nach-prozessorgeschwindigkeit/> (zuletzt geprüft am 15.09.2023).
- [199] Steam: *Anteil der Steam-User nach Größe des vorhandenen Arbeitsspeichers (RAM) im Juni 2023*. Statista, [Online]. Verfügbar unter: <https://de.statista.com/statistik/daten/studie/38769/umfrage/nutzungsanteil-von-pcs-bei-den-usern-der-plattform-steam-nach-jeweiligem-arbeitsspeicher/> (zuletzt geprüft am 15.09.2023).

Anhang

Anhang 1: UML-Klassendiagramm des Metamodells aus TIA Portal Projekten der Version 14 & 15

